Speech Recognition

Technologies and Applications

Speech Recognition
Technologies and Applications

Edited by
France Mihelič and Janez Žibert

I-Tech

Published by In-Teh

In-Teh is Croatian branch of I-Tech Education and Publishing KG, Vienna, Austria.

www.in-teh.org
Additional copies can be obtained from:
publication@ars-journal.com

First published November 2008
Printed in Croatia

A catalogue record for this book is available from the University Library Rijeka under no. 120115073
Speech Recognition, Technologies and Applications, Edited by France Mihelič and Janez Žibert
p. cm.
ISBN 978-953-7619-29-9
1. Speech Recognition, Technologies and Applications, France Mihelič and Janez Žibert

Preface

After decades of research activity, speech recognition technologies have advanced in both the theoretical and practical domains. The technology of speech recognition has evolved from the first attempts at speech analysis with digital computers by James Flanagan's group at Bell Laboratories in the early 1960s, through to the introduction of dynamic time-warping pattern-matching techniques in the 1970s, which laid the foundations for the statistical modeling of speech in the 1980s that was pursued by Fred Jelinek and Jim Baker from IBM's T. J. Watson Research Center. In the years 1980-90, when Lawrence H. Rabiner introduced hidden Markov models to speech recognition, a statistical approach became ubiquitous in speech processing. This established the core technology of speech recognition and started the era of modern speech recognition engines. In the 1990s several efforts were made to increase the accuracy of speech recognition systems by modeling the speech with large amounts of speech data and by performing extensive evaluations of speech recognition in various tasks and in different languages. The degree of maturity reached by speech recognition technologies during these years also allowed the development of practical applications for voice human–computer interaction and audio-information retrieval. The great potential of such applications moved the focus of the research from recognizing the speech, collected in controlled environments and limited to strictly domain-oriented content, towards the modeling of conversational speech, with all its variability and language-specific problems. This has yielded the next generation of speech recognition systems, which aim to reliably recognize large-scale vocabulary, continuous speech, even in adverse acoustic environments and under different operating conditions. As such, the main issues today have become the robustness and scalability of automatic speech recognition systems and their integration into other speech processing applications. This book on Speech Recognition Technologies and Applications aims to address some of these issues.

Throughout the book the authors describe unique research problems together with their solutions in various areas of speech processing, with the emphasis on the robustness of the presented approaches and on the integration of language-specific information into speech recognition and other speech processing applications. The chapters in the first part of the book cover all the essential speech processing techniques for building robust, automatic speech recognition systems: the representation for speech signals and the methods for speech-features extraction, acoustic and language modeling, efficient algorithms for searching the hypothesis space, and multimodal approaches to speech recognition. The last part of the book is devoted to other speech processing applications that can use the information from automatic speech recognition for speaker identification and tracking, for

prosody modeling in emotion-detection systems and in other speech-processing applications that are able to operate in real-world environments, like mobile communication services and smart homes.

We would like to thank all the authors who have contributed to this book. For our part, we hope that by reading this book you will get many helpful ideas for your own research, which will help to bridge the gap between speech-recognition technology and applications.

Editors

France Mihelič,
University of Ljubljana, Slovenia
Janez Žibert,
University of Primorska, Slovenia

Contents

Feature extraction

1

A Family of Stereo-Based Stochastic Mapping Algorithms for Noisy Speech Recognition

Mohamed Afify[1], Xiaodong Cui[2] and Yuqing Gao[2]
[1]*Orange Labs, Smart Village,*
[2]*IBM T.J. Watson Research Center, Yorktown Heights,*
[1]*Cairo, Egypt*
[2]*NY, USA*

1. Introduction

The performance of speech recognition systems degrades significantly when they are operated in noisy conditions. For example, the automatic speech recognition (ASR) front-end of a speech-to-speech (S2S) translation prototype that is currently developed at IBM [11] shows noticeable increase in its word error rate (WER) when it is operated in real field noise. Thus, adding noise robustness to speech recognition systems is important, especially when they are deployed in real world conditions. Due to this practical importance noise robustness has become an active research area in speech recognition. Interesting reviews that cover a wide variety of techniques can be found in [12], [18], [19].

Noise robustness algorithms come in different flavors. Some techniques modify the features to make them more resistant to additive noise compared to traditional front-ends. These novel features include, for example, sub-band based processing [4] and time-frequency distributions [29]. Other algorithms adapt the model parameters to better match the noisy speech. These include generic adaptation algorithms like MLLR [20] or robustness techniques as model-based VTS [21] and parallel model combination (PMC) [9]. Yet other methods design transformations that map the noisy speech into a clean-like representation that is more suitable for decoding using clean speech models. These are usually referred to as feature compensation algorithms. Examples of feature compensation algorithms include general linear space transformations [5], [30], the vector Taylor series approach [26], and ALGONQUIN [8]. Also a very simple and popular technique for noise robustness is multi-style training (MST)[24]. In MST the models are trained by pooling clean data and noisy data that resembles the expected operating environment. Typically, MST improves the performance of ASR systems in noisy conditions. Even in this case, feature compensation can be applied in tandem with MST during both training and decoding. It usually results in better overall performance compared to MST alone. This combination of feature compensation and MST is often referred to as adaptive training [22].

In this chapter we introduce a family of feature compensation algorithms. The proposed transformations are built using stereo data, i.e. data that consists of simultaneous recordings of both the clean and noisy speech. The use of stereo data to build feature mappings was very popular in earlier noise robustness research. These include a family of cepstral

normalization algorithms that were proposed in [1] and extended in robustness research at CMU, a codebook based mapping algorithm [15], several linear and non-linear mapping algorithms as in [25], and probabilistic optimal filtering (POF) [27]. Interest in stereo-based methods then subsided, mainly due to the introduction of powerful linear transformation algorithms such as feature space maximum likelihood liner regression (FMLLR)[5], [30] (also widely known as CMLLR). These transformations alleviate the need for using stereo data and are thus more practical. In principle, these techniques replace the clean channel of the stereo data by the clean speech model in estimating the transformation. Recently, the introduction of SPLICE [6] renewed the interest in stereo-based techniques. This is on one hand due to its relatively rigorous formulation and on the other hand due to its excellent performance in AURORA evaluations. While it is generally difficult to obtain stereo data, it can be relatively easy to collect for certain scenarios, e.g. speech recognition in the car or speech corrupted by coding distortion. In some other situations it could be very expensive to collect field data necessary to construct appropriate transformations. In our S2S translation application, for example, all we have available is a set of noise samples of mismatch situations that will be possibly encountered in field deployment of the system. In this case stereo-data can also be easily generated by adding the example noise sources to the existing ”clean” training data. This was our basic motivation to investigate building transformations using stereo-data.

The basic idea of the proposed algorithms is to stack both the clean and noisy channels to form a large augmented space and to build statistical models in this new space. During testing, both the observed noisy features and the joint statistical model are used to predict the clean observations. One possibility is to use a Gaussian mixture model (GMM). We refer to the compensation algorithms that use a GMM as stereo-based stochastic mapping (SSM). In this case we develop two predictors, one is iterative and is based on maximum a posteriori (MAP) estimation, while the second is non-iterative and relies on minimum mean square error (MMSE) estimation. Another possibility is to train a hidden Markov model (HMM) in the augmented space, and we refer to this model and the associated algorithm as the stereo-HMM (SHMM). We limit the discussion to an MMSE predictor for the SHMM case. All the developed predictors are shown to reduce to a mixture of linear transformations weighted by the component posteriors. The parameters of the linear transformations are derived, as will be shown below, from the parameters of the joint distribution. The resulting mapping can be used on its own, as a front-end to a clean speech model, and also in conjunction with multistyle training (MST). Both scenarios will be discussed in the experiments. GMMs are used to construct mappings for different applications in speech processing. Two interesting examples are the simultaneous modeling of a bone sensor and a microphone for speech enhancement [13], and learning speaker mappings for voice morphing [32]. HMMcoupled with an N-bset formulation was recently used in speech enhancement in [34].

As mentioned above, for both the SSMand SHMM, the proposed algorithm is effectively a mixture of linear transformations weighted by component posteriors. Several recently proposed algoorithms use linear transformations weighted by posteriors computed from a Gaussian mixture model. These include the SPLICE algorithm [6] and the stochastic vector mapping (SVM)[14]. In addition to the previous explicit mixtures of linear transformations, a noise compensation algorithm in the log-spectral domain [3] shares the use of a GMM to model the joint distribution of the clean and noisy channels with SSM. Also joint uncertainty

decoding [23] employs a Gaussian model of the clean and noisy channels that is estimated using stereo data. Last but not least probabilistic optimal filtering (POF) [27] results in a mapping that resembles a special case of SSM. A discussion of the relationships between these techniques and the proposed method in the case of SSM will be given. Also the relationship in the case of an SHMMbased predictor to the work in [34] will be highlighted.
The rest of the chapter is organized as follows. We formulate the compensation algorithm in the case of a GMM and describe MAP-based and MMSE-based compensation in Section II. Section III discusses relationships between the SSM algorithm and some similar recently proposed techniques. The SHMM algorithm is then formulated in Section IV. Experimental results are given in Section V. We first test several variants of the SSM algorithm and compare it to SPLICE for digit recognition in the car environment. Then we give results when the algorithm is applied to large vocabulary English speech recognition. Finally results for the SHMM algorithm are presented for the Aurora database. A summary is given in Section VI.

2. Formulation of the SSM algorithm

This section first formulates the joint probability model of the clean and noisy channels in Section II-A, then derives two clean feature predictors; the first is based on MAP estimation in Section II-B, while the second is based on MMSE estimation in Section II-C. The relationships between the MAP and MMSE estimators are studied in Section II-D.

A. The Joint Probability Gaussian Mixture Model

Assume we have a set of stereo data $\{(x_i, y_i)\}$, where x is the clean (matched) feature representation of speech, and y is the corresponding noisy (mismatched) feature representation. Let N be the number of these feature vectors, i.e $1 \le i \le N$. The data itself is an M-dimensional vector which corresponds to any reasonable parameterization of the speech, e.g. cepstrum coefficients. In a direct extension the y can be viewed as a concatenation of several noisy vectors that are used to predict the clean observations. Define $z \equiv (x, y)$ as the concatenation of the two channels. The first step in constructing the mapping is training the joint probability model for $p(z)$. We use Gaussian mixtures for this purpose, and hence write

$$p(z) = \sum_{k=1}^{K} c_k \mathcal{N}(z; \mu_{z,k}, \Sigma_{zz,k}) \tag{1}$$

where K is the number of mixture components, c_k, $\mu_{z,k}$, and $\Sigma_{zz,k}$, are the mixture weights, means, and covariances of each component, respectively. In the most general case where L_n noisy vectors are used to predict L_c clean vectors, and the original parameter space is M-dimensional, z will be of size $M(L_c + L_n)$, and accordingly the mean μ_z will be of dimension $M(L_c + L_n)$ and the covariance Σ_{zz} will be of size $M(L_c + L_n) \times M(L_c + L_n)$. Also both the mean and covariance can be partitioned as

$$\mu_{z,k} = \begin{pmatrix} \mu_{x,k} \\ \\ \mu_{y,k} \end{pmatrix} \tag{2}$$

$$\Sigma_{zz,k} = \begin{pmatrix} \Sigma_{xx,k} & \Sigma_{xy,k} \\ \Sigma_{yx,k} & \Sigma_{yy,k} \end{pmatrix} \tag{3}$$

where subscripts x and y indicate the clean and noisy speech respectively.
The mixture model in Equation (1) can be estimated in a classical way using the expectation-maximization (EM) algorithm. Once this model is constructed it can be used during testing to estimate the clean speech features given the noisy observations. We give two formulations of the estimation process in the following subsections.

B. MAP-based Estimation

MAP-based estimation of the clean feature x given the noisy observation y can be formulated as:

$$\hat{x} = \underset{x}{\operatorname{argmax}}\, p(x|y) \tag{4}$$

The estimation in Equation (4) can be further decomposed as

$$\begin{aligned} \hat{x} &= \underset{x}{\operatorname{argmax}}\, p(x|y) = \underset{x}{\operatorname{argmax}} \sum_k p(x,k|y) \\ &\equiv \underset{x}{\operatorname{argmax}} \log \sum_k p(x,k|y) \end{aligned} \tag{5}$$

Now, define the log likelihood as $L(x) \equiv \log \sum_k p(x, k \mid y)$ and the auxiliary function $Q(x, \bar{x}) \equiv \sum_k p(k \mid \bar{x}, y) \log p(x, k \mid y)$. It can be shown by a straightforward application of Jensen's inequality that

$$L(x) - L(\bar{x}) \geq Q(x,\bar{x}) - Q(\bar{x},\bar{x}) \tag{6}$$

The proof is simple and is omitted for brevity. The above inequality implies that iterative optimization of the auxiliary function leads to a monotonic increase of the log likelihood. This type of iterative optimization is similar to the EM algorithm and has been used in numnerous estimation problems with missing data. Iterative optimization of the auxiliary objective function proceeds at each iteration as follows

$$\begin{aligned} \hat{x} &= \underset{x}{\operatorname{argmax}} \sum_k p(k|\bar{x},y) \log p(k|y) p(x|k,y) \\ &= \underset{x}{\operatorname{argmax}} \sum_k p(k|\bar{x},y) [\log p(k|y) + \log p(x|k,y)] \\ &\equiv \underset{x}{\operatorname{argmax}} \sum_k p(k|\bar{x},y) \log p(x|k,y) \\ &\equiv \underset{x}{\operatorname{argmax}} \frac{-1}{2} \sum_k p(k|\bar{x},y) \left[\log |\Sigma_{x|y,k}| + (x - \mu_{x|y,k})^T \Sigma_{x|y,k}^{-1} (x - \mu_{x|y,k}) \right] \end{aligned} \tag{7}$$

where $\bar{x}$ is the value of x from previous iteration, and $x|y$ is used to indicate the statistics of the conditional distribution $p(x|y)$. By differentiating Equation (7) with respect to x, setting the resulting derivative to zero, and solving for x, we arrive at the clean feature estimate given by

$$\sum_k p(k|\bar{x},y)\Sigma_{x|y,k}^{-1}\hat{x} = \sum_k p(k|\bar{x},y)\Sigma_{x|y,k}^{-1}\mu_{x|y,k} \tag{8}$$

which is basically a solution of a linear system of equations. $p(k|\bar{x}, y)$ are the usual posterior probabilities that can be calculated using the original mixture model and Bayes rule, and the conditional statistics are known to be

$$\mu_{x|y,k} = \mu_{x,k} + \Sigma_{xy,k}\Sigma_{yy,k}^{-1}(y - \mu_{y,k}) \tag{9}$$

$$\Sigma_{x|y,k} = \Sigma_{xx,k} - \Sigma_{xy,k}\Sigma_{yy,k}^{-1}\Sigma_{yx,k} \tag{10}$$

Both can be calculated from the joint distribution $p(z)$ using the partitioning in Equations (2) and (3). A reasonable initialization is to set $\bar{x} = y$, i.e. initialize the clean observations with the noisy observations.

An interesting special case arises when x is a scalar. This could correspond to using the i^{th} noisy coefficient to predict the i^{th} clean coefficient or alternatively using a time window around the i^{th} noisy coefficient to predict the i^{th} clean coefficient. In this case, the solution of the linear system in Equation (8) reduces to the following simple calculation for every vector dimension.

$$\hat{x} = \frac{\sum_k p(k|\bar{x},y)\mu_{x|y,k}/\sigma_{x|y,k}^2}{\sum_k p(k|\bar{x},y)/\sigma_{x|y,k}^2} \tag{11}$$

where $\sigma_{x|y,k}^2$ is used instead of $\Sigma_{x|y,k}$ to indicate that it is a scalar. This simplification will be used in the experiments. It is worth clarifying how the scalar Equation (11) is used for SSM with a time-window as mentioned above. In this case, and limiting our attention to a single feature dimension, the clean speech x is 1-dimensional, while the noisy speech y has the dimension of the window say L_n, and accordingly the mean $\mu_{x|y,k}$ and the variance $\sigma_{x|y,k}^2$ will be 1-dimensional. Hence, everything falls into place in Equation (11).

The mapping in Equations (8)-(10) can be rewritten, using simple rearrangement, as a mixture of linear transformations weighted by component posteriors as follows:

$$\hat{x} = \sum_k p(k|\bar{x},y)(A_k y + b_k) \tag{12}$$

where $A_k = CD_k$, $b_k = Ce_k$, and

$$C = \left(\sum_k p(k|\bar{x},y)\Sigma_{x|y,k}^{-1}\right)^{-1} \tag{13}$$

$$e_k = \Sigma_{x|y,k}^{-1}\left(\mu_{x,k} - \Sigma_{yy,k}^{-1}\Sigma_{xy,k}\mu_{y,k}\right) \tag{14}$$

$$D_k = \Sigma_{x|y,k}^{-1}\Sigma_{yy,k}^{-1}\Sigma_{xy,k} \tag{15}$$

C. MMSE-based Estimation

The MMSE estimate of the clean speech feature x given the noisy speech feature y is known to be the mean of the conditional distribution $p(x\,|\,y)$. This can be written as:

$$\hat{x} = E[x|y] \tag{16}$$

Considering the GMM structure of the joint distribution, Equation (16) can be further decomposed as

$$\begin{aligned} \hat{x} &= \int_x p(x|y)x\,dx = \sum_k \int_x p(x,k|y)x\,dx \\ &= \sum_k p(k|y)\int_x p(x|k,y)x\,dx \\ &= \sum_k p(k|y)E[x|k,y] \end{aligned} \tag{17}$$

In Equation (17), the posterior probability term $p(k\,|\,y)$ can be computed as

$$p(k|y) = \frac{p(k,y)}{p(y)} = \frac{p(y|k)p(k)}{\sum_k p(y|k)p(k)} \tag{18}$$

and the expectation term $E[x\,|\,k,\,y]$ is given in Equation (9).

Apart from the iterative nature of the MAP-based estimate the two estimators are quite similar. The scalar special case given in Section II-B can be easily extended to the MMSE case. Also the MMSE predictor can be written as a weighted sum of linear transformations as follows:

$$\hat{x} = \sum_k p(k|y)(F_k y + g_k) \tag{19}$$

where

$$F_k = \Sigma_{xy,k}\Sigma_{yy,k}^{-1} \tag{20}$$

$$g_k = \mu_{x,k} - \Sigma_{xy,k}\Sigma_{yy,k}^{-1}\mu_{y,k} \tag{21}$$

From the above formulation it is clear that the MMSE estimate is not performed iteratively and that no matrix inversion is required to calculate the estimate of Equation (19). More indepth study of the relationships between the MAP and the MMSE estimators will be given in Section II-D.

D. Relationships between MAP and MMSE Estimators

This section discusses some relationships between the MAP and MMSE estimators. Strictly speaking, the MMSE estimator is directly comparable to the MAP estimator only for the first iteration and when the latter is initialized from the noisy speech. However, the following discussion can be seen as a comparison of the structure of both estimators.

To highlight the iterative nature of the MAP estimator we rewrite Equation (12) by adding the iteration index as

$$\hat{x}^{(l)} = \sum_k p(k|\bar{x}^{(l-1)}, y)(A_k y + b_k) \tag{22}$$

where l stands for the iteration index. First, if we compare one iteration of Equation (22) to Equation (19) we can directly observe that the MAP estimate uses a posterior $p(k \mid \bar{x}^{(l-1)}, y)$ calculated from the joint probability distribution while the MMSE estimate employs a posterior $p(k \mid y)$ based on the marginal probability distribution. Second, if we compare the coefficients of the transformations in Equations (13)-(15) and (20)-(21) we can see that the MAP estimate has the extra term

$$\left(\sum_k p(k|\hat{x}^{(l-1)}, y) \Sigma_{x|y,k}^{-1} \right)^{-1} \tag{23}$$

which is the inversion of the weighted summation of conditional covariance matrices from each individual Gaussian component and that requires matrix inversion during run-time[1].

If we assume the conditional covariance matrix $\Sigma_{x|y,k}$ in Equation (23) is constant across k, i.e. all Gaussians in the GMM share the same conditional covariance matrix $\Sigma_{x|y}$, Equation (23) turns to

$$\begin{aligned} & \left(\sum_k p(k|\hat{x}^{(l-1)}, y) \Sigma_{x|y,k}^{-1} \right)^{-1} \\ = & \left(\Sigma_{x|y}^{-1} \sum_k p(k|\hat{x}^{(l-1)}, y) \right)^{-1} \\ = & \left(\Sigma_{x|y}^{-1} \cdot 1 \right)^{-1} = \Sigma_{x|y} \end{aligned} \tag{24}$$

and the coefficients A_k and b_k for the MAP estimate can be written as

$$\begin{aligned} A_k &= \Sigma_{x|y} \Sigma_{x|y}^{-1} \Sigma_{xy,k} \Sigma_{yy,k}^{-1} \\ &= \Sigma_{xy,k} \Sigma_{yy,k}^{-1} \end{aligned} \tag{25}$$

[1] Note that other inverses that appear in the equations can be pre-computed and stored.

$$\begin{aligned} b_k &= \Sigma_{x|y}\Sigma_{x|y}^{-1}\left(\mu_{x,k} - \Sigma_{xy,k}\Sigma_{yy,k}^{-1}\mu_{y,k}\right) \\ &= \mu_{x,k} - \Sigma_{xy,k}\Sigma_{yy,k}^{-1}\mu_{y,k} \end{aligned} \tag{26}$$

The coefficients in Equations (25) and (26) are exactly the same as those for the MMSE estimate that are given in Equations (20) and (21).

To summarize, the MAP and MMSE estimates use slightly different forms of posterior weighting that are based on the joint and marginal probability distributions respectively. The MAP estimate has an additional term that requires matrix inversion during run-time in the general case, but has a negligible overhead in the scalar case. Finally, one iteration of the MAP estimate reduces to the MMSE estimate if the conditional covariance matrix is tied across the mixture components. Experimental comparison between the two estimates is given in Section V.

3. Comparison between SSM and other similar techniques

As can be seen from Section II, SSM is effectively a mixture of linear transformations weighted by component posteriors. This is similar to several recently proposed algorithms. Some of these techniques are stereo-based such as SPLICE while others are derived from FMLLR. We discuss the relationships between the proposed method and both SPLICE and FMLLR-based methods in Sections III-A and III-B, respectively. Another recently proposed noise compensation method in the log-spectral domain also uses a Gaussian mixture model for the joint distribution of clean and noisy speech [3]. Joint uncertainty decoding [23] employs a joint Gaussian model for the clean and noisy channels, and probabilistic optimal filtering has a similar structure to SSMwith a time window. We finally discuss the relationship of the latter algorithms and SSM in Sections III-C,III-D, and III-E, respectively.

A. SSM and SPLICE

SPLICE is a recently proposed noise compensation algorithm that uses stereo data. In SPLICE, the estimate of the clean feature $\hat{x}$ is obtained as

$$\hat{x} = \sum_k p(k|y)(y + r_k) \tag{27}$$

where the bias term r_k of each component is estimated from stereo data (x_n, y_n) as

$$r_k = \frac{\sum_n p(k|y_n)(x_n - y_n)}{\sum_n p(k|y_n)} \tag{28}$$

and n is an index that runs over the data. The GMM used to estimate the posteriors in Equations (27) and (28) is built from noisy data. This is in contrast to SSM which employs a GMM that is built on the joint clean and noisy data.

Compared to MMSE-based SSM in Equations (19), (20) and (21), we can observe the following. First, SPLICE builds a GMM on noisy features while in this paper a GMM is built on the joint clean and noisy features (Equation (1)). Consequently, the posterior probability $p(k|y)$ in Equation (27) is computed from the noisy feature distribution while $p(k|y)$ in Equation (19) is computed from the joint distribution. Second, SPLICE is a special case of

SSM if the clean and noisy speech are assumed to be perfectly correlated. This can be seen as follows. If perfect correlation is assumed between the clean and noisy feature then $\Sigma_{xy,k} = \Sigma_{yy,k}$, and $p(k \mid x_n)=p(k \mid y_n)$. In this case, Equation (28) can be written as

$$\begin{aligned} r_k &= \frac{\sum_n p(k|y_n)(x_n - y_n)}{\sum_n p(k|y_n)} \\ &= \frac{\sum_n p(k|y_n)x_n - \sum_n p(k|y_n)y_n}{\sum_n p(k|y_n)} \\ &= \frac{\sum_n p(k|y_n)x_n}{\sum_n p(k|y_n)} - \frac{\sum_n p(k|y_n)y_n}{\sum_n p(k|y_n)} \\ &\doteq \frac{\sum_n p(k|x_n)x_n}{\sum_n p(k|x_n)} - \frac{\sum_n p(k|y_n)y_n}{\sum_n p(k|y_n)} \\ &= \mu_{x,k} - \mu_{y,k} \end{aligned} \tag{29}$$

The latter estimate will be identical to the MMSE estimate in Equations (20) and (21) when $\Sigma_{xy,k} = \Sigma_{yy,k}$.

To summarize, SPLICE and SSM have a subtle difference concerning the calculation of the weighting posteriors (noisy GMM vs. joint GMM), and SSM reduces to SPLICE if perfect correlation is assumed for the clean and noisy channels. An experimental comparison of SSM and SPLICE will be given in Section V.

B. SSM and FMLLR-based methods

There are several recently proposed techniques that use a mixture of FMLLR transforms. These can be written as

$$\hat{x} = \sum_k p(k|y)(U_k y + v_k) \tag{30}$$

where $p(k \mid y)$ is calculated using an auxiliary Gaussian mixture model that is typically trained on noisy observations, and U_k and v_k are the elements of FMLLR transformations that do not require stereo data for their estimation. These FMLLR-based methods are either applied during run-time for adaptation as in [28], [33], [16] or the transformation parameters are estimated off-line during training as in the stochastic vector mapping (SVM) [14]. Also online and offline transformations can be combined as suggested in [14]. SSM is similar in principle to training-based techniques and can be also combined with adaptation methods. This combination will be experimentally studied in Section V.

The major difference between SSM and the previous methods lies in the used GMM (again noisy channel vs. joint), and in the way the linear transformations are estimated (implicitly derived from the joint model vs. FMLLR-like). Also the current formulation of SSM allows the use of a linear projection rather than a linear transformation and most these techniques assume similar dimensions of the input and output spaces. However, their extension to a projection is fairly straightforward. In future work it will be interesting to carry out a systematic comparison between stereo and non-stereo techniques.

C. SSM and noise compensation in the log-spectral domain

A noise compensation technique in the log-spectral domain was proposed in [3]. This method, similar to SSM, uses a Gaussian mixture model for the joint distribution of clean

and noisy speech. However, the model of the noisy channel and the correlation model are not set free as in the case of SSM. They are parametrically related to the clean and noise distributions by the model of additive noise contamination in the log-spectral domain, and expressions of the noisy speech statistics and the correlation are explicitly derived. This fundamental difference results in two important practical consequences. First, in contrast to [3] SSM is not limited to additive noise compensation and can be used to correct for any type of mismatch. Second, it leads to relatively simple compensation transformations during run-time and no complicated expressions or numerical methods are needed during recognition.

D. SSM and joint uncertainty decoding

A recently proposed technique for noise compensation is joint uncertainty decoding (JUD)[23]. Apart from the fact that JUD employs the uncertainty decoding framework[7], [17], [31][2] instead of estimating the clean feature, it uses a joint model of the clean and noisy channels that is trained from stereo data. The latter model is very similar to SSM except it uses a Gaussian distribution instead of a Gaussian mixture model. On one hand, it is clear that a GMM has a better modeling capacity than a single Gaussian distribution. However, JUD also comes in a model-based formulation where the mapping is linked to the recognition model. This model-based approach has some similarity to the SHMM discussed below.

E. SSM and probabilistic optimal filtering (POF)

POF [27] is a technique for feature compensation that, similar to SSM, uses stereo data. In POF, the clean speech feature is estimated from a window of noisy features as follows:

$$\hat{x} = \sum_{i=1}^{I} p(i|z) W_i^T Y \tag{31}$$

where i is the vector quantization region index, I is the number of regions, z is a conditioning vector that is not necessarily limited to the noisy speech, Y consists of the noisy speech in a time window around the current vector, and W_i is the weight vector for region i. These weights are estimated during training from stereo data to minimize a conditional error for the region.

It is clear from the above presentation that POF bears similarities to SSM with a time window. However, some differences also exist. For example, the concept of the joint model allows the iterative refinement of the GMM parameters during training and these parameters are the equivalent to the region weights in POF. Also the use of a coherent statistical framework facilitates the use of different estimation criteria e.g. MAP and MMSE, and even the generalization of the transformation to the model space as will be discussed below. It is not clear how to perform these generalizations for POF.

4. Mathematical formulation of the stereo-HMM algorithm

In the previous sections we have shown how a GMM is built in an augmented space to model the joint distribution of the clean and noisy features, and how the resulting model is

[2] In uncertainty decoding the noisy speech pdf $p(y)$ is estimated rather than the clean speech feature.

used to construct feature compensation algorithm. In this section we extend the idea by training an HMM in the augmented space and formulate an appropriate feature compensation algorithm. We refer to the latter model as the stereo-HMM (SHMM).

Similar to the notation in Section II, denote a set of stereo features as $\{(x, y)\}$, where x is the clean speech feature vector, y is the corresponding noisy speech feature vector. In the most general case, y is L_n concatenated noisy vectors, and x is L_c concatenated clean vectors. Define $z \equiv (x, y)$ as the concatenation of the two channels. The concatenated feature vector z can be viewed as a new feature space where a Gaussian mixture HMM model can be built[3]. In the general case, when the feature space has dimension M, the new concatenated space will have a dimension $M(L_c + L_n)$. An interesting special case that greatly simplifies the problem arises when only one clean and noisy vectors are considered, and only the correlation between the same components of the clean and noisy feature vectors are taken into account. This reduces the problem to a space of dimension $2M$ with the covariance matrix of each Gaussian having the diagonal elements and the entries corresponding to the correlation between the same clean and noisy feature element, while all other covariance values are zeros.

Training of the above Gaussian mixture HMM will lead to the transition probabilities between states, the mixture weights, and the means and covariances of each Gaussian. The mean and covariance of the k^{th} component of state i can, similar to Equations (2) and (3), be partitioned as

$$\mu_{z,i,k} = \begin{pmatrix} \mu_{x,i,k} \\ \\ \mu_{y,i,k} \end{pmatrix} \tag{32}$$

$$\mu_{z,i,k} = \begin{pmatrix} \mu_{x,i,k} \\ \\ \mu_{y,i,k} \end{pmatrix} \tag{33}$$

where subscripts x and y indicate the clean and noisy speech features respectively.

For the k^{th} component of state i, given the observed noisy speech feature y, the MMSE estimate of the clean speech x is given by $E[x \mid y, i, k]$. Since (x, y) are jointly Gaussian, the expectation is known to be

$$\begin{aligned} & E[x|y,i,k] \\ = \; & \mu_{x|y,i,k} \\ = \; & \mu_{x,i,k} + \Sigma_{xy,i,k}\Sigma_{yy,i,k}^{-1}(y - \mu_{y,i,k}) \end{aligned} \tag{34}$$

[3] We will need the class labels in this case in contrast to the GMM.

The above expectation gives an estimate of the clean speech given the noisy speech when the state and mixture component index are known. However, this state and mixture component information is not known during decoding. In the rest of this section we show how to perform the estimation based on the N-best hypotheses in the stereo HMM framework.

Assume a transcription hypothesis of the noisy feature is *H*. Practically, this hypothesis can be obtained by decoding using the noisy marginal distribution $p(y)$ of the joint distribution $p(x, y)$. The estimate of the clean feature, $\hat{x}$, at time *t* is given as:

$$\begin{aligned}
\hat{x}_t &= E[x_t|y_1^T] \\
&= \int_{x_t} p(x_t|y_1^T) x_t dx_t \\
&= \sum_H \sum_i \sum_k \int_{x_t} p(x_t, i, k, H|y_1^T) x_t dx_t \\
&= \sum_H \sum_i \sum_k \int_{x_t} p(x_t, i, k|y_1^T, H) p(H|y_1^T) x_t dx_t \\
&= \sum_H p(H|y_1^T) \sum_i \sum_k p(i, k|y_1^T, H) \cdot \\
&\quad \int_{x_t} p(x_t|i, k, y_1^T, H) x_t dx_t
\end{aligned} \tag{35}$$

where the summation is over all the recognition hypotheses, the states, and the Gaussian components. The estimate in Equation (35) can be rewritten as:

$$\hat{x}_t = \sum_H p(H|y_1^T) \sum_i \sum_k \gamma_{ik}^H(t) E[x_t|y_t, i, k] \tag{36}$$

where $\gamma_{ik}^H(t) = p(s_t = i, \xi_t = k|y_1^T, H)$ is the posterior probability of staying at mixture component *k* of state *i* given the feature sequence y_1^T and hypothesis *H*. This posterior can be calculated by the forward-backward algorithm on the hypothesis *H*. The expectation term is calculated using Equation (34). $p(H|y_1^T)$ is the posterior probability of the hypothesis *H* and can be calculated from the N-best list as follows:

$$p(H|y_1^T) = \frac{p(y_1^T|H)^\nu p(H)^\nu}{\sum_j p(y_1^T|H_j)^\nu p(H_j)^\nu} \tag{37}$$

where the summation in the denominator is over all the hypotheses in the N-best list, and υ is a scaling factor that need to be experimentally tuned.

By comparing the estimation using the stereo HMM in Equation (36) with that using a GMM in the joint feature space as shown, for convenience, in Equation (38),

$$\hat{x}_t = \sum_k p(k|y_t) E[x_t|k, y_t] \tag{38}$$

we can find out the difference between the two estimates. In Equation (36), the estimation is carried out by weighting the MMSE estimate at different levels of granularity including Gaussians, states and hypotheses. Additionally, the whole sequence of feature vectors, y_1^T = (y_1, y_2, $\cdots$, y_T), has been exploited to denoise each individual feature vector x_t. Therefore, a better estimation of x_t is expected in Equation (36) over Equation (38).

Figure (1) illustrates the whole process of the proposed noise robust speech recognition scheme on stereo HMM. First of all, a traditional HMM is built in the joint (clean-noisy) feature space, which can be readily decomposed into a clean HMM and a noisy HMM as its marginals. For the input noisy speech signal, it is first decoded by the noisy marginal HMM to generate a word graph and also the N-best candidates. Afterwards, the MMSE estimate of the clean speech is calculated based on the generated N-best hypotheses as the conditional expectation of each frame given the whole noisy feature sequence. This estimate is a weighted average of Gaussian level MMSE predictors. Finally, the obtained clean speech estimate is re-decoded by the clean marginal HMM in a reduced searching space on the previously generated word graph.

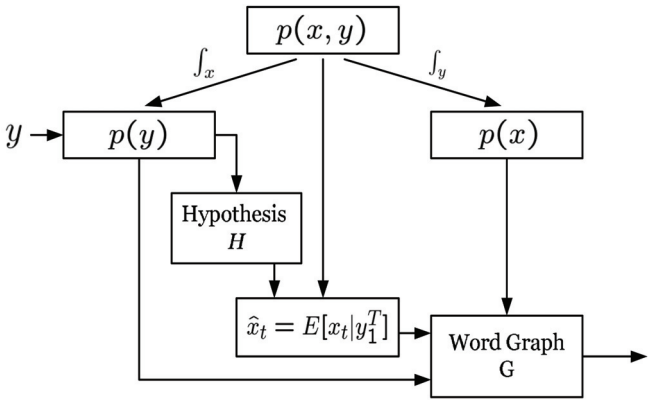


Fig. 1. Denoising scheme of N-best hypothesis based on stereo acoustic model.

A word graph based feature enhancement approach was investigated in [34] which is similar to the proposed work in the sense of two pass decoding using word graph. In [34], the word graph is generated by the clean acoustic model on enhanced noisy features using signal processing techniques and the clean speech is actually "synthesized" from the HMM Gaussian parameters using posteriori probabilities. Here, the clean speech is estimated from the noisy speech based on the joint Gaussian distributions between clean and noisy features.

5. Experimental evaluation

In the first part of this section we give results for digit recognition in the car environment and compare the SSM method to SPLICE. In the second part, we provide results when SSMis applied to large vocabulary spontaneous English speech recognition. Finally, we present SHMM results for the Aurora database.

A. SSM experiments for digit recognition in the car

The proposed algorithm is evaluated on a hands-free database (CARVUI database) recorded inside a moving car. The data was collected in Bell Labs area, under various driving conditions (highway/city roads) and noise environments (with and without radio/music in the background). About 2/3rd of the recordings contain music or babble noise in the background. Simultaneous recordings were made using a close-talking microphone and a 16-channel array of 1st order hypercardioid microphones mounted on the visor. A total of 56 speakers participated in the data collection, including many non-native speakers of American English. Evaluation is limited to the digit part of the data base. The speech material from 50 speakers is used for training, and the data from the 6 remaining speakers is used for test, leading to a total of about 6500 utterances available for training and 800 utterances for test. The test set contains about 3000 digits. The data is recorded at 24kHz sampling rate and is down-sampled to 8kHz and followed by MFCC feature extraction step for our speech recognition experiments. The feature vector consists of 39 dimensions, 13 cepstral coefficients and their first and second derivatives. Cepstral mean normalization (CMN) is applied on the utterance level. CMN is considered, to a first order approximation, as compensating for channel effects, and hence a channel parameter is not explicitly included in the compensation algorithm. The recognition task consists of simple loop grammar for the digits. In our experiments, data from 2 channels only are used. The first one is the close-talking microphone (CT), the second one is a single channel from the microphone array, referred to as Hands-Free data (HF) henceforward. 10 digit models and a silence model are built. Each model is left to right with no skipping having six states, and each state has 8 Gaussian distributions. Training and recognition is done using HTK [35]. A baseline set of results for this task are given in Table I.

Condition	WER
clean/clean	3.7
clean/noisy	14.1
noisy/noisy	6.1
clean/VTS	9.4
clean/COMP	6.9

Table I Baseline word error rate (WER) results (in %) of the close-talking (CT) microphone data and hands-free (HF) data

The first three lines refer to train/test conditions where the clean refers to the CT and noisy to the HF. The third line, in particular, refers to matched training on the HF data. The fourth and fifth lines correspond to using clean training and noisy test data that is compensated using conventional first order vector Taylor series (VTS) [26], and the compensation method in [3]. Both methods use a Gaussian mixture for the clean speech of size 64, and no explicit channel compensation is used as CMN is considered to partially account for channel effects. It can be observed from the table that the performance is clearly effected, as expected, by the addition of noise. Using noisy data for training improves the result considerably but not to the level of clean speech performance. VTS gives an improvement over the baseline, while the method in [3] shows a significant gain. More details about these compensation experiments can be found in [3] and other related publications.

The mapping is applied to the MFCC coefficients before CMN. After applying the compensation, CMN is performed followed by calculating the delta and delta-delta. Two methods were tested for constructing the mapping. In the first, a map is constructed between the same MFCC coefficient for the clean and noisy channels. In the second, a time window, including the current frame and its left and right contexts, around the i^{th} MFCC noisy coefficient is used to calculate the i^{th} clean MFCC coefficient. We tested windows of sizes three and five respectively. Thus we have mappings of dimensions 1 × 1, 3 × 1, and 5 × 1 for each cepstral dimension. These mappings are calculated according to Equation (11). In all cases, the joint Gaussian mixture model $p(z)$ is initialized by building a codebook on the stacked cepstrum vectors, i.e. by concatenation of the cepstra of the clean and noisy speech. This is followed by running three iterations of EMtraining. Similar initialization and training setup is also used for SPLICE. In this subsection only one iteration of the compensation algorithm is applied during testing. It was found in initial experiments that more iterations improve the likelihood, as measured by the mapping GMM, but slightly increase the WER. This comes in contrast to the large vocabulary results of the following section where iterations in some cases significantly improve performance. We do not have an explanation of this observation at the time of this writing.

In the first set of experiments we compare between SPLICE,MAP-SSMand MMSE-SSM, for different GMM sizes. No time window is used in these experiments. The results are shown in Table II. It can be observed that the proposed mapping outperforms SPLICE for all GMM sizes with the difference decreasing with increasing the GMM size. This makes sense because with increasing the number of Gaussian components, and accordingly the biases used in SPLICE, we can theoretically approximate any type of mismatch. Both methods are better than the VTS result in Table I, and are comparable to the method in [3]. The mapping in [3] is, however, more computationally expensive than SPLICE and SSM. Also, MAP-SSM and MMSE-SSM show very similar performance. This again comes in cotrast to what is observed in large vocabulary experiments where MMSE-SSMoutperforms MAP-SSM in some instances.

	16	64	256
SPLICE	9.0	8.6	8.3
MAP-SSM	8.3	8.3	8.0
MMSE-SSM	8.5	8.3	8.2

Table II Word error rate results (in %) of hands-free (HF) data using the proposed map-based mapping (MAP-SSM), SPLICE, and MMSE-SSM for different GMM sizes.

Finally Table III compares the MAP-SSM with and without the time window. We test windows of sizes 3 and 5. The size of the GMM used is 256. Using a time window gives an improvement over the baseline SSM with a slight cost during runtime. These results are not given for SPLICE because using biases requires that both the input and output spaces have the same dimensions, while the proposed mapping can be also viewed as a projection. The best SSM configuration, namely SSM-3, results in about 45% relative reduction in WER over the uncompensated result.

	SSM-1	SSM-3	SSM-5
WER	8.0	7.4	7.6

Table III Word error rate results (in %) of hands-free (HF) data using three different configurations of MAP-SSM for 256 GMM size and different time window size.

B. SSM experiments for large vocabulary spontaneous speech recognition

In this set of experiments the proposed technique is used for large vocabulary spontaneous English speech recognition. The mapping is applied with the clean speech models and also in conjunction with MST. The speech recognition setup and practical implementation of the mapping are first described in Section V-B.1. This is followed by compensation experiments for both the MAP and MMSE estimators in Section V-B.2.

B.1 Experimental setup

Experiments are run for a large vocabulary spontaneous speech recognition task. The original (clean) training data has about 150 hours of speech. This data is used to build the clean acoustic model. In addition to the clean model an MST model is also trained from the MST data. The MST data is formed by pooling the clean training data and noisy data. The noisy data are generated by adding humvee, tank and babble noise to the clean data at 15 dB. Different noise types are randomly added to different parts of each utterance. These three types of noise are chosen to match the military deployment environments in the DARPA Transtac Project. Thus, there are about 300 hours of training data in the MST case corresponding to the clean and 15 db SNR. When SSM is applied in conjunction with MST, the MST models are trained from SSM compensated data. This is done as follows. The SSM mapping is first trained as will be detailed below. It is then applied back to the noisy training data to yield noise-compensated features. Finally, the clean and noise compensated features are pooled and used to train the acoustic model. This is in the same spirit of using speaker-adaptive training (SAT) scheme, where some adaptation or compensation method is used in both training and decoding. The acoustic models are constructed in the same way and only differ in the type of the data used. Feature extraction, model training, mapping construction, and decoding will be outlined below.

The feature space of the acoustic models is formed as follows. First, 24 dimensional Mel-frequency cepstrum coefficients (MFCC) are calculated. The MFCC features are then mean normalized. 9 vectors, including the current vector and its left and right neighbours, are stacked leading to a 216-dimensional parameter space. The feature space is finally reduced to 40 dimensions using a combination of linear discriminant analysis (LDA) and a global semi-tied covariance (STC)matrix [10].

The acoustic model uses Gaussian mixture models associated to the leaves of a decision tree. The tree clustering is done by asking questions about quinphone context. The phoneme inventory has 54 phonemes for American English, and each phoneme is represented by a 3-state HMM. The model parameters are estimated using the forward-backward algorithm. First, a quinphone decision tree is built, an LDA matrix is computed and the model parameters are estimated using 40 EM iterations with the STC matrix updated each iteration. Upon finishing the estimation, the new model is used to re-generate the alignments based on which a new decision tree is built, the LDA matrix is re-computed and another 40 EM iterations are performed for model parameter estimation and STC matrix update. The clean model has 55K Gaussians while the MST models have 90K Gaussians. This difference is due to the difference in the amount of training data. The training and decoding are carried out on the IBM Attila toolkit.

Generally speaking SSM is SNR-specific and noise-type specific, i.e. a different mapping is built for each SNR and each noise type. However, as mentioned above we constructed only one mapping (at 15 dB) that corresponds to the mean SNR of the training data. The training of the mapping is straightforward. It amounts to the concatenation of the clean and noisy channels in the desired way and building a GMM using the EMalgorithm. All the mappings

used in the following experiments are of size 1024. It was confirmed in earlier work [2] that using larger sizes only give marginal improvements. The mapping is trained by starting from 256 random vectors, and then running one EM iteration and splitting until reaching the desired size. The final mapping is then refined by running 5 EM iterations. The mapping used in this section is scalar, i.e. it can be considered as separate mappings between the same coefficients in the clean and noisy channels. Although using different configurations can lead to better performance, as for example in Section V-A, this was done for simplicity. Given the structure of the feature vector used in our system, it is possible to build the mapping either in the 24-dimensional MFCC domain or in the 40-dimensional final feature space. It was also shown in [2] that building the mapping in the final feature space is better, and hence we restrict experiments in this work to mappings built in the 40-dimensional feature space. As discussed in Section II there are two possible estimators that can be used with SSM. Namely, the MAP and MMSE estimators. It should be noted that the training of the mapping in both cases is the same and that the only difference happens during testing, and possibly in storing some intermediate values for efficient implementation.

A Viterbi decoder that employs a finite state graph is used in this work. The graph is formed by first compiling the 32K pronunciation lexicon, the HMM topology, the decision tree, and the trigram language model into a large network. The resulting network is then optimized offline to a compact structure which supports very fast decoding. During decoding, generally speaking, the SNR must be known to be able to apply the correct mapping. Two possibilities can be considered, one is rather unrealistic and assumes that the SNR is given while the other uses an environment detector. The environment detector is another GMM that is trained to recognize different environments using the first 10 frames of the utterance. In [2], it was found that there is almost no loss in performance due to using the environment detector. In this section, however, only one mapping is trained and is used during decoding. Also as discussed in Section II the MAP estimator is iterative. Results with different number of iterations will be given in the experiments.

The experiments are carried out on two test sets both of which are collected in the DARPA Transtac project. The first test set (Set A) has 11 male speakers and 2070 utterances in total recorded in the clean condition. The utterances are spontaneous speech and are corrupted artificially by adding humvee, tank and babble noise to produce 15dB and 10dB noisy test data. The other test set (Set B) has 7 male speakers with 203 utterances from each. The utterances are recorded in a real-world environment with humvee and tank noise running in the background. This is a very noisy evaluation set and the utterances SNRs are measured around 5dB to 8dB, and we did not try to build other mappings to match these SNRs. This might also be considered as a test for the robustness of the mapping.

B.2 Experimental results

In this section SSM is evaluated for large vocabulary speech recognition. Two scenarios are considered, one with the clean speech model and the other in conjunction with MST. Also the combination of SSM with FMLLR adaptation is evaluated in both cases. For MAP-based SSM both one (MAP1) and three (MAP3) iterations are tested.

Table IV shows the results for the clean speech model. The first part of the table shows the uncompensated result, the second and third parts give the MAP-based SSM result for one and three iterations, respectively, while the final part presents MMSE-based SSM. In each part the result of combining FMLLR with SSM compensation is also given. The columns of the table correspond to the clean test data, artificially corrupted data at 15 dB, and 10 dB, and real field data. In all cases it can be seen that using FMLLR brings significant gain,

except in theMMSE-based SSMwhere it only leads to a slight improvement. MAP-based SSMshows some improvement only for test set B and using three iterations, in all other cases it does not improve on the clean result. MMSE-based SSM, on the other hand, shows excellent performance in all cases and outperforms its MAP-based counterpart. One explanation for this behavior can be attributed to the tying effect that is shown in Section II for MMSE estimation. In large vocabulary experiments a large mapping is needed to represent the new acoustic space with sufficient resolution. However, this comes at the expense of the robustness of the estimation. The implicit tying of the conditional covariances in the MMSE case can address this tradeoff and might be a reason of the improved performance in this case. Another way to address this, and that might be of benefit to the MAP-based algorithm is to construct the mapping in subspaces but this has to be experimentally confirmed. Finally, it is clear from the table that SSM does not hurt the clean speech performance. The best result for the real field data, which is for MMSE-based SSM with FMLLR, is 41% better than the baseline, and is 35% better than FMLLR alone.

	Set A			Set B
	clean	15 dB	10 dB	5-8 dB
clean model	4.84	18.40	33.66	47.72
clean model + fmllr	3.23	14.30	27.89	43.28
SSM_MAP1	4.87	18.05	33.32	48.24
SSM_MAP1 + fmllr	3.23	14.41	28.79	43.63
SSM_MAP3	4.87	18.03	33.36	46.04
SSM_MAP3 + fmllr	3.23	14.43	28.36	41.68
SSM_MMSE	4.84	13.39	25.52	28.43
SSM_MMSE + fmllr	3.26	13.23	25.12	28.25

Table IV Word error rate results (in %) of the compensation schemes against clean acoustic model

	Set A			Set B
	clean	15 dB	10 dB	5-8 dB
MST model	7.67	11.06	18.90	46.74
MST model + fmllr	3.87	7.69	14.13	25.87
SSM_MAP1	4.57	9.75	18.46	43.59
SSM_MAP1 + fmllr	2.74	6.96	14.07	23.83
SSM_MAP3	4.77	9.32	17.59	40.58
SSM_MAP3 + fmllr	2.76	6.79	13.78	22.85
SSM_MMSE	4.15	10.41	20.39	31.57
SSM_MMSE + fmllr	2.76	8.50	17.66	18.31

Table V Word error rate results (in %) of the compensation schemes against mst acoustic model

Table V displays the same results as table IV but for the MST case. The same trend as in table IV can be observed, i.e. FMLLR leads to large gains in all situations, and SSMbrings

decent improvements over FMLLR alone. In cotrast to the clean model case, MAP-based SSM and MMSE-based SSM are quite similar in most cases. This might be explained by the difference in nature in the mapping required for the clean and MST cases, and the fact that the model is trained on compensated data which in some sense reduces the effect of the robustness issue raised for the clean case above. The overall performance of the MST model is, unsurprisingly, better than the clean model. In this case the best setting for real field data, also MMSE-based SSM with FMLLR, is 60% better than the baseline and 41% better than FMLLR alone.

C. Experimental Results for Stereo-HMM

This section gives results of applying stereo-HMM compensation on the Sets A and B of the Aurora 2 database. There are four types of noise in the training set which include subway, babble, car and exhibition noise. The test set A has the same four types of noise as the training set while set B has four different types of noise, namely, restaurant, street, airport and station. For each type of noise, training data are recorded under five SNR conditions: clean, 20 dB, 15 dB, 10 dB and 5 dB while test data consist of six SNR conditions: clean, 20 dB, 15 dB, 10 dB, 5 dB and 0 dB. There are 8440 utterances in total for the four types of noise contributed by 55 male speaker and 55 female speakers. For the test set, each SNR condition of each noise type consists of 1001 utterances leading to 24024 utterances in total from 52 male speakers and 52 female speakers.

Word based HMMs are used, with each model having 16 states and 10 Gaussian distributions per state. The original feature space is of dimension 39 and consists of 12 MFCC coefficients, energy, and their first and second derivatives. In the training set, clean features and their corresponding noisy features are spliced together to form the stereo features. Thus, the joint space has dimension 78. First, a clean acoustic model is trained on clean features only on top of which single-pass re-training is performed to obtain the stereo acoustic model where the correlation between the corresponding clean and noisy components is only taken into account. Also a multi-style trained (MST) model is constructed in the original space to be used as a baseline. The results are shown in Tables VI-VIII. Both the MST model and the stereo model are trained on the mix of four types of training noise.

Set A	Clean	20dB	15dB	10dB	5dB	0dB
MST	99.0	98.5	97.8	95.4	89.4	67.0
SSM(1.0)	99.1	98.5	97.9	95.4	89.7	66.3
SSM(0.6)	99.1	98.6	98.0	95.6	89.7	66.6
SSM(0.3)	99.1	98.6	98.0	95.8	90.2	67.2
Set B	Clean	20dB	15dB	10dB	5dB	0dB
MST	99.0	97.7	95.7	92.1	83.2	59.3
SSM(1.0)	99.1	98.4	97.4	94.6	86.8	63.0
SSM(0.6)	99.1	98.5	97.3	94.6	86.7	63.1
SSM(0.3)	99.2	98.6	97.3	94.5	86.6	63.1

Table VI Accuracy on aurora 2 set A and set B. evaluated with $N = 5$.

A word graph, or lattice, is constructed for each utterance using the noisy marginal of the stereo HMMand converted into an N-best list. Different sizes of the list were tested and results for lists of sizes 5, 10 and 15 are shown in the tables. Hence, the summation in the

denominator of Equation (37) is performed over the list, and different values (1.0, 0.6 and 0.3) of the weighting v are evaluated (denoted in the parentheses in the tables). The language model probability $p(H)$ is taken to be uniform for this particular task. The clean speech feature is estimated using Equation (36). After the clean feature estimation, it is rescored using the clean marginal of the stereo HMM on the word graph. The accuracies are presented as the average across the four types of noise in each individual test set.

Set A	Clean	20dB	15dB	10dB	5dB	0dB
MST	99.0	98.5	97.8	95.4	89.4	67.0
SSM(1.0)	99.2	98.6	97.9	95.7	89.6	66.4
SSM(0.6)	99.2	98.6	97.9	95.7	89.8	66.7
SSM(0.3)	99.2	98.6	98.0	95.9	90.0	67.3
Set B	Clean	20dB	15dB	10dB	5dB	0dB
MST	99.0	97.7	95.7	92.1	83.2	59.3
SSM(1.0)	99.2	98.6	97.5	94.8	87.1	63.7
SSM(0.6)	99.2	98.6	97.5	94.7	87.1	63.7
SSM(0.3)	99.2	98.5	97.4	94.6	87.0	63.8

Table VII Accuracy on aurora 2 set A and set B. evaluated with N = 10.

From the tables we observe that the proposed N-best based SSMon stereo HMMperforms better than theMST model especially for unseen noise in Set B and at low SNRs. There are about 10%-20% word error rate (WER) reduction in Set B compared to the baseline MST model. It can be also seen that there is little influence for the weighting factor, this might be due to the uniform language model used in this task but might change for other scenarios. By increasing the number of N-best candidates in the estimation, the performance increases but not significantly.

Set A	Clean	20dB	15dB	10dB	5dB	0dB
MST	99.0	98.5	97.8	95.4	89.4	67.0
SSM(1.0)	99.2	98.6	97.9	95.7	89.8	66.4
SSM(0.6)	99.2	98.6	97.9	95.8	89.9	66.7
SSM(0.3)	99.2	98.6	98.0	96.0	90.1	67.1
Set B	Clean	20dB	15dB	10dB	5dB	0dB
MST	99.0	97.7	95.7	92.1	83.2	59.3
SSM(1.0)	99.2	98.4	97.5	94.8	87.2	63.8
SSM(0.6)	99.2	98.5	97.5	94.8	87.2	63.9
SSM(0.3)	99.2	98.5	97.4	94.6	87.1	64.0

Table VIII Accuracy on aurora 2 set A and set B. evaluated with N = 15.

6. Summary

This chapter presents a family of feature compensation algorithms for noise robust speech recognition that use stereo data. The basic idea of the proposed algorithms is to stack the features of the clean and noisy channels to form a new augmented space, and to train

statistical models in this new space. These statistical models are then used during decoding to predict the clean features from the observed noisy features. Two types of models are studied. Gaussian mixture models which lead to the so-called stereo-based stochastic mapping (SSM)algorithm, and hidden Markov models which result in the stereo-HMM (SHMM) algorithm. Two types of predictors are examined for SSM, one is based on MAP estimation while the other is based on MMSE estimation. Only MMSE estimation is used for the SHMM, where an N-best list is used to provide the required recognition hypothesis. The algorithms are extensively evaluated in speech recognition experiments. SSM is tested for both digit recognition in the car, and a large vocabulary spontaneous speech task. SHMM is evaluated on the Aurora task. In all cases the proposed methods lead to significant gains.

7. References

A. Acero, Acoustical and environmental robustness for automatic speech recognition, Ph.D. Thesis, ECE Department, CMU, September 1990.

M. Afify, X. Cui and Y. Gao, "Stereo-Based Stochastic Mapping for Robust Speech Recognition," in Proc. ICASSP'07, Honolulu, HI, April 2007.

M. Afify, "Accurate compensation in the log-spectral domain for noisy speech recognition," in IEEE Trans. on Speech and Audio Processing, vol. 13, no. 3, May 2005.

H. Bourlard, and S. Dupont, "Subband-based speech recognition," in Proc. ICASSP'97, Munich, Germany, April 1997.

V. Digalakis, D. Rtischev, and L. Neumeyer, "Speaker adaptation by constrained estimation of Gaussian mixtures," IEEE Transactions on Speech and Audio Processing, vol. 3, no. 5, pp. 357-366, 1995.

J. Droppo, L. Deng, and A. Acero, "Evaluation of the SPLICE Algorithm on the AURORA 2 Database," in Proc. Eurospeech'01, Aalborg, Denmark, September, 2001.

J. Droppo, L. Deng, and A. Acero, "Uncertainity decoding with splice for noise robust speech recognition," in Proc. ICASSP'02, Orlando, Florida, May 2002.

B. Frey, L. Deng, A. Acero, and T. Kristjanson, "ALGONQUIN: Iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition," in Proc. Eurospeech'01, Aalborg, Denmark, September, 2001.

M. Gales, and S. Young, "Robust continuous speech recognition using parallel model combination," IEEE Transactions on Speech and Audio Processing, vol. 4, 1996.

M. Gales, "Semi-tied covariance matrices for hidden Markov models," IEEE Transactions on Speech and Audio Processing, vol. 7, pp. 272-281, 1999.

Y. Gao, B. Zhou, L. Gu, R. Sarikaya, H.-K. Kuo. A.-V.I. Rosti,M. Afify,W. Zhu, "IBMMASTOR:Multilingual automatic speech-to-speech translator," Proc. ICASSP'06, Tolouse, France, 2006.

Y.Gong, '"Speech recognition in noisy environments: A survey," Speech Communication, Vol.16, pp.261-291, April 1995.

J. Hershey, T. Kristjansson, and Z. Zhang, "Model-based fusion of bone and air sensors for speech enhancement and robust speech recognition," in ISCAWorkshop on statistical and perceptual audio processing, 2004.

Q. Huo, and D. Zhu, "A maximum likelihood training approach to irrelevant variability compensation based on piecewise linear transformations," in Proc. Interspeech'06, Pittsburgh, Pennsylvania, September, 2006.

B.H. Juang, and L.R. Rabiner,"Signal restoration by spectral mapping," in Proc. ICASSP'87, pp.2368-2372, April 1987.

S. Kozat, K. Visweswariah, and R. Gopinath, "Feature adaptation based on Gaussian posteriors," in Proc. ICASSP'06, Tolouse, France, April 2006.
T. Kristjansson, B. Frey, "Accounting for uncertainity in observations: A new paradigm for robust speech recognition," in Proc. ICASSP'02, Orlando, Florida, May 2002.
C.H. Lee, "On stochastic feature and model compensation approaches to robust speech recognition," Speech Communication, vol. 25, pp. 29-47, 1998.
C.H. Lee, and Q. Huo, "On adaptive decision rules and decision parameter adaptation for automatic speech recognition," Proceedings of the IEEE, vol. 88, no. 88, pp. 1241-1269, August 2000.
C. Leggetter, and P. Woodland, "Flexible speaker adaptation using maximum likelihood linear regression," in Proc. ARPA spoken language technology workshop, pp. 104-109, Feb 1995.
J. Li, L. Deng, Y. Gong, and A. Acero, "High performance HMMadaptation with joint compensation of additive and convolutive distortions via vector Taylor series," in Proc. ASRU 2007, Kyoto, Japan, 2007.
H. Liao, and M. Gales, "Adaptive training with joint uncertainity decoding for robust recognition of noisy data," in Proc. ICASSP'07, Honolulu, HI, April 2007.
H. Liao, and M. Gales, "Joint uncertainity decoding for noise robust speech recognition," in Proc. Eurospeech'05, Lisbone, Portugal, September 2005.
R. Lippmann, E. Martin, and D. Paul, "Multi-style training for robust isolated-word recognition," Proc. of DARPA Speech Recognition Workshop, Mar. 24-26, 1987, pp. 96-99.
C. Mokbel, and G. Chollet,"Word recognition in the car:Speech enhancement/Spectral transformations," in Proc. ICASSP'91, Toronto, 1991.
P.J. Moreno, B. Raj, and R.M. Stern, "A vector Taylor series approach for environment-independent speech recognition," in Proc. ICASSP, Atlanta, GA, May 1996, pp.733-736.
L. Neumeyer, and M. Weintraub, "Probabilistic optimal filtering for robust speech recognition," in Proc. ICASSP'94, Adelaide, Australia, April 1994.
M.K. Omar, personal communication.
A. Potamianos, and P. Maragos, "Time-frequency distributions for automatic speech recognition, "IEEE Transactions on Speech and Audio Processing, vol. 9, pp. 196-200, March 2001.
G. Saon, G. Zweig, and M. Padmanabhan, "Linear feature space projections for speaker adaptation," in Proc. ICASSP'01, Salt lake City,Utah, April, 2001.
V. Stouten, H. Van Hamme, and P. Wambacq, "Accounting for the uncertainity of speech estimates in the context of model-based feature enhancement," in Proc. ICSLP'04, Jeju, Korea, September 2004.
Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," IEEE Transactions on Speech and Audio Processing, vol. 6, pp. 131-142, January 1998.
K. Visweswariah, and P. Olsen, "Feature adaptation using projection of Gaussian posteriors," in Proc. Interspeech'05, Lisbone, Portugal, September 2005.
Zh. Yan, F. Soong, and R.Wang, "Word graph based feature enhancement for noisy speech recognition," Proc. ICASSP, Honolulu, HI, April 2007.
S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P.Woodland, The HTK book (for HTK Version 3.1), December 2001.

2

Histogram Equalization for Robust Speech Recognition

Luz García, Jose Carlos Segura, Ángel de la Torre,
Carmen Benítez and Antonio J. Rubio
University of Granada
Spain

1. Introduction

Optimal Automatic Speech Recognition takes place when the evaluation is done under circumstances identical to those in which the recognition system was trained. In the speech applications demanded in the actual real world this will almost never happen. There are several variability sources which produce mismatches between the training and test conditions.
Depending on his physical or emotional state, a speaker will produce sounds with unwanted variations transmitting no acoustic relevant information. The phonetic context of the sounds produced will also introduce undesired variations. Inter-speaker variations must be added to those intra-speaker variations. They are related to the peculiarities of speakers' vocal track, his gender, his socio-linguistic environment, etc. A third source of variability is constituted by the changes produced in the speaker's environment and the characteristics of the channel used to communicate. The strategies used to eliminate the group of environmental sources of variation are called *Robust Recognition Techniques*. Robust Speech Recognition is therefore the recognition made as invulnerable as possible to the changes produced in the evaluation environment. Robustness techniques constitute a fundamental area of research for voice processing. The current challenges for automatic speech recognition can be framed within these work lines:

- Speech recognition of coded voice over telephone channels. This task adds an additional difficulty: each telephone channel has its own SNR and frequency response. Speech recognition over telephone lines must perform a channel adaptation with very few specific data channels.
- Low SNR environments. Speech Recognition during the 80's was done inside a silent room with a table microphone. At this moment, the scenarios demanding automatic speech recognition are:
  - Mobile phones.
  - Moving cars.
  - Spontaneous speech.
  - Speech masked by other speech.
  - Speech masked by music.
  - Non-stationary noises.
- Co-channel voice interferences. Interferences caused by other speakers constitute a bigger challenge than those changes in the recognition environment due to wide band noises.

- Quick adaptation for non-native speakers. Current voice applications demand robustness and adaptation to non-native speakers' accents.
- Databases with realistic degradations. Formulation, recording and spreading of voice databases containing realistic examples of the degradation existing in practical environments are needed to face the existing challenges in voice recognition.

This chapter will analyze the effects of additive noise in the speech signal, and the existing strategies to fight those effects, in order to focus on a group of techniques called statistical matching techniques. Histogram Equalization –*HEQ*- will be introduced and analyzed as main representative of this family of Robustness Algorithms. Finally, an improved version of the Histogram Equalization named Parametric Histogram Equalization -PEQ- will be exposed.

2. Voice feature normalization

2.1 Effects of additive noise

Within the framework of Automatic Speech Recognition, the phenomenon of noise can be defined as the non desired sound which distorts the information transmitted in the acoustic signal difficulting its correct perception. There are two main sources of distortion for the voice signal: **additive noise** and **channel distortion**.

Channel distortion is defined as the noise convolutionally mixed with speech in the time domain. It appears as a consequence of the signal reverberations during its transmission, the frequency response of the microphone used, or peculiarities of the transmission channel such as an electrical filter within the A/D filters for example. The effects of channel distortion have been fought with certain success as they become linear once the signal is analyzed in the frequency domain. Techniques such as RASTA filtering, echo cancellation or Cepstral mean subtraction have proved to eliminate its effects.

Additive noise is summed to the speech signal in the time domain and its effects in the frequency domain are not easily removed as it has the peculiarity to transform speech non-linearly in certain domains of analysis. Nowadays, additive noise constitutes the driving force of research in ASR: additive white noises, door slams, spontaneous overlapped voices, background music, etc.

The most used model to analyze the effects of noise in the oral communication (Huang, 2001) represents noise as a combination of additive and convolutional noise following the expression:

$$y[m] = x[m] * h[m] + n[m] \tag{1}$$

Assuming that the noise component *n[m]* and the speech signal *x[m]* are statistically independent, the resulting noisy speech signal *y[m]* will follow equation (2) for the *ith* channel of the filter bank:

$$\left|Y(f_i)\right|^2 \cong \left|X(f_i)\right|^2 \bullet \left|H(f_i)\right|^2 + \left|N(f_i)\right|^2 \tag{2}$$

Taking logarithms in expression *(2)* and operating, the following approximation in the frequency domain can be obtained:

$$\ln\left|Y(f_i)\right|^2 \cong \ln\left|X(f_i)\right|^2 + \ln\left|H(f_i)\right|^2 + \ln(1+\exp(\left|N(f_i)\right|^2 - \ln\left|X(f_i)\right|^2 - \ln\left|H(f_i)\right|^2)) \tag{3}$$

In order to move expression *(3)* to the Cepstral domain with *M+1* Cepstral coefficients, the following 4 matrixes are defined, using *C()* to denote the discrete cosine transform:

$$\begin{aligned} x &= C(\ln|X(f_0)|^2 \quad \ln|X(f_1)|^2 \ .... \ \ln|X(f_M)|^2) \\ h &= C(\ln|H(f_0)|^2 \quad \ln|H(f_1)|^2 \ .... \ \ln|H(f_M)|^2) \\ n &= C(\ln|N(f_0)|^2 \quad \ln|N(f_1)|^2 \ .... \ \ln|N(f_M)|^2) \\ y &= C(\ln|Y(f_0)|^2 \quad \ln|Y(f_1)|^2 \ .... \ \ln|Y(f_M)|^2) \end{aligned} \tag{4}$$

The following expression can be obtained for the noisy speech signal *y* in the Cepstral domain combining equations *(3)* and *(4)*:

$$\hat{y} = \hat{x} + \hat{h} + g(\hat{n} - \hat{x} - \hat{h}) \tag{5}$$

being function *g* of equation *(5)* defined as:

$$g(z) = C(\ln(1 + e^{C^{-1}(z)})) \tag{6}$$

Based on the relative facility to remove it (via linear filtering), and in order to simplify the analysis, we will consider absence of convolutional channel distortion, that is, we will consider *H(f)=1*. The expression of the noisy signal in the Cepstral domain becomes then:

$$y = x + \ln(1 + \exp(n - x)) \tag{7}$$

The relation between the clean signal *x* and the noisy signal *y* contaminated with additive noise is modelled in expression *(7)*. There is a linear relation between both for high values of *x*, which becomes non linear when the signal energy approximates or is lower than the energy of noise.
Figure 1 shows a numeric example of this behaviour. The logarithmic energy of a signal *y* contaminated with an additive Gaussian noise with average μ_n=3 and standard deviation σ_n=0,4 is pictured. The solid line represents the average transformation of the logarithmic energy, while the dots represent the transformed data. The average transformation can be inverted to obtain the expected value for the clean signal once the noisy signal is observed. In any case there will be a certain degree of uncertainty in the clean signal estimation, depending on the SNR of the transformed point. For values of *y* with energy much higher than noise the degree of uncertainty will be small. For values of *y* close to the energy of noise, the degree of uncertainty will be high. This lack of linearity in the distortion is a common feature of additive noise in the Cepstral domain.

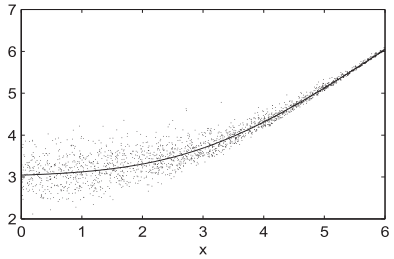


Fig. 1. Transformation due to additive noise.

The analysis of the histograms of the MFCCs probability density function of a clean signal versus a noisy signal contaminated with additive noise shows the following effects of noise (De la Torre et al., 2002):

- A shift in the mean value of the MFCC histogram of the contaminated signal.
- A reduction in the variance of such histogram.
- A modification in the histogram global shape. This is equivalent to a modification of the histogram's statistical higher order moments. This modification is especially remarkable for the logarithmic energy and the lower order coefficients C_0 and C_1.

2.2 Robust speech recognition techniques

There are several classifications of the existing techniques to make speech recognition robust against environmental changes and noise. A commonly used classification is the one that divides them into pre-processing techniques, feature normalization techniques and model adaptation techniques according to the point of the recognition process in which robustness is introduced (see Figure 2):

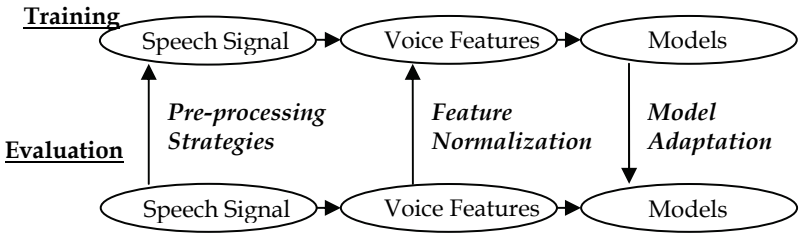


Fig. 2. Robust Recognition Strategies.

- **Signal Pre-processing Techniques**: their aim is to remove noise before the voice signal parameterization is done, in order to obtain a parameterization as close as possible to the clean signal parameterization. They are based on the idea that voice and noise are uncorrelated, and therefore they are additive in the time domain. Consequently their power spectrum of a noisy signal will be the sum of the voice and noise power spectra. The main techniques within this group are Linear Spectral Subtraction (Boll, 1979), Non-linear Spectral Subtraction (Lockwood & Boudy, 1992), Wiener Filtering (Wiener, 1949) or Ephraim Malah noise suppression rule (Ephraim & Malah, 1985).
- **Feature Normalization Techniques**: the environment distortion is eliminated once the voice signal has been parameterized. Through different processing techniques like high pass Cepstral filtering, models of the noise effects, etc., the clean voice features are recovered from the noisy voice features. Three sub-categories can be found within this group of techniques:
  - **High Band Pass filtering techniques**. They add a quite high level of robustness to the recognizer with a low cost and therefore they are included in the most of the automatic recognition front-ends. Their objective is forcing the mean value of the Cepstral coefficients to be zero. With this condition they eliminate unknown linear filtering effects that the channel might have. The most important techniques within

this subgroup are RASTA filtering (Hermansky & Morgan, 1994) and CMN, *Cepstral Mean Normalization*- (Furui, 1981).

- **Noise compensation with stereo data.** This group of techniques compares the noisy voice features with those of clean stereo data. The result of such comparison is a correction of the environment which is added to the feature vector before entering the recognizer. RATZS –*multivaRiate gAussian based cepsTral normaliZation*- (Moreno et al., 1995) and SPLICE –*Stereo-based Piecewise LInear Compensation for Environments*- (Deng et al., 2000) are the most representative strategies in this group.
- **Noise compensation based on an environment model.** These techniques give an analytical expression of the environmental degradation and therefore need very few empirical data to normalize the features. (In contraposition to the compensation using stereo data). Degradation is defined as a filter and a noise such that when they are inversely applied, the probability of the normalized observations becomes the maximum. The most relevant algorithm within this category is VTS –*Vector Taylor Series approach*- (Moreno et al., 2006).
- **Statistical Matching Algorithms.** Set of algorithms for feature normalization which define linear and non-linear transformations in order to modify the statistics of noisy speech and make them equal to those of clean speech. Cepstral Mean Normalization, which was firstly classified as a high band pass filtering technique, corresponds as well to the definition of statistical matching algorithms. The most relevant ones are CMNV –*Cepstral Mean and Variance Normalization*- (Viiki et al., 1998), Normalization of a higher number of statistical moments (Khademul et al., 2004),(Chang Wen & Lin Shan, 2004),(Peinado & Segura, 2006) and Histogram Equalization (De la Torre et al., 2005),(Hilger & Ney, 2006). This group of strategies, and specially Histogram Equalization, constitute the core of this chapter and they will be analyzed in depth in order to see their advantages and to propose an alternative to overcome their limitations.

- **Model Adaptation Techniques**. They modify the classifier in order to make the classification optimal for the noisy voice features. The acoustic models obtained during the training phase are adapted to the test conditions using a set of adaptation data from the noisy environment. This procedure is used both for environment adaptation and for speaker adaptation. The most common adaptation strategies are MLLR –*Maximum Likelihood Linear Regression*- (Gales & Woodland, 1996) (Young et al. 1995), MAP – *Mamixum a Posteriori Adaptation* - (Gauvain & Lee, 1994), PMC- *Parallel Model Combination* (Gales & Young, 1993), and non linear model transformations like the ones performed using Neural Networks (Yuk et al., 1996) or (Yukyz & Flanagany, 1999).

The robust recognition methods exposed below work on the hypothesis of a stationary additive noise, that is, the noise power spectral density does not change with time. They are narrow-band noises. Other type of non-stationary additive noises with a big importance on robust speech recognition exist: door slams, spontaneous speech, the effect of lips or breath, etc. For the case of these transient noises with statistical properties changing with time, other techniques have been developed under the philosophy of simulating the human perception mechanisms: signal components with a high SNR are processed, while those components with low SNR are ignored. The most representative techniques within this group are the Missing Features Approach (Raj et al. 2001) (Raj et al. 2005), and Multiband Recognition (Tibrewala & Hermansky, 1997) (Okawa et al. 1999).

2.3 Statistical matching algorithms

This set of features normalization algorithms define linear and non linear transforms in order to modify the noisy features statistics and make them equal to those of a reference set of clean data. The most relevant algorithms are:

- CMVN: Cepstral Mean ad Variance Normalization (Viiki et al., 1998)*:*
  The additive effect of noise implies a shift on the average of the MFCC coefficients probability density function added to a scaling of its variance. Given a noisy Cepstral coefficient y contaminated with an additive noise with mean value h, and given the clean Cepstral coefficient x with mean value μ_x and variance σ_x, the contaminated MFCC y will follow expression *(8)*, representing α the variance scaling produced:

$$\begin{aligned} y &= \alpha \cdot x + h \\ \mu_y &= \alpha \cdot \mu_x + h \\ \sigma_y &= \alpha \cdot \sigma_x \end{aligned} \tag{8}$$

  If we normalize the mean and variance of both coefficients x and y, their expressions will be:

$$\hat{x} = \frac{x - \mu_x}{\sigma_x}$$

$$\hat{y} = \frac{y - \mu_y}{\sigma_y} = \frac{(\alpha \cdot x + h) - (\alpha \cdot \mu_x + h)}{\alpha \cdot \sigma_x} = \hat{x} \tag{9}$$

  Equation *(9)* shows that CMVN makes the coefficients robust against the shift and scaling introduced by noise.

- **Higher order statistical moments normalization***:*
  A natural extension of CMVN is to normalize more statistical moments apart from the mean value and the variance. In 2004, Khademul (Khademul et al. 2004) adds the MFCCs first four statistical moments to the set of parameters to be used for automatic recognition obtaining some benefits in the recognition and making the system converge more quickly. Also in 2004 Chang Wen (Chang Wen & Lin Shan, 2004) proposes a normalization for the higher order Cepstral moments. His method permits the normalization of an eve or odd order moment added to the mean value normalization. Good results are obtained when normalizing moments with order higher than 50 in the original distribution. Prospection in this direction (Peinado & Segura J.C., 2006) is limited to the search of parametric approximations to normalize no more than 3 simultaneous statistical moments with a high computational cost that does not make them attractive when compared to the Histogram Equalization.
- **Histogram Equalization:**
  The linear transformation performed by CMNV only eliminates the linear effects of noise. The non-linear distortion produced by noise does not only affect the mean and variance of the probability density functions but it also affects the higher order moments. Histogram Equalization (De la Torre et al., 2005; Hilger & Ney, 2006) proposes generalizing the normalization to all the statistical moments by transforming

the Cepstral coefficients probability density function –*pdf*- in order to make it equal to a reference probability density function. The appeal of this technique is its low computational and storage cost, added to the absence of stereo data or any kind of supposition or model of noise. It is therefore a convenient technique to eliminate residual noise from other normalization techniques based on noise models like *VTS* (Segura et al., 2002). The objective of Section 3 will be to exhaustively analyze Histogram Equalization pointing at its advantages and limitations in order to overcome the last ones.

3. Histogram equalization

3.1 Histogram equalization philosophy

Histogram Equalization is a technique frequently used in Digital Image Processing (Gonzalez & Wintz, 1987; Russ, 1995) in order to improve the image contrast and brightness and to optimize the dynamic range of the grayscale. With a simple procedure it automatically corrects the images too bright, too dark or with not enough contrast. The gray level values are adjusted within a certain margin and the image's entropy is maximized.

Since 1998 and due to the work of Balchandran (Balchandran & Mammone, 1998), Histogram Equalization –*HEQ*- started to be used for robust voice processing. HEQ can be located within the family of statistical matching voice feature normalization techniques. The philosophy underneath its application to speech recognition is to transform the voice features both for train and test in order to make them match a common range. This *equalization* of the ranges of both the original emission used to train the recognizer and the parameters being evaluated, has the following effect: the automatic recognition system based on the Bayes classifier becomes ideally invulnerable to the linear and non linear transformations originated by additive Gaussian noise in the test parameters once those test parameters have been equalized. One condition must be accomplished for this equalization to work: the transformations to which the recognizer becomes invulnerable must be invertible.

In other words, recognition moves to a domain where any invertible transformation does not change the error of Bayes classifier. If CMN and CMNV normalized the mean and average of the Cepstral coefficients probability density functions, what HEQ does is normalizing the probability density function of the train and test parameters, transforming them to a third common *pdf* which becomes the *reference pdf*.

The base theory (De la Torre et al., 2005) for this normalization technique is the property of the random variables according to which, a random variable x with probability density function $p_x(x)$ and cumulative density function $C_x(x)$ can be transformed into a random variable $\hat{x} = T_x(x)$ with a reference probability density function $\phi_{\hat{x}}(x)$ preserving an identical cumulative density function ($C_x(x) = \Phi(\hat{x})$), as far as the transformation applied $T_x(x)$ is invertible (Peyton & Peebles, 1993). The fact of preserving the cumulative density function provides a univocal expression of the invertible transformation $T_x(x)$ to be applied to the transformed variable $\hat{x} = T_x(x)$ in order to obtain the desired probability density function $\phi_x(\hat{x})$:

$$\Phi(\hat{x}) = C_{\hat{x}}(x) = \Phi(T_x(x)) \tag{10}$$

$$\hat{x} = T_x(x) = \Phi_{\hat{x}}^{-1}(C_x(x)) \tag{11}$$

The transformation $T_x(x)$ defined in equation *(11)* is a non-decreasing monotonic function that will be non linear in general. Expression *(11)* shows that the transformation is defined using de *CDF* of the variable being transformed.

Once the random variables have been transformed, they become invulnerable to any linear or non-linear transformation applied to them as far as such transformation is reversible. Lets x be a random variable experimenting a generic reversible non linear transformation G to become the transformed random variable $y=G(x)$. If both original and transformed variables are equalized to a reference *pdf* ϕ_{ref}, the equalized variables will follow the expressions:

$$\hat{x} = T_x(x) = \Phi_{ref}^{-1}(C_x(x)) \tag{12}$$

$$\hat{y} = T_y(y) = \Phi_{ref}^{-1}(C_y(G(x))) \tag{13}$$

If G is an invertible function, then the *CDFs* of x and $y=G(x)$ will be equal:

$$C_x(x) = C_y(G(x)) \tag{14}$$

And in the same way, the transformed variables will also be equal:

$$\hat{x} = T_x(x) = \Phi_{ref}^{-1}(C_x(x)) = \Phi_{ref}^{-1}(C_y(G(x))) = \hat{y} \tag{15}$$

Expression *(15)* points out that if we work with equalized variables, the fact of them being subject to an invertible distortion does not affect nor training nor recognition. Their value remains identical in the equalized domain.

The benefits of this normalization method for robust speech recognition are based on the hypothesis that noise, denominated G in the former analysis, is an invertible transformation in the feature space. This is not exactly true. Noise is a random variable whose average effect can also be considered invertible (it can be seen in Figure 1). This average effect is the one that HEQ can eliminate.

HEQ was first used for voice recognition by Balchandran and Mammone (Balchandran & Mammone, 1998). In this first incursion of equalization in the field of speech, it was used to eliminate the non-linear distortions of the LPC Cepstrum of a speaker identification system. In 2000 Dharanipragada (Dharanipragada & Padmanabhan, 2000) used HEQ to eliminate the environmental mismatch between the headphones and the microphone of a speech recognition system. He added an adaptation step using non-supervised MLLR and obtained good results summing the benefits of both techniques. Since that moment, Histogram Equalization has been widely used and incorporated to voice front-ends in noisy environments. Molau, Hilger and Herman Ney apply it since 2001 (Molau et al., 2001; Hilger & Ney, 2006) in the Mel Filter Bank domain. They implement HEQ together with other

techniques like LDA –*Linear Discriminat Analysis*- or VTLN –*Vocal Track Length Normalization*- obtaining satisfactory recognition results. De la Torre and Segura (De la Torre et al., 2002; Segura et al., 2004; De la Torre et al., 2005) implement HEQ in the Cepstral domain and analyse its benefits when using it together with VTS normalization.

3.2 Equalization domain and reference distribution

3.2.1 Equalization domain

The parameterization used by practically the whole scientific community for voice recognition is the MFCC (Mel Frequency Cepstral Coefficients). These coefficients are obtained (see Figure 3) by moving the spectral analysis obtained at the end of a Mel Filter Bank to the domain of *quefrency*, defined as the Fourier inverse transform of the spectral logarithm (De la Torre et al., 2001). The quefrency domain is a temporal domain and the coefficients obtained in such domain are named Cepstral coefficients. They give results quite better than those obtained using the LPC Cepstrum and comparable to those obtained using auditive models without the high computational load of these last ones (Davis & Merlmenstein, 1980).

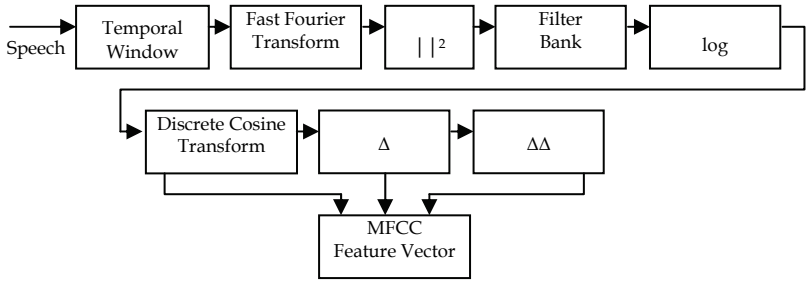


Fig. 3. Generation of MFCC coefficients.

Hilger and Molau apply the equalization after taking logarithms at the end of the Mel Filter Bank arguing that the logarithmic compression decreases the histograms discretization error. Their arguments for equalizing before going back to the quefrency time domain are the capability to compensate the distortions of certain specific frequencies with independent effects on certain components of the filter bank. Once the features are transformed to the quefrency domain, those distortions will be redistributed to all MFCCs via the lineal combination of the Mel filter bank outputs made by the Discrete Cosine Transform. This argument can be questioned as a strong correlation between the outputs of the filters within the Mel filter bank exists. An independent transformation in such a correlated domain does no seem the most appropriate.

The rest of authors using HEQ have obtained better results equalizing the MFCCs in the Cepstral Domain. Balchandran, Dharanipragada and De la Torre and Segura have made the equalization in the quefrency domain acting on the MFCC coefficients and their derivatives.

Finally, the feedback equalization technique used by Obuchi (Obuchi & Stern, 2003) must be mentioned when analyzing the equalization domain. He maintains that the temporal regression coefficients Δ and ΔΔ are not independent of the Cepstral coefficients and therefore he proposes to calculate those using the already equalized Cepstral coefficients and re-adjusting in an optimal way the static coefficients based on the Δ and ΔΔ calculated.

3.2.2 Reference distribution analysis

The election of the reference distribution Φ_{ref} used as common *CDF* to equalize the random variables is a relevant decision as the probability density function represents the global voice statistics. The analysis of equation *(10)* shows the relation between the original *pdf* and the reference *pdf* in the equalized domain:

$$p_x(x) = \frac{dC_x(x)}{dx} = \frac{d\Phi(T_x(x))}{dx} = \phi(T_x(x))\frac{dT_x(x)}{dx} = \phi(\hat{x})\frac{dT_x(x)}{dx} \tag{16}$$

Dharanipragada explains in (Dharanipragada & Padmanabhan, 2000) the relation that the original and reference *pdfs* must satisfy in terms of information. He uses the Kullback-Liebler distance as a measure of the existing mutual information between the original *pdf* and the equalized domain reference *pdf*:

$$D(\phi \mid p_x) = \int_{\hat{x}} \phi(\hat{x}) * \log(p_x(\hat{x})) * d\hat{x} \tag{17}$$

to conclude that such distance will become null in case the condition expressed in equation *(18)* is satisfied:

$$\phi(\hat{x}) = p_x(\hat{x}) \tag{18}$$

It is difficult to find a transformation $T_x(x)$ which satisfies equation *(18)* considering that x and $\hat{x}$ are random variables with dimension *N*. If the simplification of independency between the dimensions of the feature vector is accepted, equation *(18)* can be one-dimensionally searched for.

Two reference distributions have been used when implementing HEQ for speech recognition:

- **Gaussian distribution**: When using a Gaussian *pdf* as reference distribution, the process of equalization is called *Gaussianization*. It seems an intuitive distribution to be used in speech processing as the speech signal probability density function has a shape close to a bi-modal Gaussian. Chen and Gopinath (Chen S.S. and Gopinath R.A., 2000) proposed a Gaussianization transformation to model multi-dimensional data. Their transformation alternated linear transformations in order to obtain independence between the dimensions, with marginal one-dimensional *Gaussianizations* of those independent variables. This was the origin of Gaussianization as a probability distribution scaling technique which has been successfully applied by many authors (Xiang B. et al., 2002) (Saon G. et al., 2004), (Ouellet P. et al., 2005), (Pelecanos J. and Sridharan S., 2001), (De la Torre et al. 2001). Saon and Dharanipragada have pointed out the main advantage of its use: the most of the recognition systems use mixtures of Gaussians with diagonal covariance. It seems reasonable to expect that "*Gaussianizing*" the features will strengthen that assumption.
- **Clean Reference distribution:**
  The election of the training clean data probability density function (empirically built using cumulative histograms) as reference *pdf* for the equalization has given better results than Gaussianization (Molau et al., 2001) (Hilger & Ney, 2006) (Dharanipragada

& Padmanabhan, 2000). It can be seen as the non-parametrical version of Gaussianization in which the shape of the pdf is calculated empirically. The only condition needed is counting on enough data not to introduce bias or errors in the global voice statistic that it represents.

3.3 HEQ implementation

A computationally effective implementation of the Histogram Equalization can be done using quantiles to define the cumulative density function used. The algorithm is then called Quantile-Based Equalization –QBEQ- (Hilger & Ney, 2001) (Segura et al., 2004). Using this implementation, the equalization procedure for a sentence would be following one:

i. The sentence's order statistic is produced. If the total number of frames in the sentence is *2*T+1,* those *2*T+1* values will be ordered as equation (19) shows. The frame $x_{(r)}$ represents the frame with the *r-th* position within the ordered sequence of frames:

$$x_{(1)} \le x_{(2)} \ldots \le x_{(r)} \le \ldots \le x_{(2T+1)} \tag{19}$$

ii. The reference *CDF* set of quantiles are calculated. The number of quantiles per sample is chosen (N_Q). The CDF values for each quantile probability value p_r are registered:

$$Q_{\hat{x}}(p_r) = \Phi^{-1}(p_r) \tag{20}$$

$$p_r = (\frac{r-0,5}{N_Q}), \ \forall r = 1, \ldots, N_Q \tag{21}$$

iii. The quantiles of the original data will follow expression *(22)* in which *k* and *f* denote the integer and decimal part operators of *(1+2*Tp$_r$)* respectively:

$$Q_x(p_r) = \begin{cases} (1-f)x_k + fx_{k+1,} & 1 \le k \le 2*T \\ x_{(2T+1)} & k = 2T+1 \end{cases} \tag{22}$$

iv. Each pair of quantiles $(Q_x(p_r), Q_{\hat{x}}(p_r))$ represents a point of the equalization transformation that will be linearly approximated using the set of points obtained.

Figure 4. shows the results of implementing Histogram Equalization normalization using the QBEQ approximation, and performing the automatic speech recognition tasks for three databases: AURORA2, AURORA4 and HIWIRE:

- AURORA2: database created (Pearce & Hirsch, 2000) adding four different types of noise with 6 different SNRs to the clean database TIDigits (Leonard, 1984). It contains recording from adults pronouncing isolated and connected digits (up to seven) in English.
- AURORA4: Continuous speech database standardized (Hirsch, 2002) by the ETSI group STQ. It was built as a dictation task on texts from the Wall Street Journal with a size of 5000 words. It has 7 types of additive noises and convolutional channel noise to be put on top of them.

- HIWIRE Database (Segura et al., 2007): contains oral commands from the CPDLC (Controller Pilot Data Link Communications) communication system between the plane crew members and the air traffic controllers. The commands are pronounced in English by non-native speakers. Real noises recorded in the plane cockpit are added to the clean partitions.

Tests have been performed to compare the usage of two difference reference distributions. Equalization using a Gaussian distribution has been denoted as *HEQ-G* in the figure, while equalization using a clean reference probability density function (calculated using clean training data set) has been denoted as *HEQ-Ref Clean*. In order to have a wider vision of the effects of the equalization, two more tests have been performed. The one denoted as *Baseline* contains the results of evaluating the databases directly using the plane MFCCs. The test named *AFE* contains the results of implementing the ETSI Advanced Front End Standard parameterization (ETSI, 2002).

Comparative results seen in figure 4 show that better results are obtained when using clean reference distributions. The most evident case is the HIWIRE database. For this database, HEQ-G underperforms the Baseline parameterization results.

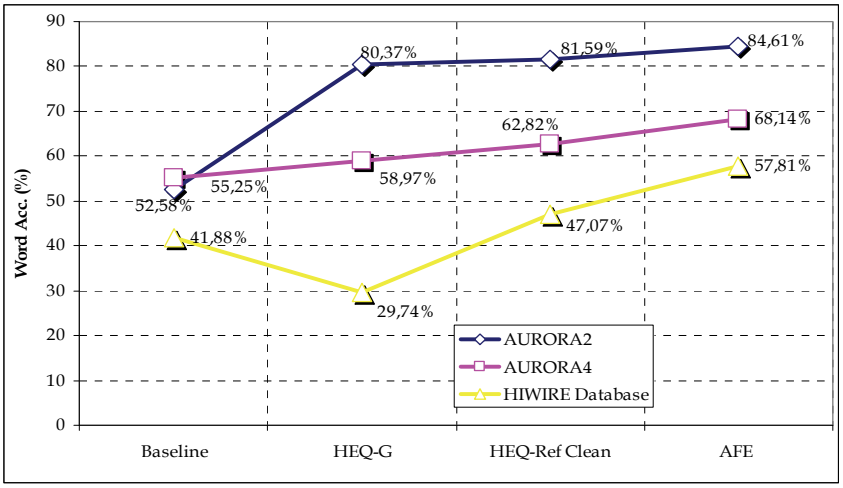


Fig. 4. HEQ compared to other normalization techniques.

3.4 Analysis of benefits and limitations

As a synthesis of the analysis of the HEQ done up to now, the following advantages of the algorithm can be enumerated:

- HEQ is implemented in the domain of the MFCC feature vector, and therefore it is independent of the recognizer back-end.
- It does not require a priori information about the type of noise or SNR expected during recognition. This fact makes the technique useful for noises with an unknown model or combinations of different types of noise.
- It is computationally un-expensive.
- It can be applied for real-time systems, dealing with commands applications or control for dialogue systems.

Nevertheless a series of limitations exist which justify the development of new versions of HEQ to eliminate them:

- The effectiveness of HEQ depends on the adequate calculation of the original and reference *CDFs* for the features to be equalized. There are some scenarios in which sentences are not long enough to provide enough data to obtain a trustable global speech statistic. The original *CDF* is therefore miscalculated and it incorporates an error transferred to the equalization transformation defined on the basis of this original *CDF*.
- HEQ works on the hypothesis of statistical independence of the MFCCs. This is not exactly correct. The real MFCCs covariance matrix is not diagonal although it is considered as such for computational viability reasons.

4. Parametric histogram equalization

4.1 Parametric histogram equalization philosophy

The two limitations of HEQ mentioned in section 3 have led to the proposal and analysis of a parametric version of Histogram Equalization (Garcia L. et al., 2006) to solve them. As we have just outlined in the former paragraph:

1. There is a minimum amount of data per sentence needed to correctly calculate statistics. This lack of data to generate representative statistics is also reflected in the following behaviour: the percentage of speech frames and silence frames contained in a sentence has a non desired influence on the calculated *CDF* and therefore on the transformation defined to equalize the sentence:

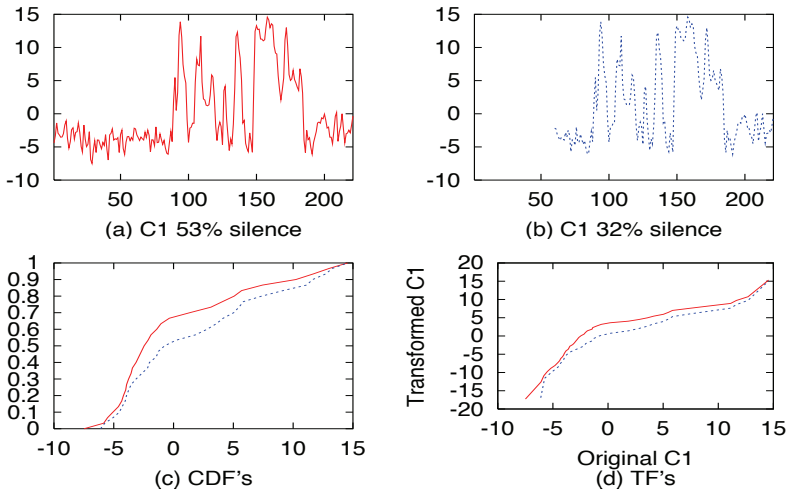


Fig. 5. Influence of silence percentage on the transformation

Figure 5 shows the effect of the silence percentage in the process of equalization. Subfigure (a) shows the value in time of Cepstral coefficient C_1 for a typical sentence. Subfigure (b) shows this same coefficient C_1 for the same sentence having removed part of the sentence's initial silence. Cumulative density functions for both sentences are shown in subfigure (c) where we can appreciate that even if both sentences have the same values for the speech frames, the different amount of silence frames alters the

shape of their global *CDF*. This difference in the *CDF* estimation introduces a non desired variation in the transformation calculated (see subfigure (d)).
The existing strategies to face the short sentences producing non representative statistics are mainly the usage of a parametric expression for the *CDF* (Molau et al., 2002; Haverinen & Kiss, 2003; Liu et al., 2004). The usage of order statistics (Segura et al., 2004) can also improve slightly the *CDF* estimation.

2. The second limitation of HEQ is that due to the fact that equalization is done independently for each MFCC vector component, all the information contained in the relation between components is being lost. It would be interesting to capture this information, and in case noise has produced a rotation in the feature space it would be convenient to recover from it. This limitation has originated a whole family of techniques to capture relations between coefficients, using vector quantization with different criteria, or defining classes via Gaussian Mixture Models (Olsen et al., 2003) (Visweswariah & Gopinath, 2002; Youngjoo et al., 2007). In the group of vector quantization we must mention (Martinez P. et al., 2007) that does an equalization followed by a vector quantization of the Cepstral coefficient in a 4D space, adding temporal information. (Dat T.H. et al.,2005) (Youngjoo S. and Hoirin K.) must also be mentioned.

As an effective alternative to eliminate the exposed limitations, the author of this chapter has proposed (Garcia et al., 2006) to use a parametric variant of the equalization transformation based in modelling the MFCCs probability density function with a mixture of two Gaussians. In ideal clean conditions, speech has a distribution very close to a bi-modal Gaussian. For this reason, Sirko Molau proposes in (Molau S. et al., 2002) the usage of two independent histograms for voice and silence. In order to do so, he separates frames as speech or silence using a Voice Activity Detector. Results are not as good as expected, as the discrimination between voice and silence is quite aggressive. Bo Liu proposes in (Bo L. et al., 2004) to use two Gaussian cumulative histograms to define the *pdf* of each Cepstral coefficient. He solves the distinction between the classes of speech or silence using a weighing factor calculated with each class probability.
The algorithm proposed in this work is named Parametric Equalization –*PEQ*-. It defines a parametric equalization transformation based on a two-Gaussians mixture model. The first Gaussian is used to represent the silence frames, and the second Gaussian is used to represent the speech frames. In order to map the clean and noisy domains, a parametric linear transformation is defined for each one of those two frame classes:

$$\hat{x} = \mu_{n,x} + (y - \mu_{n,y}) \cdot \left(\frac{\Sigma_{n,x}}{\Sigma_{n,y}}\right)^{1/2} \quad \text{being } y \text{ a silence frame} \tag{23}$$

$$\hat{x} = \mu_{s,x} + (y - \mu_{s,y}) \cdot \left(\frac{\Sigma_{s,x}}{\Sigma_{s,y}}\right)^{1/2} \quad \text{being } y \text{ a silence frame} \tag{24}$$

The terms of equations *(23)* and *(24)* are defined as follows:

- $\mu_{n,x}$ and $\Sigma_{n,x}$ are the mean and variance of the clean reference Gaussian distributions for the class of silence.

- $\mu_{s,x}$ and $\Sigma_{s,x}$ are the mean and variance of the clean reference Gaussian distributions for the class of speech.
- $\mu_{n,y}$ and $\Sigma_{n,y}$ correspond to the mean and variance of the noisy environment Gaussian distributions for the class of silence.
- $\mu_{s,y}$ and $\Sigma_{s,y}$ correspond to the mean and variance of the noisy environment Gaussian distributions for the class of speech.

Equations *(23)* and *(24)* transform the averages of the noisy environment $\mu_{n,y}$ and $\mu_{s,y}$ into clean reference averages $\mu_{n,x}$ and $\mu_{s,x}$. The noisy variances $\Sigma_{n,y}$ and $\Sigma_{s,y}$ are transformed into clean reference averages $\Sigma_{n,x}$ and $\Sigma_{s,x}$.

The clean reference Gaussian parameters are calculated using the data of the clean training set. The noisy environment Gaussian parameters are individually calculated for every sentence in process of equalization.

Before equalizing each frame we have to choose if it belongs to the speech or silence class. One possibility for taking this decision is to use a voice activity detector. That would imply binary election between both linear transformations (transformation according to voice class parameters or transformation according to silence class parameters). In the border between both classes taking a binary decision would create a discontinuity. In order to avoid it we have used a soft decision based on including the conditional probabilities of each frame to be speech or silence. Equation *(25)* shows the complete process of parametric equalization:

$$\hat{x} = P(n \mid y) \cdot (\mu_{n,x} + (y - \mu_{n,y}) \cdot (\frac{\Sigma_{n,x}}{\Sigma_{n,y}})^{1/2}) + P(s \mid y) \cdot (\mu_{s,x} + (y - \mu_{s,y})(\frac{\Sigma_{s,x}}{\Sigma_{s,y}})^{1/2}) \tag{25}$$

The terms $P(n \mid y)$ and $P(s \mid y)$ of equation *(25)* are the posterior probabilities of the frame belonging to the silence or speech class respectively. They have been obtained using a 2-class Gaussian classifier and the logarithmic energy term (Cepstral coefficient C_0) as classification threshold. Initially, the frames with a C_0 value lower than the C_0 average in the particular sentence are considered as noise. Those frames with a C_0 value higher than the sentence average are considered as speech. Using this initial classification the initial values of the means, variances and priori probabilities of the classes are estimated. Using the Expected Maximization Algorithm –*EM*-, those values are later iterated until they converge. This classification originates the values of $P(n \mid y)$ and $P(s \mid y)$ added to the mean and covariance matrixes for the silence and speech classes in the equalization process $\mu_{n,y}$, $\mu_{s,y}$, $\Sigma_{n,y}$ and $\Sigma_{s,y}$.

If we call n the number of silence frames in the sentence x and s the number of speech frames in the same sentence x being equalized, the mentioned parameters will be defined iteratively using *EM*:

$$n_n = \sum_x p(n \mid x) \cdot x$$

$$n_s = \sum_x p(s \mid x) \cdot x$$

$$\mu_n = \frac{1}{n_n} \cdot \sum_x p(n \mid x)$$

$$\mu_s = \frac{1}{n_s} \cdot \sum_x p(s \mid x)$$

$$\bar{\Sigma}_n = \frac{1}{n_n} \cdot \sum_x p(n \mid x) \cdot (x - \mu_n) \cdot (x - \mu_n)^T$$

$$\bar{\Sigma}_s = \frac{1}{n_s} \cdot \sum_x p(s \mid x) \cdot (x - \mu_s) \cdot (x - \mu_s)^T \quad (26)$$

The posterior probabilities used in *(26)* have been calculated using the Bayes rule:

$$p(n \mid x) = \frac{p(n) \cdot (N(x, \mu_n, \bar{\Sigma}_n))}{p(n) \cdot (N(x, \mu_n, \bar{\Sigma}_n)) + p(s) \cdot (N(x, \mu_s, \bar{\Sigma}_s))}$$

$$p(s \mid x) = \frac{p(s) \cdot (N(x, \mu_s, \bar{\Sigma}_s))}{p(n) \cdot (N(x, \mu_n, \bar{\Sigma}_n)) + p(s) \cdot (N(x, \mu_s, \bar{\Sigma}_s))} \quad (27)$$

Subfigures (a) and (b) from Figure 6 show the two-Gaussian parametric model for the probability density functions of Cepstral coeficients C_0 and C_1, put on top of the cumulative histograms of speech and silences frames for a set of clean sentences. Subfigures (c) and (d) show the same models and histograms for a set of noisy sentences.
The former figures show the convenience of using bi-modal Gaussians to approximate the two-class histogram, specially in the case of the coefficient C_0. They also show how the distance between both Gaussians or both class histograms decreases when the noise increases.

4.2 Histogram equalization versus parametric histogram equalization

The solid line of Figure 7 represents the transformation defined for a noisy sentence according to Parametric Equalization in two classes, PEQ. The dotted line of the graph represents the equalization transformation for the same sentence defined with HEQ. Parametric equalization is based on the class probabilities $P(n \mid y)$ and $P(s \mid y)$ which depend on the level of the Cepstral coefficient C_0. In the case of PEQ, equation *(25)* will define an equalized variable $\hat{x}$ as a non-linear function of y tending to the linear mapping given by:

- Equation *(24)* when the condition $P(s \mid y) >> P(n \mid y)$ is fulfilled.
- Equation *(23)* when the condition $P(n \mid y) >> P(s \mid y)$ is fulfilled.

The case of coefficient C_1 contains a interesting difference when working with PEQ: as $P(n \mid y)$ and $P(s \mid y)$ depend on the value of C_0, the relation between the clean and noisy data is not a monotonous function. A noisy value of C_1 can originate different values of C_1 equalized, depending on the value of C_0 for the frame.

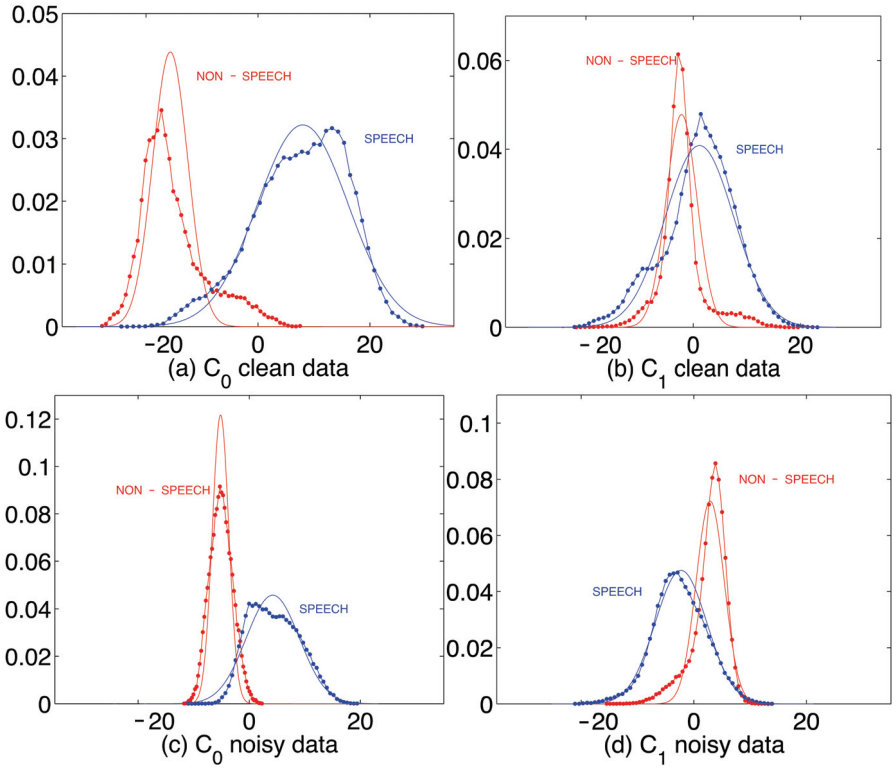


Fig. 6. Histogram versus two Gaussian parametric model.

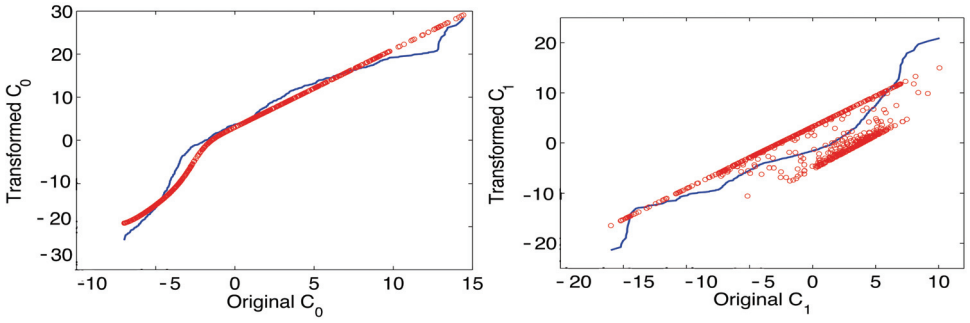


Fig. 7. HEQ transformation versus PEQ transformation

Figure 8 shows the comparative results of implementing HEQ and PEQ. An improvement is obtained for the three databases used in the experiments and described in the former section. The highest optimization is obtained for the HIWIRE database followed by AURORA4. AURORA2 obtains a lower improvement when using the parametric version of the equalization.

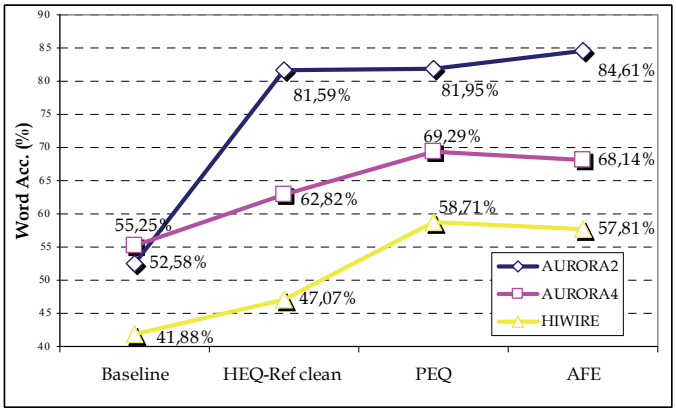


Fig. 8. Word accuracy in recognition for HEQ and PEQ.

5. Conclusions and future work

After analyzing the effects of additive noise on the speech recognition, this chapter has described the Histogram Equalization as the main representative of the statistical matching normalization strategies for automatic speech recognition. Its main attractive is the low computational cost added to the advantage of not needing any noise model or SNR hypothesis to work. When compared to other approximations within the same group of techniques, its peculiarity is the fact that it eliminates the non-linear distortions of noise. The main limitations of the technique are its dependency on the amount of data to work properly, and the waste of the correlations between the MFCC coefficients as an acoustic information source. A modified equalization technique denoted as Parametric Equalization has been presented in order to overcome those limitations and improve the recognition results. Comparative tests on 3 very different databases have been performed and presented showing an interesting improvement in the word accuracy results especially for the more complicated databases.

There are nevertheless some work lines open to improve the benefits of Histogram Equalization in robust speech recognition:

- Up to now and due to computational feasibility reasons, the MFCCs have been considered independent. Although it is small, a certain correlation between them exists. It would be desirable to capture such correlation.
- HEQ and PEQ (although PEQ does it to a lesser extent) introduce certain degradation if there is no noise distortion in the evaluation data. The reduction of such degradation is a challenge in order to use the algorithm in applications combining simultaneous noisy and clean evaluation environments.
- The concept of Histogram Equalization has been applied only for normalizing voice features. Its application as a Model Adaptation technique is under analysis.

6. Acknowledgements

This work has received research funding from the EU 6th Framework Programme, under contract number IST-2002-507943 (HIWIRE, Human Input that Works in Real

Environments) and SR3-VoIP projects (TEC2004-03829/TCM) from the Spanish government. The views expressed here are those of the authors only. The Community is not liable for any use that may be made of the information contained therein.

7. References

Bo L. Ling-Rong D., Jin-Lu L. and Ren-Hua W. (2004). Double Gaussian based feature normalization for robust speech recognition. Proc. of ICSLP'04, pages 253-246. 2004

Boll, S. F. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustic, Speech, Signal Processing.* ASSP-27. Nº2 . Pag 112-120, 1979.

Chang wen H. and Lin Shan L. (2004). Higher order cepstrla moment normalization (hocmn) for robust speech recognition. *Proc. of ICASSP'04,* pages 197-200. 2004.

Chen S.S. and Gopinath R.A. (2000). Gaussianization. *Proc. of NIPS 2000*. Denver, USA. 2000.

Dat T.H., Takeda K. and Itakura F.(2005). A speech enhancement system based on data clustering and cumulative histogram equalization. *Proc. of ICDE'05*. Japan. 2005.

Davis S.B. and Merlmenstein P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustic, Speech and Signal Processing*. ASSP-28, 4:357-365. 1980.

Dharanipragada S. and Padmanabhan M. (2000). A non supervised adaptation tehcnique for speech recognition. *Proc. of ICSLP 2000.* pages 556-559. China. 2000.

De la Torre A., Peinado A. and Rubio A. *Reconocimiento Automático de voz en condiciones de ruido*. Monografías del Depto. de Electrónica, nº 47. Universidad de Granada, Granada, España. 2001.

De la Torre A., Segura J.C., Benítez C., Peinado A., Rubio A. (2002). Non linear transformation of the feature space for robust speech recognition. *Proceedings of ICASSP 2002*. Orlando, USA, IEEE 2002.

De la Torre A., Peinado A., Segura J.C., Pérez Córdoba J.L., Benítez C., Rubio A. (2005). Histogram Equalization of speech representation for robust speech recognition. *IEEE Transactiosn on Speech and Audio Processing,* Vol. 13, nº3: 355-366. 2005

Deng L., Acero A., Plumpe M. and Huang X. (2000). Large vocabulary speech recognition under adverse acoustic environments. *Proceedings of ICSLP'00..* 2000.

Ephraim Y. and Malah D. (1985). Speech enhancement using a minimum mean square error log-spectral amplitude estimator. *IEEE Transactions on speech and audio processing,* Vol. 20, nº 33: 443-335. IEEE 1985.

ETSI ES 2002 050 v1.1.1 (2002). Speech processing, transmission and quality aspects; distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms. *Recommendation 2002-10.*

Furui S. (1981). Cepstral analysis technique for automatic speaker verification. *IEEE Transaction on Speech and Audio Processing* Vol. 29, nº 2: 254-272. 1981.

Gales M.J. and Woodland P.C. (1996). Mean and variance adaptation within the MLLR framework. *Computer Speech and Language.* Vol. 10: 249-264. 1996.

Gales M.J. and Young S., (1993). Cepstral parameter compensation for the update of the parameters of a single mixture density hmm recognition in noise. *Speech Communications.*Vol 12: 231-239. 1993.

García L., Segura J.C., Ramírez J., De la Torre A. and Benítez C. (2006). Parametric Non-Linear Features Equalization for Robust Speech Recognition. *Proc. of ICASSP'06*. France. 2006.

Gauvain J.L. and Lee C.H. (1994). Maximum a posteriori estimation for multivariate Gaussian mixture observation of Markov chains. *IEEE Transactions on speech and audio processing.* Vol. 2, nº291-298. 1994.

González R.C. and Wintz P. (1987). *Digital Image Processing*. Addison-Wesley. 1987.

Haverinen H. and Kiss I. (2003). On-line parametric histogram equalization techniques for noise robust embedded speech recognition. *Proc. of Eurospech'03*. Switzerland. 2003.

Hermansky H. and Morgan N. (1991). Rasta Processing of Speech. *IEEE Transactions on acoustic speech and signal processing.* Vol. 2, nº 4: 578-589. 1991.

Hilger F. and Ney H. (2006). Quantile based histogram equalization for noise robust large vocabulary speech recognition. *IEEE Transactions on speech and audio processing*. 2006.

Hirsch H.G. (2002). Experimental framework for the performance evaluation of speech recognition front-ends of large vocabulary tasks. STQ AURORA DSR Working Group. 2002.

Leonard R.G. (1984). A database for independent digits recognitions. *Proc. of ICASSP'84*. United States. 1984.

Lockwood P. and Boudy J. (1992). Experiments with a Non Linear Spectral Subtractor (NSS), Hidden Markov Models and the projection, for robust speech recognition in cars. *Speech Communications,* Vol. 11. Issue 2-3, 1992.

Martinez P., Segura J.C. and García L. (2007). Robust distributed speech recognitio using histogram equalization and correlation information. Proc. of Interspeech'07. Belgium. 2007.

Molau S., Pitz M. and Ney H. (2001). Histogram based normalization in the acoustic feature space. *Proc. of ASRU'01.* 2001.

Molau S., Hilger F., Keyser D. and Ney H. (2002). Enhanced histogram equalization in the acoustic feature space. Proc. of ICSLP'02. pages 1421-1424. 2002.

Moreno P.J., Raj B., Gouvea E. and Stern R. (1995). Multivariate-gaussian-based Cepstral normalization for robust speech recognition. *Proc. of ICASSP 1995.* p. 137-140. 1995

Moreno P.J., Raj B. and Stern R. (1986). A Vector Taylor Series approach for environment-independent speech recognition. *Proc. of ICASSP'96*. USA. 1996.

Obuchi Y. and Stern R. (2003). Normalization of time-derivative parameters using histogram equalization. *Proc. of EUROSPEECH'03*. Geneva, Switzerland. 2003.

Olsen P., Axelrod S., Visweswariah K and Gopinath R. (2003). Gaussian mixture modelling with volume preserving non-linear feature space transforms. *Proc. of ASRU'03*. 2003.

Oullet P., Boulianne G. and Kenny P.(2005). Flavours of Gaussian Warping. *Proc. of INTERSPEECH'05.*, pages 2957-2960. Lisboa, Portugal. 2005

Pearce O. and Hirsch H.G. (2000). The Aurora experimental framework for the performance evaluationo of speech recognition systems under noisy conditions. *Proc. of ICSLP'00*. China. 2000.

Peinado A. and Segura J.C. (2006). Robust Speech Recognition over Digital Channels. John Wiley , England. ISBN: 978-0-470-02400-3. 2006.

Pelecanos J. and Sridharan S., (2001). Feature Warping for robust speaker verification. *Proceeding of Speaker Odyssey 2001 Conference*. Greece. 2001

Peyton Z. and Peebles J.R. (1993). *Probability, Random Variables and Random Signal Principles*. Mac-Graw Hill. 1993.

Raj B., Seltser M. and Stern R. (2001). Robust Speech Recognition: the case for restoring missing feaures. *Proc. of CRAC'01*. pages 301-304. 2001.

Raj B., Seltser M. and Stern R. (2005). Missing Features Apporach in speech recognition. *IEEE Signal Processing Magazine*, pages 101-116. 2005.

Russ J.C.(1995). *The Image Processing HandBook*. BocaRatón. 1995.

Saon G., Dharanipragada S. and Povey D. (2004). Feature Space Gaussianization. *Proc. of ICASSP'04*, pages 329-332. Quèbec, Canada. 2004

Segura J.C., Benítez C., De Torre A, Dupont A. and Rubio A. (2002). VTS residual noise compensation. *Proc. of ICASSP'02,* pages 409-412 .2002.

Segura J.C., Benítez C., De la Torre A. and Rubio A. (2004). Cepstral domain segmental non-linear feature transformations for robust speech recognition. *IEEE Signal Processing Letters*, 11, nº 5: 517-520. 2004.

Segura J.C., Ehrette T., Potamianos A. and Fohr D. (2007). The HIWIRE database, a noisy and non-native English speech Corpus for cockpit Communications. http://www.hiwire.org. 2007.

Viiki O., Bye B. and Laurila K. (1998). A recursive feature vector normalization approach for robust speech recognition in noise. *Proceedings of ICASSP'98*. 1998.

Visweswariah K. and Gopinath R. (2002). Feature adaptation using projections of Gaussian posteriors. *Proc. of ICASSP'02*. 2002.

Wiener, N. (1949). *Extrapolation, Interpolation and Smoothing of temporary Time Series.* New York, Wiley ISBN: 0-262-73005.

Xiang B., Chaudhari U.V., Navratil J., Ramaswamhy G. and Gopinath R. A. (2002). Short time Gaussianization for robust speaker verification. *Proc. of ICASSP'2002,* pages 197-200. Florida, USA. 2002.

Young S. et al. *The HTK Book.* Microsoft Corporation & Cambridge University Engineering Department. 1995.

Younjoo S., Mikyong J. and Hoiring K. (2006). Class-Based Histogram Equalization for robust speech recognition. *ETRI Journal*, Volume 28, pages 502-505. August 2006.

Younjoo S., Mikyong J. and Hoiring K. (2007). Probabilistic Class Histogram Equalization for Robust Speech Recognition. Signal Processing Letters, Vol. 14, nº 4. 2007

Yuk D., Che L. and Jin L. (1996). Environment independent continuous speech recognition using neural networks and hidden markov models. *Proc. of ICASSP'96*. USA (1996).

Yukyz D. and Flanagany J. (1999). Telephone speech recognition using neural networks and Hidden Markov models. *Proceedings of ICASSP'99.* 1999.

3

Employment of Spectral Voicing Information for Speech and Speaker Recognition in Noisy Conditions

Peter Jančovič and Münevver Köküer
University of Birmingham
United Kingdom

1. Introduction

In this chapter, we describe our recent advances on representation and modelling of speech signals for automatic speech and speaker recognition in noisy conditions. The research is motivated by the need for improvements in these research areas in order the automatic speech and speaker recognition systems could be fully employed in real-world applications which operate often in noisy conditions.

Speech sounds are produced by passing a source-signal through a vocal-tract filter, i.e., different speech sounds are produced when a given vocal-tract filter is excited by different source-signals. In spite of this, the speech representation and modelling in current speech and speaker recognition systems typically include only the information about the vocal-tract filter, which is obtained by estimating the envelope of short-term spectra. The information about the source-signal used in producing speech may be characterised by a voicing character of a speech frame or individual frequency bands and the value of the fundamental frequency (F0). This chapter presents our recent research on estimation of the voicing information of speech spectra in the presence of noise and employment of this information into speech modelling and in missing-feature-based speech/speaker recognition system to improve noise robustness. The chapter is split into three parts.

The first part of the chapter introduces a novel method for estimation of the voicing information of speech spectrum. There have been several methods previously proposed to this problem. In (Griffin & Lim, 1988), the estimation is performed based on the closeness of fit between the original and synthetic spectrum representing harmonics of the fundamental frequency (F0). A similar measure is also used in (McAulay & Quatieri, 1990) to estimate the maximum frequency considered as voiced. In (McCree & Barnwell, 1995), the voicing information of a frequency region was estimated based on the normalised correlation of the time-domain signal around the F0 lag. The author in (Stylianou, 2001) estimates the voicing information of each spectral peak by using a procedure based on a comparison of magnitude values at spectral peaks within the F0 frequency range around the considered peak. The estimation of voicing information was not the primary aim of the above methods, and as such, no performance evaluation was provided. Moreover, the above methods did not consider speech corrupted by noise and required an estimation of the F0, which may be difficult to estimate accurately in noisy speech. Here, the presented method for estimation of

the spectral voicing information of speech does not require information about the F0 and is particularly applicable to speech pattern processing. The method is based on calculating a similarity, which we refer to as voicing-distance, between the shape of signal short-term spectrum and the spectrum of the frame-analysis window. To reflect filter-bank (FB) analysis that is typically employed in feature extraction for ASR, the voicing information associated with an FB channel is computed as an average of voicing-distances (within the channel) weighted by corresponding spectral magnitude values. Evaluation of the method is presented in terms of false-rejection and false-acceptance errors.

The second part of the chapter presents an employment of the estimated spectral voicing information within the speech and speaker recognition based on the missing-feature theory (MFT) for improving noise robustness. There have been several different approaches to improve robustness against noise. Assuming availability of some knowledge about the noise, such as spectral characteristics or stochastic model of noise, speech signal can be enhanced prior to its employment in the recogniser, e.g., (Boll, 1979; Vaseghi, 2005; Zou et al., 2007), or noise-compensation techniques can be applied in the feature or model domain to reduce the mismatch between the training and testing data, e.g., (Gales & Young, 1996). Recently, the missing feature theory (MFT) has been used for dealing with noise corruption in speech and speaker recognition, e.g., (Lippmann & Carlson, 1997; Cooke et al. 2001; Drygajlo &. El-Maliki, 1998). In this approach, the feature vector is split into a sub-vector of reliable and unreliable features (considering a binary reliability). The unreliable features are considered to be dominated by noise and thus their effect is eliminated during the recognition, for instance, by marginalising them out. The performance of the MFT method depends critically on the accuracy of the feature reliability estimation. The reliability of spectral-based features can be estimated based on measuring the local signal-to-noise ratio (SNR) (Drygajlo &. El-Maliki, 1998; Renevey & Drygajlo, 2000) or employing a separate classification system (Seltzer et al., 2004). We demonstrate that the employment of the spectral voicing information can play a significant role in the reliability estimation problem. Experimental evaluation is presented for MFT-based speech and speaker recognition and significant recognition accuracy improvements are demonstrated.

The third part of the chapter presents an incorporation of the spectral voicing information to improve speech signal modelling. Up to date, the spectral voicing information of speech has been mainly exploited in the context of speech coding and speech synthesis research. In speech/speaker recognition research, the authors in (Thomson & Chengalvarayan, 2002; Ljolje, 2002; Kitaoka et al., 2002; Zolnay et al., 2003; Graciarena et al., 2004) investigated the use of various measures for estimating the voicing-level of an entire speech frame and appended these voicing features into the feature representation. In addition to voicing features, the information on F0 was employed in (Ljolje, 2002; Kitaoka et al., 2002). In (Thomson & Chengalvarayan, 2002), the effect of including the voicing features under various training procedures was also studied. Experiments in the above papers were performed only on speech signals not corrupted by an additional noise and modest improvements have been reported. In (Jackson et al., 2003), the voicing information was included by decomposing speech signal into simultaneous periodic and aperiodic streams and weighting the contribution of each stream during the recognition. This method requires information about the fundamental frequency. Significant improvements on noisy speech recognition on Aurora 2 connected-digit database have been demonstrated, however, these results were achieved by using the F0 estimated from the clean speech. The authors in

(O'Shaughnessy & Tolba, 1999) divided phoneme-based models of speech into a subset of voiced and unvoiced models and used this division to restrict the Viterbi search during the recognition. The effect of such division of models itself was not presented. In (Jančovič & Ming, 2002) an HMM model was estimated based only on high-energy frames, which effectively corresponds to the voiced speech. This was observed to improve the performance in noisy conditions. The incorporation of the voicing information we present here differs from the above works in the following: i) the voicing information employed is estimated by a novel method that can provide this information for each filter-bank channel, while requiring no information about the F0; ii) the voicing-information is incorporated within an HMM-based statistical framework in the back-end of the ASR system; iii) the evaluation is performed on noisy speech recognition. In the proposed model, having the trained HMMs, with each mixture at each HMM state is associated a voicing-probability, which is estimated by a separate Viterbi-style training procedure (without altering the trained HMMs). The incorporation of the voicing-probability serves as a penalty during recognition for those mixtures/states whose voicing information does not correspond to the voicing information of the signal. The incorporation of the voicing information is evaluated in a standard model and in a missing-feature model that had compensated for the effect of noise. Experiments are performed on the Aurora 2 database. Experimental results show significant improvements in recognition performance in strong noisy conditions obtained by the models incorporating the voicing information.

2. Estimation of the voicing information of speech spectra

In this section we present a novel method for estimation of the voicing information of speech spectra which was introduced and analysed in (Jančovič & Köküer, 2007a).

2.1 Principle

Speech sounds are produced by passing a source-signal through a vocal-tract filter. The production of voiced speech sounds is associated with vibration of the vocal-folds. Due to this, the source-signal consists of periodic repetition of pulses and its spectrum approximates to a line spectrum consisting of the fundamental frequency and its multiples (referred to as harmonics). As a result of the short-time processing, the short-time Fourier spectrum of a voiced speech segment can be represented as a summation of scaled (in amplitude) and shifted (in frequency) versions of the Fourier transform of the frame-window function. The estimation of the voicing character of a frequency region can then be performed based on comparing the short-time magnitude spectrum of the signal to the spectrum of the frame-window function, which is the principle of the voicing estimation algorithm. Note that this algorithm does not require any information about the fundamental frequency; however, if this information is available it can be incorporated within the algorithm as indicated below.

2.2 Algorithm description

Below are the steps of the algorithm:

1. Short-time magnitude-spectrum calculation:

A frame of a time-domain signal is weighted by a frame-analysis window function, expanded by zeros and the FFT is applied to provide a short-time magnitude-spectrum.

Throughout the chapter we work with signals sampled at Fs=8 kHz, the frame length of 256 samples and the FFT-size of 512 samples.

2. Voicing-distance calculation:

For each peak of the short-time signal magnitude-spectrum, a distance, referred to as voicing-distance $vd(k)$, between the signal spectrum around the peak and spectrum of the frame window is computed, i.e.,

$$\mathrm{vd}(\mathrm{k_p}) = \left[\frac{1}{2\mathrm{L}+1} \sum_{\mathrm{k}=-\mathrm{L}}^{\mathrm{L}} \left(\left|\mathrm{S}(\mathrm{k_p}+\mathrm{k})\right| - \left|\mathrm{W}(\mathrm{k})\right| \right)^2 \right]^{1/2} \quad (1)$$

where k_p is the frequency-index of a spectral peak and L determines the number of components of the spectra at each side around the peak to be compared (the value 2 was used). The spectrum of the signal, $S(k)$, and frame-window, $W(k)$, are normalised to have magnitude value equal to 1 at the peak prior to their use in Eq. 1. The voicing-distances for frequency components around the peak were set to the voicing-distance at the peak, i.e., $vd(k)= vd(k_p)$ for $k \in [k_p\text{-}L, k_p+L]$. Note that if the information about the fundamental frequency is available, the voicing-distance could be calculated at frequency-indices corresponding to multiples of the fundamental frequency instead of the peaks of the spectrum. Also note that the estimate of the fundamental frequency could be obtained based on the minimum cumulated voicing-distance calculated at multiples of considered fundamental frequency values (Jančovič & Köküer, 2007a).

3. Voicing-distance calculation for filter-bank channels:

The voicing-distance for each filter-bank (FB) channel is calculated as a weighted average of the voicing-distances within the channel, reflecting the calculation of filter-bank energies that are used to derive features for recognition, i.e.,

$$\mathrm{vd}^{\mathrm{fb}}(\mathrm{b}) = \frac{1}{\mathrm{X(b)}} \sum_{\mathrm{k}=\mathrm{k_b}}^{\mathrm{k_b}+\mathrm{N_b}-1} \mathrm{vd(k)} \cdot \mathrm{G_b(k)} \cdot \left|\mathrm{S(k)}\right|^2 \quad \text{where } \mathrm{X(b)} = \sum_{\mathrm{k}=\mathrm{k_b}}^{\mathrm{k_b}+\mathrm{N_b}-1} \mathrm{G_b(k)} \cdot \left|\mathrm{S(k)}\right|^2 \quad (2)$$

where $G_b(k)$ is the frequency-response of the filter-bank channel b, and k_b and N_b are the lowest frequency-component and number of components of the frequency response, respectively, and $X(b)$ is the overall filter-bank energy value.

4. Post-processing of the voicing-distances:

The voicing-distance obtained from steps (2) and (3) may accidentally become of a low value for an unvoiced region or vice versa. To reduce these errors, we have filtered the voicing-distances by employing 2-D median filters due to their effectiveness in eliminating outliers and simplicity. In our set-up, median filters of size 5x9 and 3x3 (the first number being the number of frames and the second the number of frequency indices) were used to filter the voicing-distances $vd(k)$ and $vd^{fb}(b)$, respectively.

Examples of spectrograms of noisy speech and the corresponding voicing-distances for spectrum and filter-bank channels are depicted in Figure 1.

2.3 Experimental evaluation

This section presents evaluation of the voicing estimation algorithm in terms of false-acceptance (FA) and false-rejection (FR) errors.

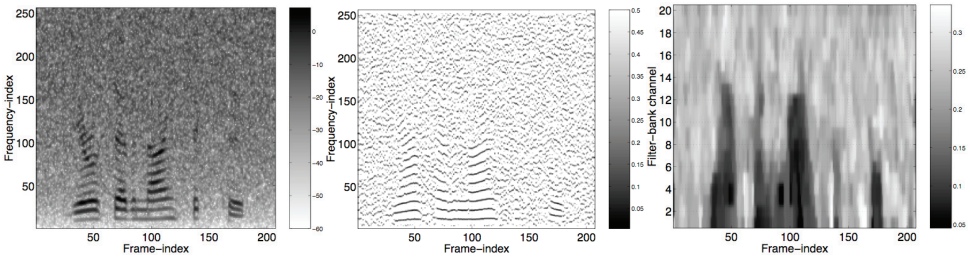


Fig. 1. Spectrogram (left), voicing-distance in the frequency-domain (middle), and in the filter-bank domain (right) of a speech utterance corrupted by white noise at SNR=5dB.

As the true information about the voicing information of FB channels is not available, it is defined based on *a priori* knowledge of clean speech signal and noise; this will be referred to as the "oracle" voicing label. Based on the analysis presented in (Jančovič & Köküer, 2007a), an FB channel of noisy speech is assigned oracle label *voiced* if its corresponding voicing-distance on clean speech is below 0.18 and its local-SNR is above 0 dB, and *unvoiced* otherwise. The decision on the voicing information of an FB channel is made based on a comparison of its corresponding voicing-distance value to a threshold. The experiments were carried out on 2486 digit utterances corrupted by white noise.

The experimental results in terms of FA and FR errors are depicted as a function of the local-SNR in Figure 2. It can be seen that the voicing information can be estimated with a good accuracy; for instance, the FA and FR errors are below 5% when speech signal is corrupted at 10 dB local-SNR and the voicing-distance threshold of 0.21 is used.

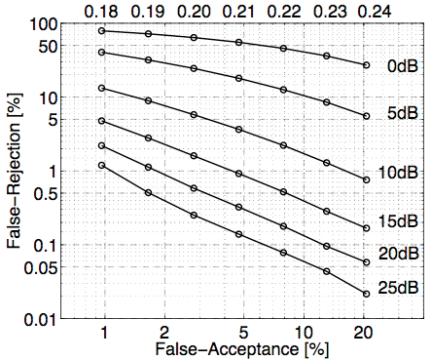


Fig. 2. Performance of the algorithm for estimation of the voicing information of FB channels in terms of FA and FR errors on speech corrupted by white noise. Results presented as a function of the local-SNR and the voicing-distance threshold (depicted above the figure).

3. Employment of the voicing information in the missing-feature-based speech/speaker recognition

The missing-feature theory (MFT) has been successfully employed to improve noise-robustness in automatic speech and speaker recognition systems, e.g., (Cooke, 2001 ; Drygajlo &. El-Maliki, 1998). This approach considers that (in recognition) the feature vector can be split into elements that are not affected (or affected only little) by noise, referred to as

reliable, and elements that are dominated by noise, referred to as *unreliable*. This information is stored in the so-called mask. The unreliable elements are then eliminated during the recognition. In this section we demonstrate that the voicing information of filter-bank channels can provide vital information for estimation of mask employed in MFT-based speech and speaker recognition systems.

3.1 Automatic speech/speaker recognition based on the missing-feature theory

Let $\mathbf{y}_t=(y_t(1), ..., y_t(B))$ denote the feature vector representing the t^{th} frame of the signal and $\mathbf{m}_t=(m_t(1), ..., m_t(B))$ be the corresponding mask vector determining whether an element of $\mathbf{y}_t$ is reliable (equal to 1) or unreliable (equal to 0). Let us consider that speech is modelled by a Hidden Markov Model (HMM) with state output probabilities modelled by a mixture of Gaussian distributions with diagonal covariance matrices. In the marginalisation-based MFT, the observation probability $P(\mathbf{y}_t \mid s,j)$ of the feature vector $\mathbf{y}_t$ at state s and mixture component j is calculated by integrating out the unreliable elements of the feature vector $\mathbf{y}_t$

$$P\left(\mathbf{y}_t \middle| s, j\right) = \prod_{b\in rel} P\left(y_t(b) \middle| s, j\right) \prod_{b\in unrel} \int P\left(y_t(b) \middle| s, j\right) = \prod_{b\in rel} P\left(y_t(b) \middle| s, j\right) \tag{3}$$

We also employed the MFT model within a GMM-based speaker recognition system; Eq. (3) applies also in this case as a GMM can be seen as a 1-state HMM.

In order to apply the MFT marginalisation model, the noise-corruption needs to be localised into a subset of features. This makes the standard full-band cepstral coefficients (MFCCs) (Davis & Mermelstein, 1980), which are currently the most widely used parameterisation of speech, unsuitable as the application of DCT over the entire vector of logarithm filter-bank energies (logFBEs) will cause any corruption localised in the frequency-domain to become spread over all cepstral coefficients. The logFBEs may be employed in the MFT model, however, they suffer from a high correlation between the features, which makes the diagonal covariance matrix modelling not appropriate. The parameterisations often used in MFT model are the sub-band cepstral coefficients, e.g., (Bourlard & Dupont, 1996), and frequency-filtered logFBEs (FF-logFBEs), e.g., (Jančovič & Ming, 2001). The FF-logFBEs, which are obtained by applying a (short) FIR filter over the frequency dimension of the logFBEs, were employed in this paper. These features in standard recognition system have been shown to obtain similar performance as the standard full-band cepstral coefficients (Nadeu et al., 2001), while having the advantage of retaining the noise-corruption localised.

3.2 Mask estimation for the MFT-based ASR

The performance of the MFT-based recognition system depends critically on the quality of the mask. The mask estimation is a complex task and the design of a method for this task is not of our main focus here, rather, we aim to demonstrate that the voicing estimation method can provide useful information to be employed for this task. As such, we are here concerned only with mask estimation for speech detected as voiced in clean data.

We have demonstrated in (Jančovič & Köküer, 2007a) that the voicing-distance $vd^{fb}(b)$ is related to the local-SNR of a voiced FB-channel corrupted by noise. Based on this, the voicing-distance can be used to define the *voicing mask* as

$$m_t^{voic}(b) = 1 \qquad \text{if} \qquad vd_t^{fb}(b) < \beta \tag{4}$$

where the threshold β was set to 0.21. In order to evaluate the quality of the estimated voicing mask, i.e., the effect of errors in the voicing information estimation on the recognition performance, we defined *oracle voicing mask* for an FB-channel as 1 if and only if the channel is estimated as voiced on clean data and its oracle mask (defined as below) is 1 on noisy data.

The so-called *oracle mask* is derived based on full a-priori knowledge of the noise and clean speech signal. The use of this mask gives an upper bound performance and thus it indicates the quality of any estimated mask. We used a-priori SNR to construct the oracle mask as

$$m_t^{\text{oracle}}(b) = 1 \qquad \text{if} \qquad 10\log(X_t(b)/N_t(b)) > \gamma \tag{5}$$

where $X_t(b)$ and $N_t(b)$ are the filter bank energy of clean speech and noise, respectively. The threshold γ was set to -6dB as it was shown to provide a good performance in our experiments.

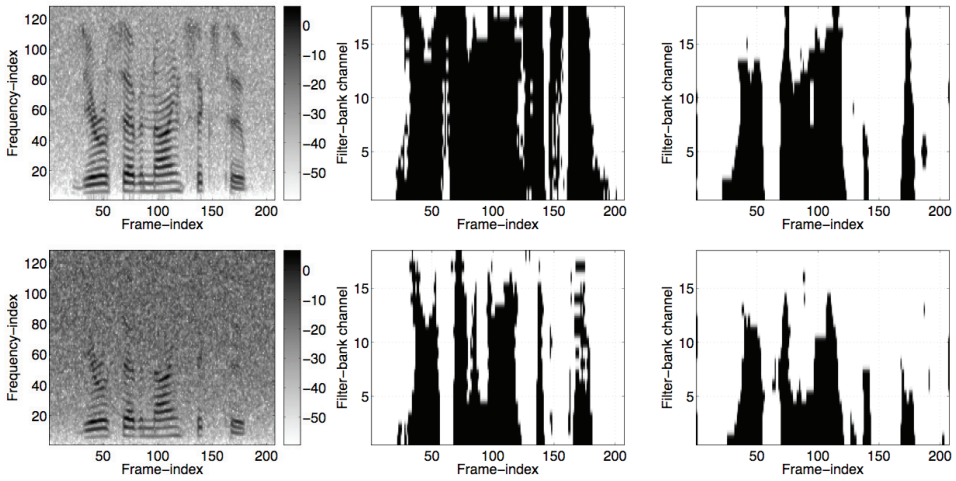


Fig. 3. Spectrograms (a), oracle masks (b), and voicing masks (c) for a speech utterance corrupted by white noise at the SNR=20dB (upper) and SNR=5dB (lower). Mask value equal to 1 and 0 depicted in black and white, respectively.

Figure 3 depicts spectrograms of a speech utterance corrupted by noise at two SNRs and the corresponding oracle and voicing masks. It can be seen that when the noise is strong, the voicing mask is relatively similar to the oracle mask as it is mainly the voiced regions (that are of a higher energy) that are affected only little by noise.

3.3 Experimental evaluations on speech recognition

Experimental evaluations were performed on the Aurora 2 English language database (Hirsch & Pearce, 2000). This database was designed for speaker-independent recognition of digit sequences in noisy conditions. The test set A from the database was used for recognition experiments. This test set consists of four sub-sets, each sub-set contains 1001 utterances of clean speech and noisy speech created by artificially adding to clean speech

one of four environmental noise types: subway, babble, car, and exhibition hall, each of these at six different SNRs: 20, 15, 10, 5, 0, and -5 dB. The clean speech training set, containing of 8440 utterances of 55 male and 55 female adult speakers, was used for training the parameters of HMMs.

The frequency-filtered logarithm filter-bank energies (FF-logFBEs) (Nadeu et al., 2001) were used as speech feature representation, due to their suitability for MFT-based recognition as discussed earlier. Note that the FF-logFBEs achieve similar performance (in average) as standard MFCCs. The FF-logFBEs were obtained with the following parameter set-up: frames of 32 ms length with a shift of 10 ms between frames were used; both preemphasis and Hamming window were applied to each frame; the short-time magnitude spectra, obtained by applying the FFT, was passed to Mel-spaced filter-bank analysis with 20 channels; the obtained logarithm filter-bank energies were filtered by using the filter $H(z)=z-z^{-1}$. A feature vector consisting of 18 elements was obtained (the edge values were excluded). In order to include dynamic spectral information, the first-order delta parameters were added to the static FF-feature vector.

The HMMs were trained following the procedures distributed with the Aurora 2 database. Each digit was modelled by a continuous-observation left-to-right HMM with 16 states (no skip allowed) and three Gaussian mixture components with diagonal covariance matrices for each state. During recognition, the MFT-based system marginalised static features according to the mask employed, and used all the delta features.

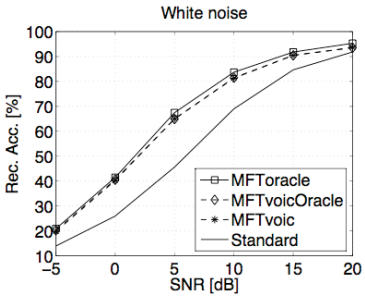


Fig. 4. Recognition accuracy results obtained by the MFT-based speech recognition system employing the voicing and oracle masks.

Experimental results are first presented in Figure 4 for speech corrupted by white noise (as this noise is considered to contain no voiced components) to evaluate the quality of the estimated voicing mask. It can be observed that the recognition accuracies achieved by the MFT-based recogniser employing the estimated voicing mask (MFTvoic) and the oracle voicing mask (MFTvoicOracle) are nearly identical. This indicates that the errors made in the voicing estimation have nearly no effect on the recognition performance of the MFT-based system in the given recognition task. It can also be seen that the MFT-based system using the voicing mask provides recognition performance significantly higher than that of the standard recognition system and indeed very close to using the oracle mask (MFToracle), i.e., the abandonment of uncorrupted unvoiced features did not harm significantly the recognition accuracy in the given task.

Results of experiments on the Aurora 2 noisy speech data are presented in Figure 5. It can be seen that employing the voicing mask in the MFT-based system provides significant

performance improvements over the standard system for most of the noisy conditions. The performance is similar (or lower) than that of the standard system at 20 dB SNR which is due to eliminating the uncorrupted unvoiced features. The MFT-based system employing the estimated voicing mask achieves less improvement in the case of Babble noise because this noise contain voiced components and thus the voicing mask captures the location of both the voiced speech regions as well as the voiced noise regions. There may be various ways to deal with the voiced noise situation. For instance, a simple way may be to consider that the speech is of a higher energy than noise and as such use only the higher energetic voiced regions. Also, signal separation algorithms may be employed, for instance, we have demonstrated in (Jančovič & Köküer, 2007b) that the sinusoidal model can be successfully used to separate two harmonic or speech signals.

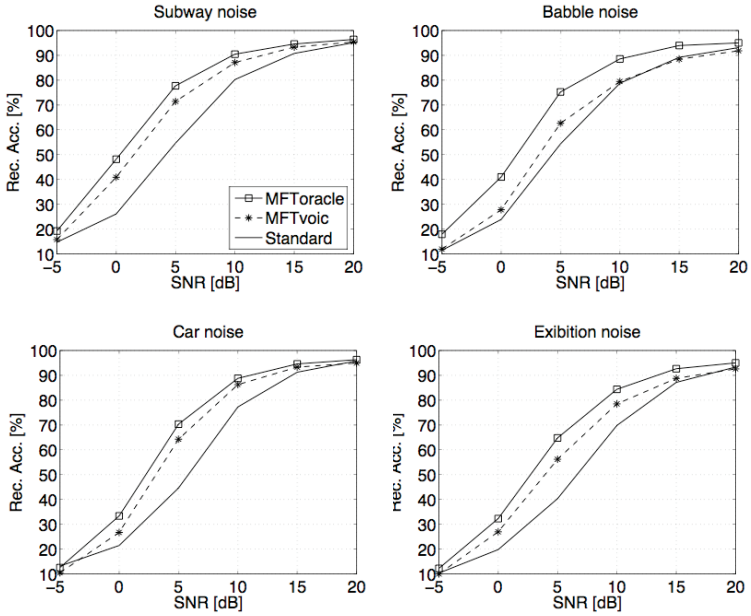


Fig. 5. Recognition accuracy results obtained by the MFT-based speech recognition system employing the voicing and oracle masks.

3.4 Experimental evaluations on speaker recognition

Experiments were performed on the TIMIT database (Garofolo et al., 1993), down sampled to 8 kHz. Hundred speakers (consisting of 64 male and 36 female) from the test subset were selected in an alphabetical order. The training data for each speaker comprised of eight sentences ('si' and 'sx'). The testing was performed using two ('sa') sentences corrupted by Gaussian white noise and Subway noise from the Aurora 2 database, at global SNRs equal to 20, 15, 10, and 5 dB, respectively. The speech feature representation was the same as used in the speech recognition experiments in the previous section. The speaker recognition system was based on Gaussian mixture model (GMM) with 32 mixture-components for each speaker, which was constructed using the HTK software (Young et al., 1999). The GMM for

each speaker was obtained by using the MAP adaptation of a general speech model, which was obtained from the training data from all speakers.
Experimental results are depicted in Figure 6. We can see that the results are of a similar trend as in the case of speech recognition. The use of the estimated voicing mask gives results close to those obtained using the oracle voicing mask. These results are significantly higher than those of the standard model and reasonably close those obtained by the MFT model using the oracle mask which assumes full a-priori knowledge of the noise. These results therefore demonstrate the effectiveness of the estimated voicing mask.

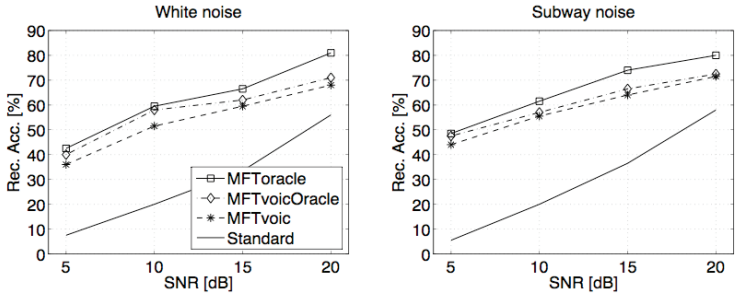


Fig. 6. Recognition accuracy results obtained by the MFT-based speaker recognition system employing the voicing and oracle masks.

4. Modelling the voicing information in the HMM-based ASR

This section presents an employment of the voicing information in an HMM-based ASR system in order to model the properties of the source-signal used to produce the speech. The modelling of the voicing information could be performed by appending the voicing features into the spectral feature vector and using the standard HMM training on the extended feature representation. We have here adopted an alternative approach, in which a separate training procedure for estimation of the voicing models is performed after the HMMs were trained on the spectral features. The presented method was introduced in (Jančovič & Köküer, 2007c).

4.1 Incorporation of the voicing information during recognition

Let $\mathbf{y}_t=(\mathbf{y}_t(1), ..., \mathbf{y}_t(B))$ denote the spectral-feature vector and $\mathbf{v}_t=(\mathbf{v}_t(1), ..., \mathbf{v}_t(B))$ the corresponding voicing vector at the frame-time t, where B is the number of FB channels.
During the recognition, the standard HMM state emission probability of a spectral-feature vector $\mathbf{y}_t$ at frame-time t in state s, i.e., $P(\mathbf{y}_t \mid s)$, is replaced by calculating the joint probability of the spectral feature vector and the voicing vector $\mathbf{v}_t$, i.e., $P(\mathbf{y}_t,\mathbf{v}_t \mid s)$. Considering that all spectral features and voicing features are independent of one another, using J mixture densities the $P(\mathbf{y}_t,\mathbf{v}_t \mid s)$ is calculated in the proposed model as

$$P\left(\mathbf{y}_t, \mathbf{v}_t \middle| s\right) = \sum_{j=1}^{J} P\left(j \middle| s\right) \prod_{b} P\left(y_t(b) \middle| s, j\right) P\left(v_t(b) \middle| s, j\right) \tag{6}$$

where $P(j \mid s)$ is the weight of the j^{th} mixture component, and $P(y_t(b) \mid s,j)$ and $P(v_t(b) \mid s,j)$ are the probability of the b^{th} spectral feature and voicing feature, respectively, given state s and

mixture j. Note that the voicing-probability term in Eq.(6) was used only when the feature was detected as voiced, i.e., the term was marginalised for features detected as unvoiced.

4.2 Estimation of the voicing-probability for HMM states

The estimation of the voicing-probability $P(\mathbf{v}\,|\,s,j)$ at each HMM state s and mixture j can be performed using the training data-set by Baum-Welch or Viterbi-style training procedure; the latter was used here.

Given a speech utterance, for each frame t we have the spectral-feature vector $\mathbf{y}_t$ and corresponding voicing vector $\mathbf{v}_t$, resulting a sequence of $\{(\mathbf{y}_1,\mathbf{v}_1), \ldots, (\mathbf{y}_T,\mathbf{v}_T)\}$. The Viterbi algorithm is then used to obtain the state-time alignment of the sequence of feature vectors $\{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$ on the HMMs corresponding to the speech utterance. This provides an association of each feature vector $\mathbf{y}_t$ to some HMM state s. The posterior probability that the mixture-component j (at the state s) have generated the feature vector $\mathbf{y}_t$ is then calculated as

$$\mathrm{P}\left(\mathrm{j}\middle|\mathbf{y}_\mathrm{t},\mathrm{s}\right)=\frac{\mathrm{P}\left(\mathrm{y_t}\middle|\mathrm{s},\mathrm{j}\right)\,\mathrm{P}\left(\mathrm{j}\middle|\mathrm{s}\right)}{\sum_{\mathrm{j'}}\mathrm{P}\left(\mathrm{y_t}\middle|\mathrm{s},\mathrm{j'}\right)\,\mathrm{P}\left(\mathrm{j'}\middle|\mathrm{s}\right)} \tag{7}$$

where the mixture-weight $P(j\,|\,s)$ and the probability density function of the spectral features used to calculate the $P(\mathbf{y}_t\,|\,s,j)$ are obtained as an outcome of the standard HMM training. For each mixture j and HMM state s, the posterior probabilities $P(j\,|\,\mathbf{y}_t,s)$ for all $\mathbf{y}_t$'s associated with the state s are collected (over the entire training data-set) together with the corresponding voicing vectors $\mathbf{v}_t$'s. The voicing-probability of the b^{th} feature can then be obtained as

$$\mathrm{P}\left(\mathrm{v(b)}=\mathrm{a}\middle|\mathrm{s},\mathrm{j}\right)=\frac{\sum_{\mathrm{t:y_t}\in\mathrm{s}}\mathrm{P}\left(\mathrm{j}\middle|\mathrm{y_t},\mathrm{s}\right)\cdot\delta\left(\mathrm{v_t(b)},\mathrm{a}\right)}{\sum_{\mathrm{t:y_t}\in\mathrm{s}}\mathrm{P}\left(\mathrm{j}\middle|\mathrm{y_t},\mathrm{s}\right)} \tag{8}$$

where $a\in\{0,1\}$ is the value of voicing information and $\delta(v_t(b),a)=1$ when $v_t(b)=a$, otherwise zero.

Examples of the estimated voicing-probabilities for HMMs of digits are depicted in Figure 7. It can be seen that, for instance, the first five states of the word 'seven' have the probability of being voiced close to zero over the entire frequency range, which is likely to correspond to the unvoiced phoneme /s/.

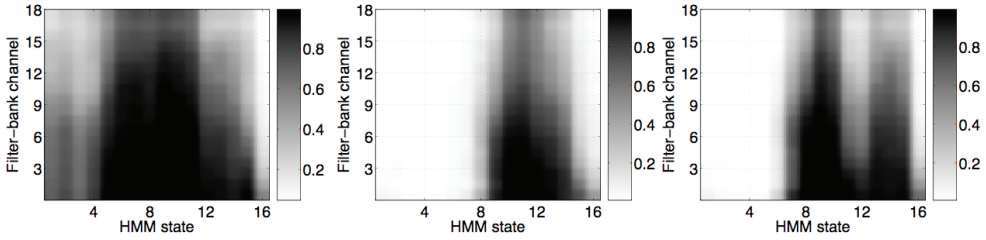


Fig. 7. Examples of the estimated voicing-probabilities for a 16 state HMM models of words 'one' (left), 'two' (middle), and 'seven' (right).

The estimated voicing-probability $P(v(b)\,|\,s,j)$ becomes zero when all features associated with the state are only voiced or unvoiced. This is not desirable, because it can cause the overall probability in Eq.(6) to become zero during the recognition. This could be avoided by setting a small minimum value for $P(v(b)\,|\,s,j)$. A more elegant solution that would also allow us to easily control the effect of the voicing-probability on the overall probability may be to employ a sigmoid function to transform the $P(v(b)\,|\,s,j)$ for each b to a new value, i.e.,

$$P\left(v(b)\middle|s,j\right)=\frac{1}{1+e^{-\alpha\left(P(v(b)|s,j)-0.5\right)}} \tag{9}$$

where α is a constant defining the slope of the function and the value 0.5 gives the shift of the function. Examples of the voicing-probability transformation with various values for α are depicted in Figure 8. The bigger the value of α is the greater the effect of the voicing-probability on the overall probability. An appropriate value for α can be decided based on a small set of experiments on a development data. The value 1.5 is used for all experiments.

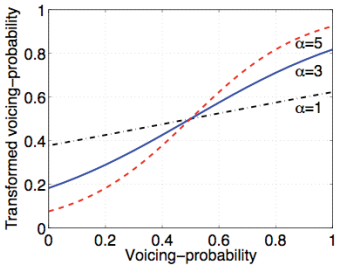


Fig. 8. Sigmoid function with various values of the slope parameter α employed for transformation of the voicing-probability.

4.3 Experimental evaluation

The experiments were performed on the Aurora 2 database. The experimental set-up is the same as described in Section 3.3.

The evaluation of the proposed model for voicing incorporation is first performed using a standard model trained on clean data. Results are presented in Figure 9. It can be seen that the incorporation of the voicing-probability provides significant recognition accuracy improvements at low SNRs in all noisy conditions. It was observed that the voicing-probability incorporation caused an increase of insertions in the case of Babble noise, which is due to this noise being of a voiced character. This could be improved by employing a speech-of-interest detection similar as discussed earlier in Section 3.3.

Next, evaluations were performed on a model that had compensated for the effect of noise – these experiments were conducted in order to determine whether the incorporation of the voicing information could still provide improvements (as employment of noise compensation would effectively decrease the amount of misalignment of the voicing information between the signal and models). For this, the marginalisation-based MFT model was used. In order to obtain the best (idealised) noise compensation, this model employs the oracle mask, obtained based on the full a-priori knowledge of the noise. Experimental results are presented in Figure 10. It can be seen that the incorporation of the voicing-probability did not improve the performance at high SNRs, which may be due to the effectiveness of the noise-compensation. The decrease at high SNRs in the case of Babble and Exhibition noise is, similarly as in the

standard model discussed earlier, due to the voiced character of the noise. It can be seen that the incorporation of the voicing-probability provides significant recognition accuracy improvements at low SNRs, even the noise effect had already been compensated.

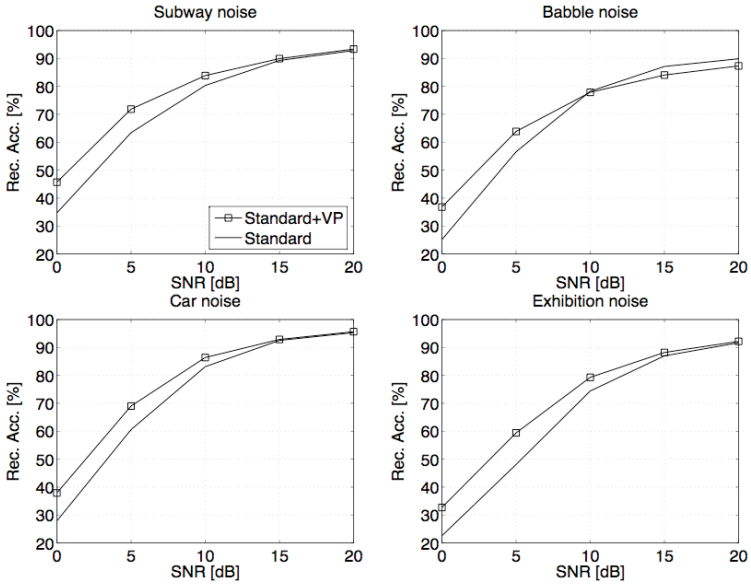


Fig. 9. Recognition accuracy results obtained by the standard ASR system without and with incorporating the voicing-probability.

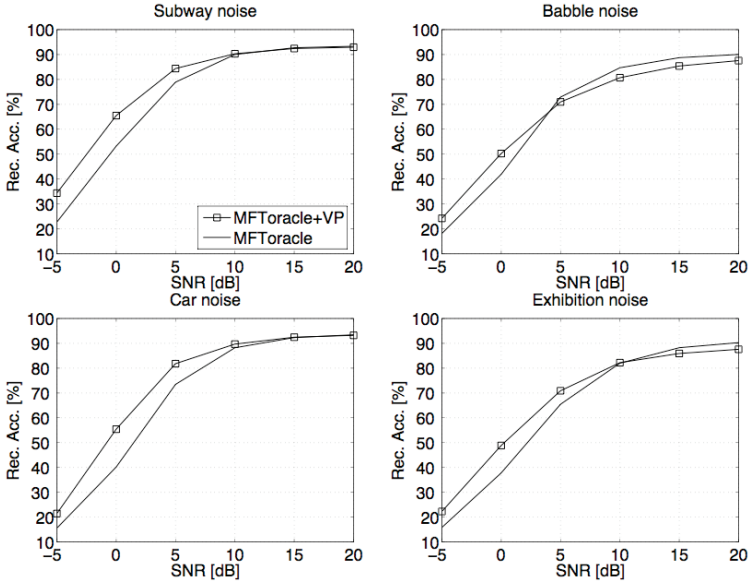


Fig. 10. Recognition accuracy results obtained by the MFT-based ASR system employing the oracle mask without and with incorporating the voicing-probability.

7. Conclusion

This chapter described our recent research on representation and modelling of speech signals for automatic speech and speaker recognition in noisy conditions. The chapter consisted of three parts. In the first part, we presented a novel method for estimation of the voicing information of speech spectra in the presence of noise. The presented method is based on calculating a similarity between the shape of signal short-term spectrum and the spectrum of the frame-analysis window. It does not require information about the F0 and is particularly applicable to speech pattern processing. Evaluation of the method was presented in terms of false-rejection and false-acceptance errors and good performance was demonstrated in noisy conditions. The second part of the chapter presented an employment of the voicing information into the missing-feature-based speech and speaker recognition systems to improve noise robustness. In particular, we were concerned with the mask estimation problem for voiced speech. It was demonstrated that the MFT-based recognition system employing the estimated spectral voicing information as a mask obtained results very similar to those of employing the oracle voicing information obtained based on full a-priori knowledge of noise. The achieved results showed significant recognition accuracy improvements over the standard recognition system. The third part of the chapter presented an incorporation of the spectral voicing information to improve modelling of speech signals in application to speech recognition in noisy conditions. The voicing-information was incorporated within an HMM-based statistical framework in the back-end of the ASR system. In the proposed model, a voicing-probability was estimated for each mixture at each HMM state and it served as a penalty during the recognition for those mixtures/states whose voicing information did not correspond to the voicing information of the signal. The evaluation was performed in the standard model and in the missing-feature model that had compensated for the effect of noise and experimental results demonstrated significant recognition accuracy improvements in strong noisy conditions obtained by the models incorporating the voicing information.

8. References

Boll, S.F. (1979). Suppression of acoustic noise in speech using spectral subtraction, *IEEE Trans. on Acoustic, Speech, and Signal Proc.*, Vol. 27, No. 2, pp. 113–120, Apr. 1979.

Bourlard, H. & Dupont, S. (1996). A new ASR approach based on independent processing and recombination of partial frequency bands, *Proceedings of ICSLP*, Philadelphia, USA, 1996.

Cooke, M.; Green, P.; Josifovski, L. & Vizinho, A. (2001). Robust automatic speech recognition with missing and unreliable acoustic data. *Speech Communication,* Vol.34, No. 3, 2001, pp.267–285.

Davis, S. & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, *IEEE Trans. on Acoustic, Speech, and Signal Proc.*, Vol. 28, No. 4, 1980, pp. 357–366.

Drygajlo A. &. El-Maliki, M. (1998). Speaker verification in noisy environment with combined spectral subtraction and missing data theory, *Proceedings of ICASSP*, Seattle, WA, Vol. I, pp. 121–124, 1998.

Gales M.J.F. & Young, S.J. (1996). Robust continuous speech recognition using parallel model combination, *IEEE Trans. on Speech and Audio Proc.,* Vol. 4, pp. 352–359, 1996.

Garofolo, J. S.; Lamel, L. F.; Fisher, W. M.; Fiscus, J. G.; Pallett, D. S. & Dahlgren, N. L. (1993). The darpa timit acoustic-phonetic continuous speech corpus. *Linguistic Data Consortium*, Philadelphia.

Graciarena, M.; Franco, H.; Zheng, J.; Vergyri, D. & Stolcke, A. (2004). Voicing feature integration in SRI's decipher LVCSR system, *Proceedings of ICASSP,* Montreal, Canada, pp. 921–924, 2004.

Griffin, D. & Lim, J. (1988). Multiband-excitation vocoder, *IEEE Trans. On Acoustic, Speech, and Signal Proc.*, Vol. 36, Feb. 1988, pp. 236–243.

Hirsch, H. & Pearce, D. (2000). The Aurora experimental framework for the performance evaluations of speech recognition systems under noisy conditions, *ISCA ITRW ASR'2000: Challenges for the New Millenium*, Paris, France, 2000.

Jackson, P.; Moreno, D.; Russell, M. & Hernando, J. (2003). Covariation and weighting of harmonically decomposed streams for ASR, *Proceedings of Eurospeech*, Geneva, Switzerland, pp. 2321–2324, 2003.

Jančovič, P. & Köküer, M. (2007a). Estimation of voicing-character of speech spectra based on spectral shape, *IEEE Signal Processing Letters,* Vol. 14, No. 1, 2007, pp. 66–69.

Jančovič, P. & Köküer, M. (2007b). Separation of harmonic and speech signals using sinusoidal modeling, *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA),* New York, USA, Oct. 21-24, 2007.

Jančovič, P. & Köküer, M. (2007c). Incorporating the voicing information into HMM-based automatic speech recognition, *IEEE Workshop on Automatic Speech Recognition and Understanding,* Kyoto, Japan, pp. 42–46, Dec. 6-13, 2007.

Jančovič, P. & Ming, J. (2001). A multi-band approach based on the probabilistic union model and frequency-filtering features for robust speech recognition, *Proceedings of Eurospeech*, Aalborg, Denmark, pp. 1111–1114, 2001.

Jančovič, P. & Ming, J. (2002). Combining the union model and missing feature method to improve noise robustness in ASR, *Proceedings of ICASSP*, Orlando, Florida, pp. 69–72, 2002.

Kitaoka, N.; Yamada, D. & Nakagawa, S. (2002). Speaker independent speech recognition using features based on glottal sound source, *Proceedings of ICSLP*, Denver, USA, pp. 2125–2128, 2002.

Lippmann, R.P. & Carlson, B.A. (1997). Using missing feature theory to actively select features for robust speech recognition with interruptions, filtering and noise, *Proceedings of Eurospeech*, Rhodes, Greece, pp. 37–40, 1997.

Ljolje, A. (2002). Speech recognition using fundamental frequency and voicing in acoustic modeling, *Proceedings of ICSLP*, Denver, USA, pp. 2137–2140, 2002.

McAulay, R. J. & Quatieri, T. F. (1990). Pitch estimation and voicing detection based on a sinusoidal speech model, *Proceedings of ICASSP*, pp. 249–252, 1990.

McCree, A.V. & Barnwell, T.P. (1995). A mixed excitation LPC vocoder model for low bit rate speech coding, *IEEE Trans. on Acoustic, Speech, and Signal Proc.,* Vol. 3, No. 4, July 1995, pp. 242–250.

Nadeu, C.; Macho, D. & Hernando, J. (2001). Time and frequency filtering of filter-bank energies for robust HMM speech recognition, *Speech Communication*, Vol. 34, 2001, pp. 93–114.

O'Shaughnessy, D. & Tolba, H. (1999). Towards a robust/fast continuous speech recognition system using a voiced-unvoiced decision, *Proceedings of ICASSP*, Phoenix, Arizona, pp. 413–416, 1999.

Renevey, P. & Drygajlo, A. (2000). Statistical estimation of unreliable features for robust speech recognition, *Proceedings of ICASSP*, Istanbul, Turkey, pp. 1731–1734, 2000.

Seltzer, M.L.; Raj, B. & Stern, R.M. (2004). A Bayesian classifier for spectrographic mask estimation for missing feature speech recognition, *Speech Communication,* Vol. 43, pp. 379–393, 2004.

Stylianou, Y. (2001). Applying the harmonic plus noise model in concatenative speech synthesis, *IEEE Trans. on Speech and Audio Proc.,* Vol. 9, No.1, Jan. 2001, pp. 21–29.

Thomson, D. & Chengalvarayan, R. (2002). The use of voicing features in HMM-based speech recognition, *Speech Communication*, Vol. 37, 2002, pp. 197–211.

Vaseghi, S.V. (2005). *Advanced Digital Signal Processing and Noise Reduction*, John Wiley & Sons, 2005.

Young, S.; Kershaw, D.; Odell, J.; Ollason, D.; Valtchev, V. & Woodland, P. (1999). *The HTK Book*. V2.2.

Zolnay, A.; Schluter, R. & Ney, H. (2003). Extraction methods of voicing feature for robust speech recognition, *Proceedings of Eurospeech*, Geneva, Switzerland, pp. 497–500, 2003.

Zou, X.; Jančovič, P.; Liu, J. & Köküer, M. (2007). ICA-based MAP algorithm for speech signal enhancement, *Proceedings of ICASSP*, Honolulu, Hawaii, Vol. IV, pp. 569-572, April 14-20, 2007.

4

Time-Frequency Masking: Linking Blind Source Separation and Robust Speech Recognition

Marco Kühne[1], Roberto Togneri[1] and Sven Nordholm[2]
[1]*The University of Western Australia*
[2]*Western Australian Telecommunications Research Institute*
[2]*Curtin University of Technology*
Australia

1. Introduction

In order to deploy automatic speech recognition (ASR) effectively in real world scenarios it is necessary to handle hostile environments with multiple speech and noise sources. One classical example is the so-called "cocktail party problem" (Cherry, 1953), where a number of people are talking simultaneously in a room and the ASR task is to recognize the speech content of one or more target speakers amidst other interfering sources. Although the human brain and auditory system can handle this everyday problem with ease it is very hard to solve with computational algorithms. Current state-of-the-art ASR systems are trained on clean single talker speech and therefore inevitably have serious difficulties when confronted with noisy multi-talker environments.

One promising approach for noise robust speech recognition is based on the missing data automatic speech recognition (MD-ASR) paradigm (Cooke et al., 2001). MD-ASR requires a time-frequency (T-F) mask indicating the reliability of each feature component. The classification of a partly corrupted feature vector can then be performed on the reliable parts only, thus effectively ignoring the components dominated by noise. If the decision about the reliability of the spectral components can be made with absolute certainty, missing data systems can achieve recognition performance close to clean conditions even under highly adverse signal-to-noise ratios (SNRs) (Cooke et al., 2001; Raj & Stern, 2005; Wang, 2005).

The most critical part in the missing data framework is the blind estimation of the feature reliability mask for arbitrary noise corruptions. The remarkable robustness of the human auditory system inspired researchers in the field of computational auditory scene analysis (CASA) to attempt auditory-like source separation by using an approach based on human hearing. CASA systems first decompose a given signal mixture into a highly redundant T-F representation consisting of individual sound elements/atoms. These elementary atoms are subsequently arranged into separate sound streams by applying a number of grouping cues such as proximity in frequency and time, harmonicity or common location (Bregman, 1990; Brown & Cooke, 1994; Wang, 2005). The output of these grouping mechanisms can often be represented as a T-F mask which separates the target from the acoustic background. Essentially, T-F masking provides a link between speech separation and speech recognition (Cooke et al., 2001; Wang, 2005).

Most previous work related to missing data mask estimation is based on single-channel data (see Cerisara et al., (2007) for a review) and relies on SNR criteria (Cooke et al., 2001; Barker et al., 2000; El-Maliki & Drygajlo, 1999), harmonicity cues (Hu & Wang, 2004; van Hamme, 2004) or cue combinations (Seltzer et al., 2004). Alternatively, binaural CASA models (Harding et al., 2006; Roman et al., 2003; Kim & Kil, 2007) exploit interaural time and intensity differences (ITD)/(IID) between two ears for missing data mask estimation. While used in the CASA community for quite some time, the concept of T-F masking has recently attracted some interest in the in the field of blind signal separation (BSS) (Yilmaz & Rickard, 2004; Araki et al., 2005). Similar to CASA, these methods exploit the potential of T-F masking to separate mixtures with more sources than sensors. However, the BSS problem is tackled from a signal processing oriented rather than psychoacoustic perspective. This, for instance, includes the use of multiple sensor pairs (Araki et al., 2007) and statistical approaches such as Independent Component Analysis (Kolossa et al., 2006; Hyvärinen, 1999).

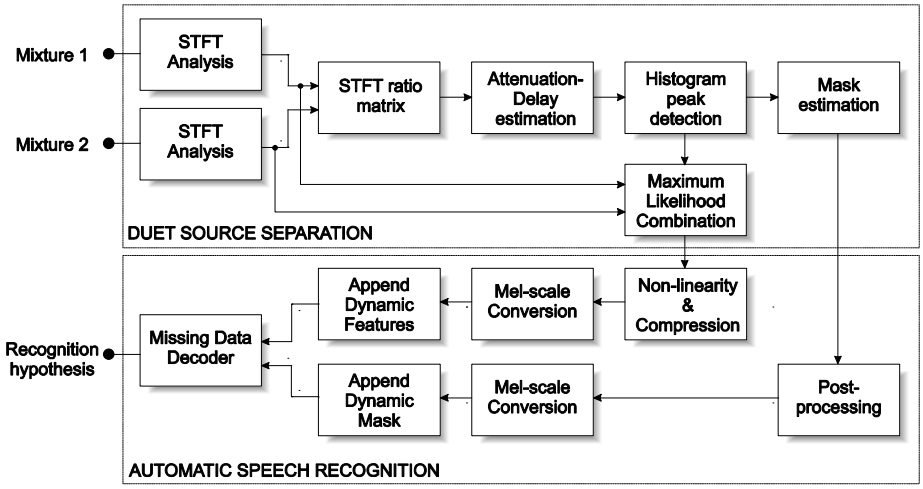


Fig. 1. Flowchart for proposed combination of DUET source separation and missing data speech recognition.

This chapter presents a scheme which combines BSS with robust ASR through the systematic application of T-F masking for both speech separation and speech recognition (Fig. 1). The outlined approach summarizes our previous work reported in Kühne et al. (2007; 2007a). In particular, we investigate the performance of a recently proposed BSS method called DUET (Yilmaz & Rickard, 2004) as front-end for missing data speech recognition. Since DUET relies on T-F masking for source demixing, this combination arises as a natural choice and is straightforward to implement. In Kühne et al. (2007) an approach was presented that avoids DUET's source reconstruction step and directly uses the mask together with the spectral mixture as input for the speech decoder. In subsequent work (Kühne et al., 2007a), a simple but effective mask post-processing step was introduced in order to remove spurious T-F points that can cause insertion errors during decoding. Our proposed combination fits seamlessly into standard feature extraction schemes (Young et al, 2006), but requires a modification of the decoding algorithm to account for missing feature components. It is particularly attractive for ASR scenarios where only limited space and resources for multi-channel processing are available (e.g., mobile phones).

The effectiveness of the proposed BSS-ASR combination is evaluated for a simulated cocktail party situation with multiple speakers. Experimental results are reported for a connected digits recognition task. Our evaluation shows that, when the assumptions made by DUET hold, the estimated feature reliability masks are comparable in terms of speech recognition accuracy to the oracle masks obtained with a prior knowledge of the sources. We further demonstrate that a conventional speech recognizer fails to operate successfully on DUET's resynthesized waveforms, which clearly shows the merit of the proposed approach.
The remainder of this chapter is organized as follows: Section 2 briefly reviews the DUET source separation method and outlines its main assumptions. Section 3 explains the methods used for feature extraction and missing data mask generation in more detail. Section 4 presents the experimental evaluation of the system. Section 5 gives a general discussion and illustrates the differences and similarities with a related binaural CASA segregation model. The section further comments on some of the shortcomings in the proposed approach. Finally, the chapter concludes in Section 6 with an outlook on future research.

2. Source separation

This section presents a short review of the DUET-BSS algorithm used in this study for blind separation of multiple concurrent talkers. We start with an introduction of the BSS problem for anechoic mixtures and highlight the main assumptions made by the DUET algorithm. After briefly outlining the main steps of the algorithm, the section closes with a short discussion on why the reconstructed waveform signals are not directly suitable for conventional speech recognition. For a more detailed review of DUET the reader is referred to Yilmaz & Rickard (2004) and Rickard (2007).

2.1 Anechoic mixing model

The considered scenario uses two microphone signals $x_1(t)$ and $x_2(t)$ to capture $N \geq 2$ speech sources $s_1(t), \ldots, s_N(t)$ assuming the following anechoic mixing model

$$x_m(t) = \sum_{j=1}^{N} a_{mj} s_j(t - \delta_{mj}), \qquad m = 1, 2 \tag{1}$$

where a_{mj} and δ_{mj} are the attenuation and delay parameters of source s_j at microphone x_m. The goal of any BSS algorithm is to recover the source signals $s_j(t), j = 1, \ldots, N$ using only the mixture observations $x_m(t), m = 1, 2$. The mixing model can be approximated in the Short-Time-Fourier-Transform (STFT) domain as an instantaneous mixture at each frequency bin *l* through

$$X_m(k,l) \approx \sum_{j=1}^{N} a_{mj} e^{-il\omega_0 \delta_{mj}} S_j(k,l). \tag{2}$$

The STFT transform $S(k,l)$ for a time domain signal $s(t)$ is defined as

$$S(k,l) := \sum_{\tau=-T/2}^{T/2-1} w(\tau) s(\tau + k\tau_0) e^{-il\omega_0 \tau}, \tag{3}$$

where τ_0 and ω_0 specify the time-frequency grid resolution and $w(\tau)$ is a window function (e.g., Hamming) of size T which attenuates discontinuities at the frame edges.
The instantaneous BSS problem can be solved quite elegantly in the frequency domain due to the sparsity of time-frequency representations of speech signals. DUET proceeds by considering the following STFT ratio

$$\frac{X_2(k,l)}{X_1(k,l)} = \frac{\sum_{j=1}^{N} a_{2j} e^{-il\omega_0\delta_{2j}} S_j(k,l)}{\sum_{j=1}^{N} a_{1j} e^{-il\omega_0\delta_{1j}} S_j(k,l)}, \tag{4}$$

where the nominator and denominator are weighted sums of complex exponentials representing the delay and attenuation of the source spectra at the two microphones.

2.2 Assumptions

The key assumption in DUET is that speech signals satisfy the so-called W-disjoint orthogonality (W-DO) requirement

$$S_i(k,l)S_j(k,l) = 0, \forall i \neq j, \forall k, l, \tag{5}$$

also known as "sparseness" or "disjointness" condition with the support of a source S_j in the T-F plane being denoted as $\Omega_j := \{(k,l) : S_j(k,l) \neq 0\}$. The sparseness condition (5) implies that the supports of two W-DO sources are disjoint, e. g., $\Omega_i \cap \Omega_j = \emptyset$. This motivates a demixing approach based on time-frequency masks, where the mask for source S_j corresponds to the indicator function for the support of this source:

$$M_j(k,l) = \begin{cases} 1, & \text{if } (k,l) \in \Omega_j \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

It has previously been shown (Wang, 2005; Yilmaz & Rickard, 2004; Roman et al., 2003) that binary time-frequency masks exist that are capable of demixing speech sources from just one mixture with high speech fidelity. For example, Wang (2005) proposed the notion of an ideal/oracle binary mask

$$O_j(k,l) := \begin{cases} 1, & \text{if } 20\log_{10}\frac{|S_j(k,l)|}{|\sum_{i\neq j} S_i(k,l)|}\text{dB} \geq 0\,\text{dB} \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

which determines all time-frequency points where the power of the source S_j exceeds or equals the power of the sum of all interfering sources (see Wang (2005) for a more detailed motivation of the ideal binary masks). Note that these masks can only be constructed if the source signals are known prior to the mixing process as they are defined by means of a SNR criterion. Instead, DUET relies on spatial cues extracted from two microphones to estimate the ideal binary mask. It solely depends on relative attenuation and delays of a sensor pair and assumes an anechoic environment where these cues are most effective. An additional assumption requires that the attenuation and delay mixing pairs for each source are unambiguous.

2.3 Estimation of relative mixing parameters using DUET

Due to (5) it follows, that only one arbitrary source S_j will be active at any T-F point such that (4) simplifies to

$$\frac{X_2(k,l)}{X_1(k,l)} = \frac{a_{2j}}{a_{1j}} e^{-il\omega_0(\delta_{2j}-\delta_{1j})} = a_j e^{-il\omega_0\delta_j}, \qquad \forall (k,l) \in \Omega_j \tag{8}$$

with a_j and δ_j denoting relative attenuation and delay parameters between both microphones and $(a_j \neq a_k)$ or $(\delta_j \neq \delta_k)$, $\forall j \neq k$. The goal is now to estimate for each source S_j the corresponding mixing parameter pair (a_j, δ_j) and use this estimate to construct a time-frequency mask that separates S_j from all other sources.
An estimate of the attenuation and delay parameter at each T-F point is obtained by applying the magnitude and phase operator to (8) leading to

$$\tilde{a}(k,l) := \left| \frac{X_2(k,l)}{X_1(k,l)} \right|, \qquad \tilde{\delta}(k,l) := -\frac{1}{l\omega_0} \arg\left(\frac{X_2(k,l)}{X_1(k,l)} \right). \tag{9}$$

If the sources are truly W-DO then accumulating the instantaneous mixing parameter estimates in (9) over all T-F points will yield exactly N distinct $(\tilde{a}, \tilde{\delta})$ pairs equal to the true mixing parameters:

$$\bigcup_{(k,l)} \left\{ (\tilde{a}(k,l), \tilde{\delta}(k,l)) \right\} = \{(a_j, \delta_j) : j = 1 \ldots, N\} \tag{10}$$

The demixing mask for each source is then easily constructed using the following binary decision

$$M_j(k,l) := \begin{cases} 1, & \text{if } \left(\tilde{a}(k,l), \tilde{\delta}(k,l)\right) = (a_j, \delta_j) \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

However, in practice the W-DO assumption holds only approximately and it will no longer be possible to observe the true mixing parameters directly through inspection of the instantaneous estimates in (9). Nevertheless, one can expect that the values will be scattered around the true mixing parameters in the attenuation-delay parameter space. Indeed, it was shown in Yilmaz & Rickard (2004) that T-F points with high power possess instantaneous attenuation-delay estimates close to the true mixing parameters. The number of sources and their corresponding attenuation-delay mixing parameters are then estimated by locating the peaks in a power weighted $(\tilde{\alpha}, \tilde{\delta})$-histogram (see Fig. 2a), where $\tilde{\alpha}(k,l) := \tilde{a}(k,l) - (\tilde{a}(k,l))^{-1}$ is the so-called symmetric attenuation (Yilmaz & Rickard, 2004). The peak detection was implemented using a weighted k-means algorithm as suggested in Harte et al. (2005).

2.4 Time-Frequency mask construction and demixing

Once the peak locations $(\hat{\alpha}_j, \hat{\delta}_j)$, $j = 1, \ldots, N$ have been determined, a second pass over the raw data set is required to assign each observation to one of the detected source locations. We used simple minimum distance classification to construct the binary T-F mask for source S_j as

$$\hat{M}_j(k,l) := \begin{cases} 1, & \text{if } j = \underset{z}{\operatorname{argmin}}\, d_z^2(k,l) \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

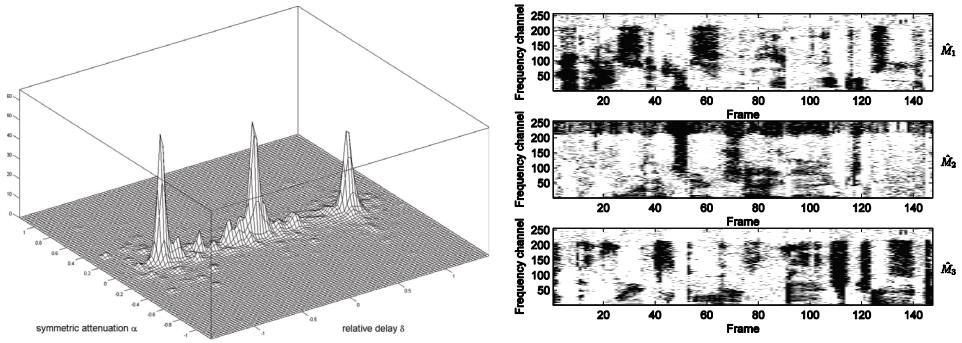
where d_z^2 is the squared Euclidean distance

$$d_z^2(k,l) = \left(\tilde{\alpha}(k,l) - \hat{\alpha}_z\right)^2 + \left(\tilde{\delta}(k,l) - \hat{\delta}_z\right)^2 \tag{13}$$

between the instantaneous mixing parameter estimate $(\tilde{\alpha}(k,l), \tilde{\delta}(k,l))$ and the histogram peak $(\hat{\alpha}_z, \hat{\delta}_z)$. The demixing then proceeds by masking the maximum likelihood combination $X_{\mathrm{ML}}(k,l)$ of both mixtures (Yilmaz & Rickard, 2004) to obtain the source estimate as

$$\hat{S}_j(k,l) = \hat{M}_j(k,l) \underbrace{\left(\frac{X_1(k,l) + \hat{a}_j e^{il\omega_0 \hat{\delta}_j} X_2(k,l)}{1 + \hat{a}_j^2} \right)}_{X_{\mathrm{ML}}(k,l)}. \tag{14}$$

$\hat{S}_j(k,l)$ can then be converted back into the time domain by means of an inverse STFT transformation. Note that for the maximum likelihood combination of both mixtures the symmetric attenuation parameter was converted back to the relative attenuation parameter $\hat{a}_j$. However, here we are interested in evaluating the DUET demixing performance using an automatic speech recognizer. The reconstruced time domain signal $\hat{s}_j(t)$ will not be directly applicable for conventional speech recognition systems because non-linear masking effects due to $\hat{M}_j$ are introduced during waveform resynthesis. Conventional speech recognizers perform decoding on complete spectra and can not deal with partial spectral representations. Therefore, additional processing steps, either in the form of data imputation to reconstruct missing spectrogram parts (Raj & Stern, 2005) or missing data marginalization schemes (Cooke et al., 2001) that can handle partial data during decoding, are required before speech recognition can be attempted.



(a) (b)

Fig. 2. Power weighted attenuation-delay histogram (a) for a mixture of three sources with mixing parameters $\{(\alpha_1;\delta_1),(\alpha_2;\delta_2),(\alpha_3;\delta_3)\} = \{(-0.03;0.94),(0;0),(0.03;-0.94)\}$ and (b) the estimated time-frequency masks with selected points marked in black.

In this work the latter option was chosen allowing us to avoid source reconstrucion and directly exploit the spectrographic masks for missing data decoding. After source separation the missing data recognizer was informed which mask corresponded to the target speaker by comparing the detected histogram peaks with the true mixing parameters. However, the high STFT resolution is usually not suitable for statistical pattern recognition as it would

lead to very high-dimensional feature vectors. The following section explains how the results of the DUET separation can be integrated into standard feature extraction schemes and be utilized for missing data speech recognition.

3. Automatic speech recognition with missing data

A Hidden Markov Model (HMM) based missing data speech recognizer (Cooke et al., 2001) was used for all speech recognition experiments reported in this study. While the HMMs are trained on clean speech in exactly the same manner as in conventional ASR the decoding is treated differently in missing data recognition. Additionally to the feature vector sequence a mask is required to declare each feature component as reliable or unreliable using a hard or soft decision (Barker et al., 2000; Morris et al., 2001).
This section starts with a detailed description of the extracted acoustic features and how the DUET masks can be utilized for missing data recognition. A mask post-processing step is introduced in order to remove isolated mask points that can cause insertion errors in the speech decoding process. We then proceed with the missing data decoding and explain how observation likelihoods are computed in the presence of missing feature components.

3.1 Feature extraction

It is known that the human ear resolves frequencies by grouping several adjacent frequency channels into so-called critical bands (Moore, 2003). For speech recognition purposes the linear STFT frequency resolution is usually converted to a perceptual frequency scale, such as the bark or mel scale (Moore, 2003; Young et al., 2006). A widely used approximation of the non-linear frequency resolution of the human auditory system is the mel-frequency scale

$$\mathfrak{f}(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right), \tag{15}$$

where f denotes the linear frequency in Hz and $\mathfrak{f}$ is the corresponding non-linear frequency scale in mel. The grouping of individual frequency channels into critical bands can be accomplished by applying a triangular mel-filterbank to the magnitude or power FFT spectrum (Young et al., 2006). The triangular filters

$$\lambda_b(l) = \begin{cases} 0 & l\omega_0 < \omega_{c_{(b-1)}}, \\ \frac{l\omega_0 - \omega_{c_{(b-1)}}}{\omega_{c_b} - \omega_{c_{(b-1)}}} & \omega_{c_{(b-1)}} \leq l\omega_0 \leq \omega_{c_b}, \\ \frac{\omega_{c_{(b+1)}} - l\omega_0}{\omega_{c_{(b+1)}} - \omega_{c_b}} & \omega_{c_b} \leq l\omega_0 \leq \omega_{c_{(b+1)}}, \\ 0 & l\omega_0 > \omega_{c_{(b+1)}}, \end{cases} \tag{16}$$

with

$$\omega_{c_b} = 2\pi \cdot 700 \left(10^{\mathfrak{f}_{c_b}/2595} - 1\right) \tag{17}$$

are equally spaced along the mel-frequency scale through

$$\mathfrak{f}_{c_b} = \mathfrak{f}_l + b \cdot \frac{\mathfrak{f}_h - \mathfrak{f}_l}{B+1}, \qquad b = 1, \ldots, B. \tag{18}$$

Here B is the number of mel-frequency channels and $\mathfrak{f}_l, \mathfrak{f}_h$ are the lower and higher cut-offs of the mel-frequency axis.

(A) **Acoustic feature extraction:** The preferred acoustic features employed in missing data speech recognition are based on spectral representations rather than the more common mel-frequency-cepstral-coefficients (MFCCs). This is due to the fact that a spectrographic mask contains localized information about the reliability of each spectral component, a concept not compatible with orthogonalized features, such as cepstral coefficients (see also de Veth et al. (2001) for a further discussion). For the scope of this study the extracted spectral features for missing data recognition followed the FBANK feature implementation of the widely accepted Hidden Markov Model Toolkit (Young et al., 2006).

Let $\boldsymbol{o}_k = (o_{k1}, \ldots, o_{kn})^T$ be the n-dimensional spectral feature vector at time frame k. The static log-spectral feature components (see Fig. 3.b) are computed as

$$o_{kb} = \log\left(\max\left\{\sum_l \lambda_b(l)|X_{\mathrm{ML}}(k,l)|, 1\right\}\right), \quad b = 1, \ldots, B, \tag{19}$$

where λ_b are the triangular mel-filterbank weights defined in (16) and X_{MI} is the maximum likelihood combination of both mixture observations as specified in (14). It is common to append time derivatives to the static coefficients in order to model their evolution over a short time period. These dynamic parameters were determined here via the standard regression formula

$$\Delta o_{kb} := o_{k(\frac{n}{2}+b)} = \frac{\sum_{\theta=1}^{\Theta} \theta\left(o_{(k+\theta)b} - o_{(k-\theta)b}\right)}{2\sum_{\theta=1}^{\Theta} \theta^2}, \quad b = 1, \ldots, B, \tag{20}$$

where Δo_{kb} is the regression coefficient at time frame k and mel-frequency subband b, computed over the corresponding static features using a temporal integration window of size Θ (Young et al., 2006). For this study, only first-order regression coefficients were used, thus producing a feature vector of dimension $n = 2B$.

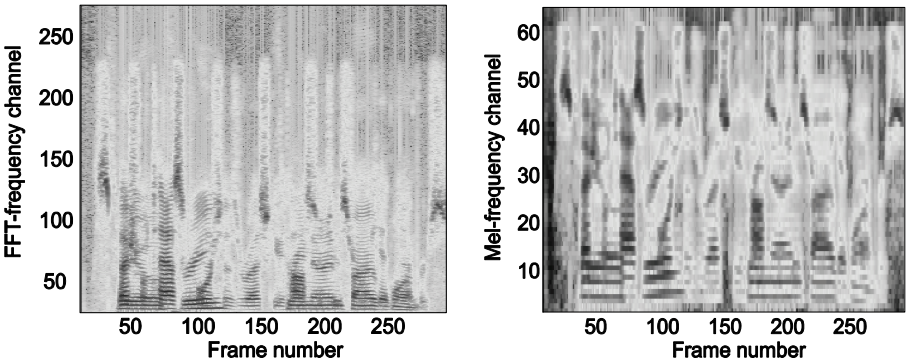


Fig. 3. Spectrograms for the TIDIGITS utterance "3o33951" mixed with three interfering speakers in anechoic condition. (a) linear FFT frequency scale; (b) non-linear mel-frequency scale

(B) Missing data reliability masks: The reliability of each feature component is indicated by a corresponding missing feature mask provided here by the source separation stage. Before converting the mask to the mel-frequency scale we introduce a mask post-processing step to eliminate spurious points in the mask. One important aspect that has not been considered so far is the high correlation of neighboring time-frequency points. That is, if a time-frequency point (k,l) is assigned to speaker S_j then it is very likely that points in the neighborhood of (k,l) are also belonging to S_j (see Fig. 5a).
The DUET method solely relies on the mask assignment in the attenuation-delay parameter space and does not take neighborhood information into account. We observed that for mixtures with more than two sources the time-frequency masks were overlaid by some scattered isolated "noise" points (compare Fig. 5a,c). This type of noise is similar to "shot-noise" known in the image processing community and can be dealt with effectively by means of a non-linear median filter (Russ, 1999). Similar smoothing techniques have been used previously for missing data mask post-processing (Harding et al., 2005). For this study, the median filter was preferred over other linear filters as it preserves the edges in the mask while removing outliers and smoothing the homogenous regions. The basic operation of a two-dimensional median filter consists in sorting the mask values of a T-F point (k,l) and its neighborhood and replacing the mask value with the computed median $\bar{M}(k,l)$. Several different neighborhood patterns exist in the literature ranging from 4-nearest neighbors over 3×3 or 5×5 square neighborhoods to octagonal regions (Russ, 1999). Here, we used a 5×5 plus sign-shaped median filter

$$\bar{M}(k,l) = \text{median}\left\{M(u,v) : (u,v) \in N_{(k,l)}\right\} \tag{21}$$

with the neighborhood pattern (Fig. 4) defined as

$$N_{(k,l)} := \{(n,m) : \max\{|n-k|, |m-l|\} \leq 2 \wedge \min\{|n-k|, |m-l|\} = 0\}\,.$$

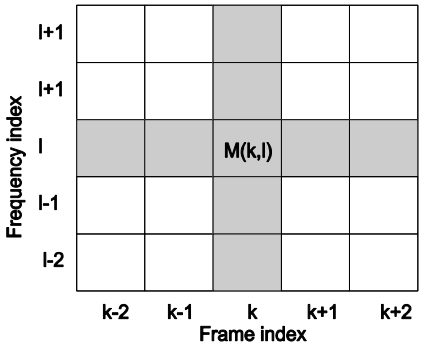


Fig. 4. Plus signed-shaped neighborhood pattern of size 5×5 used for the proposed two-dimensional median filtering of the DUET localization masks.

The filter is able to preserve vertical or horizontal lines that would otherwise be deleted by square neighborhoods. This is important in our application as these lines are often found at sound onsets (vertical, constant time) or formant frequency ridges (horizontal, constant frequency). Other more sophisticated rank filters like the hybrid median filter or cascaded median filters have not been considered here but can be found in Russ (1999). The effect of

the median filtering can be observed in Fig. 5e, where most of the isolated points have been removed while still preserving the main characteristics of the oracle mask (Fig. 5a).

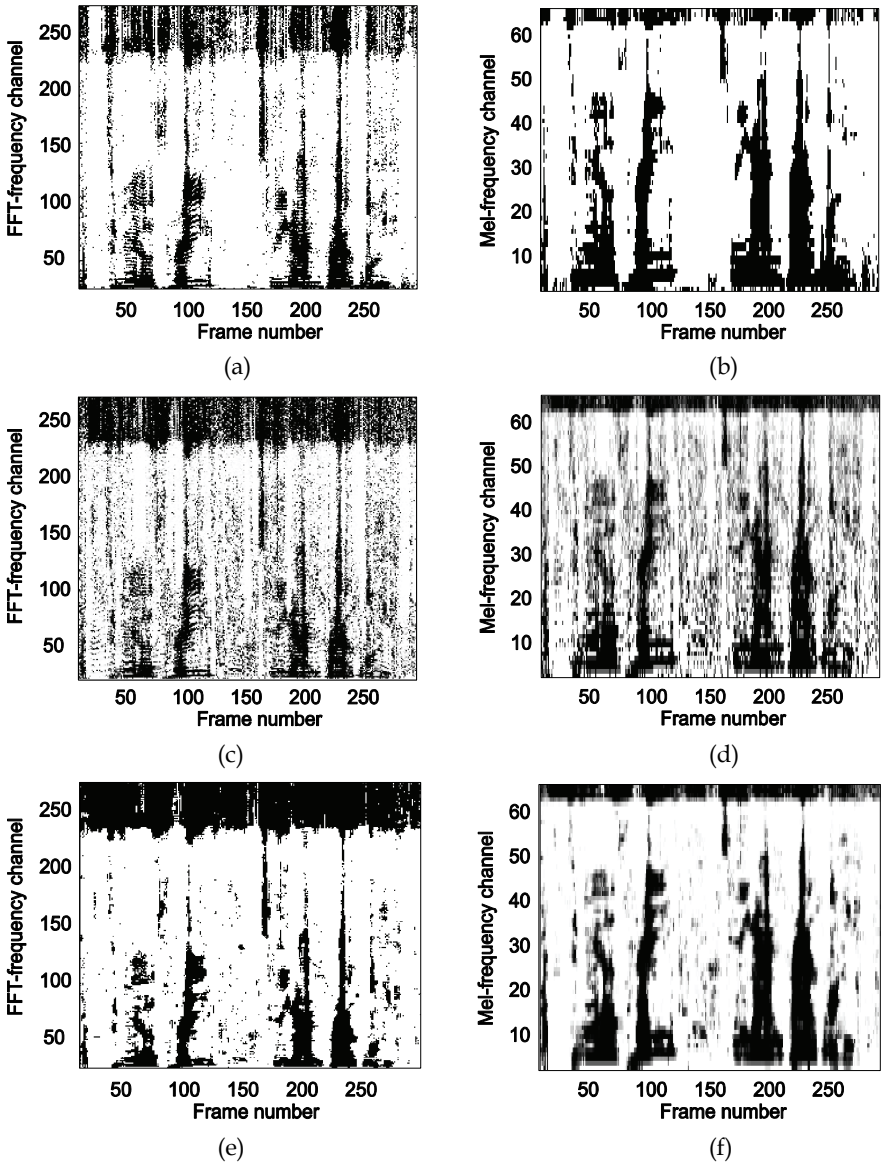


Fig.5. Example of localization masks for the TIDIGITS target source (black) "3o33951" in a mixture of three competing speakers (white). (a) oracle mask on linear FFT frequency scale; (b) oracle mask on non-linear mel-frequency scale; (c) DUET mask on linear FFT frequency scale; (d) DUET mask converted to non-linear mel-frequency scale; (e) median filtered mask of (c); (f) median filtered DUET mask from (e) converted to non-linear mel-frequency scale

The final missing data mask is then obtained by converting the high STFT resolution to the mel-frequency domain. Similar to (19), we apply the triangular mel-weighting function λ_b to obtain a soft mel-frequency mask

$$w_{kb} = \frac{\sum_l \lambda_b(l)\bar{M}(k,l)}{\sum_l \lambda_b(l)}. \tag{22}$$

While the mask (22) is valid for static features only a reliability mask is also required for the dynamic feature coefficients in (20). The corresponding mask for Δo_{kb} was determined based on the static mask values as

$$\Delta w_{kb} := w_{k(\frac{n}{2}+b)} = \prod_{\substack{\theta=-\Theta,\\ \theta\neq 0}}^{\Theta} w_{(k+\theta)b}. \tag{23}$$

3.2 HMM observation likelihoods with missing features

In this study a HMM based missing data recognizer was used for scoring the n-dimensional spectro-temporal feature vectors described in Section 3.1. The HMM state output distributions were modeled via Gaussian mixture models (GMMs) with diagonal covariance matrices. Let the GMM model parameters for a particular HMM state q be denoted as $\Lambda_q = \{\boldsymbol{c}_q, \boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2\}$, where the three components represent the mixture weights, mean and variance vectors of the Gaussian mixture probability density function. For a GMM with R mixtures the emission likelihood of $\boldsymbol{o}_k$ for HMM state q is given by

$$p(\boldsymbol{o}_k|\Lambda_q) = \sum_{r=1}^{R} c_{qr} \prod_{i=1}^{n} p\left(o_{ki}|\mu_{qri}, \sigma_{qri}^2\right), \tag{24}$$

where in the case of missing or uncertain features $p\left(o_{ki}|\mu_{qri}, \sigma_{qri}^2\right)$ is evaluated as

$$p\left(o_{ki}|\mu_{qri}, \sigma_{qri}^2\right) = w_{ki}\mathcal{N}\left(o_{ki}; \mu_{qri}, \sigma_{qri}^2\right) + (1-w_{ki})\frac{1}{b_{ki}-a_{ki}}\int_{a_{ki}}^{b_{ki}} \mathcal{N}\left(\tilde{o}_{ki}; \mu_{qri}, \sigma_{qri}^2\right) d\tilde{o}_{ki}, \tag{25}$$

with w_{ki} denoting the value of the missing data mask at T-F point (k,i), a_{ki} and b_{ki} being the lower and upper integration bound and $\mathcal{N}(o_{ki}; \mu_{qri}, \sigma_{qri}^2)$ being a univariate Gaussian

$$\mathcal{N}\left(o_{ki}; \mu_{qri}, \sigma_{qri}^2\right) = \frac{1}{\sqrt{2\pi\sigma_{qri}^2}} \exp\left[-\frac{1}{2}\frac{(o_{ki}-\mu_{qri})^2}{\sigma_{qri}^2}\right], \tag{26}$$

with mean μ_{qri} and variance σ_{qri}^2. The value of the missing data mask w_{ki} weights the present and missing data contributions with a soft "probability" between 0 and 1 (Harding et al., 2006; Barker et al., 2000). The likelihood contribution in (25) for the missing static features is evaluated as a bounded integral over the clean static feature probability density by exploiting the knowledge that the true clean speech value is confined to the interval between zero and the observed noisy spectral energy, e.g. $o_{ki}^{\text{clean}} \in [a_{ki}, b_{ki}] = [0, o_{ki}], \forall i = 1, \ldots, \frac{n}{2}$. Past research (Cooke et al., 2001; Morris et al., 2001) has shown that bounding the integral in (25) is beneficial as it provides an effective

mechanism to incorporate counter-evidence by penalizing models with insufficient spectral energy. However, no bounds on dynamic feature components were utilized here, thus $a_{ki} \to -\infty$ and $b_{ki} \to \infty, \forall i = \frac{n}{2} + 1, \ldots, n$.

4. Experimental evaluation

4.1 Setup

(A) Recognizer architecture and HMM model training: The proposed system was evaluated via connected digit experiments on the TIDIGITS database (Leonard, 1984) with a sample frequency of $20\,\mathrm{kHz}$. The training set for the recognizer consisted of 4235 utterances spoken by 55 male speakers. The HTK toolkit (Young et al., 2006) was used to train 11 word HMMs ('1'-'9','oh','zero') each with eight emitting states and two silence models ('sil','sp') with three and one state. All HMMs followed standard left-to-right models without skips using continuous Gaussian densities with diagonal covariance matrices and $R = 10$ mixture components. Two different sets of acoustic models were created. Both used $25\,\mathrm{ms}$ Hamming-windows with $10\,\mathrm{ms}$ frame shifts for the STFT analysis. Note that Yilmaz & Rickard (2004) recommend a Hamming window size of 64 ms for a sampling frequency of 16 kHz in order to maximize the W-DO measure for speech signals. However, for the ASR application considered here, the chosen settings are commonly accepted for feature extraction purposes. The first set of HMMs was used as the cepstral baseline system with 13 MFCCs derived from a $B = 32$ channel HTK mel-filterbank plus delta and acceleration coefficients ($\Theta = 2$) and cepstral mean normalization. This kind of baseline has been widely used in missing data ASR evaluations (Cooke et al., 2001; Morris et al., 2001; Harding et al., 2006). The second model set was used for the missing data recognizer and used spectral rather than cepstral features as described in Section 3.1. In particular, acoustic features were extracted from a HTK mel-filterbank with $B = 64$ channels and first order delta coefficients ($\Theta = 2$) were appended to the static features according to (19) and(20).

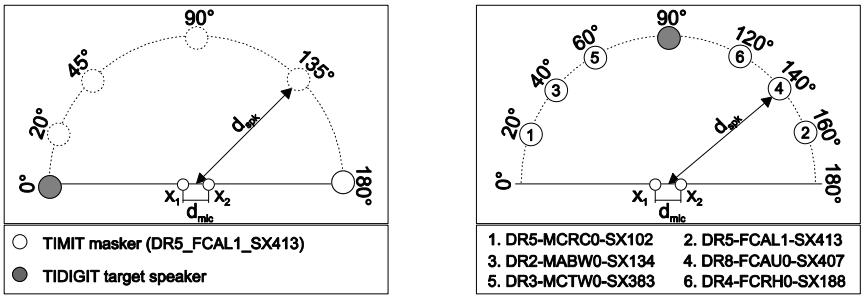
(B) Test data set and room layout: The test set consisted of 166 utterances of seven male speakers containing at least four digits mixed with several masking utterances taken from the TIMIT database (Garofolo et al., 1993; see Table 1).

TIMIT ID code			Utterance transcription
Dialect	Speaker	Sentence	
DR5	MCRC0	SX102	"Special task forces rescue hostages from kidnappers."
DR5	FCAL1	SX413	"Daphne's Swedish needlepoint scarf matched her skirt."
DR2	MABW0	SX134	"December and January are nice months to spend in Miami."
DR8	FCAU0	SX407	"Laugh, dance, and sing if fortune smiles upon you."
DR3	MCTW0	SX383	"The carpet cleaners shampooed our oriental rug."
DR4	FCRH0	SX188	"Who authorized the unlimited expense account?"

Table 1. Transcription for six utterances taken from the test section of the TIMIT database.

The signal-to-interferer ratio (SIR) for each masker was approximately $0\,\mathrm{dB}$. Stereo mixtures were created by using an anechoic room impulse response of a simulated room of size

$4\,\mathrm{m} \times 6\,\mathrm{m} \times 3\,\mathrm{m}$ (length x width x height). Two microphones were positioned in the center of the room, $2\,\mathrm{m}$ above the ground, with an interelement distance of $d_{\mathrm{mic}} = 1.72\,\mathrm{cm}$ to guarantee accurate phase parameter estimates (Yilmaz & Rickard, 2004). Fig. 6a shows the setup for a single masker scenario and Fig. 6b for a multi-speaker scenario with up to six different speech maskers (three male, three female) placed at a distance of $d_{\mathrm{spk}} = 1\,\mathrm{m}$ to the microphones. For testing, the HTK decoder (HVite) was modified according to (25) to incorporate the missing data marginalization framework.



(a) single masker (b) multiple masker

Fig. 6. Source configurations for the TIDIGITS target speaker and (a) a single TIMIT masker placed at different angles and (b) for corruption with multiple TIMIT maskers.

(C) Performance measures: The following standard performance measures were computed based on the decoder output (Young et al., 2006). The percentage correctness score is defined as

$$\mathrm{COR} = \frac{\mathrm{NUM} - \mathrm{DEL} - \mathrm{SUB}}{\mathrm{NUM}} \times 100\%, \tag{27}$$

where NUM is the total number of digits in the test set and DEL and SUB denote the deletion and substitution errors, respectively. The second performance measure, the percent accuracy is defined as

$$\mathrm{ACC} = \frac{\mathrm{NUM} - \mathrm{DEL} - \mathrm{SUB} - \mathrm{INS}}{\mathrm{NUM}} \times 100\% \tag{28}$$

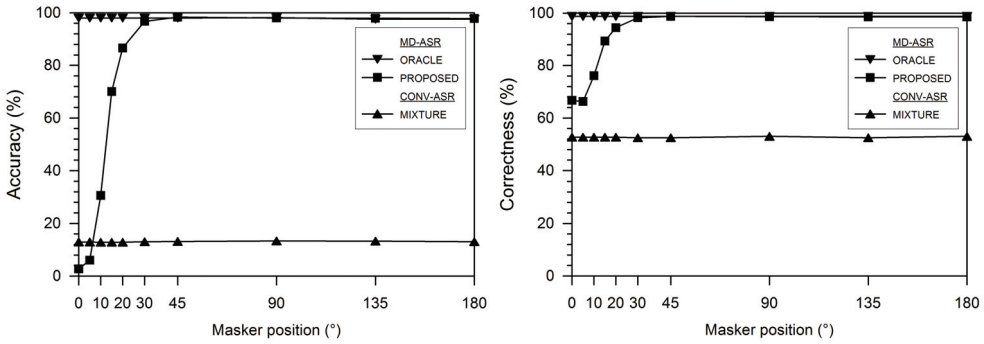
and in contrast to (27) additionally considers insertion errors denoted as INS. The accuracy score is therefore considered the more representative performance measure.

4.2 Results

A number of experiments were conducted to investigate the DUET separation in terms of speech recognition performance. The cepstral baseline measured the decoder's robustness against speech intrusions by scoring directly on the speech mixture. The missing data system reported the improvements over this baseline obtained by ignoring the spectral parts that are dominated by interfering speakers as indicated by the missing data masks. The performance in clean conditions (zero maskers) was 99.16% for the cepstral baseline and 98.54% for the spectral missing data system using the unity mask.

(A) Angular separation between target and masker: The first experiment used a female TIMIT speech masker to corrupt the target speech signal. The speaker of interest remained

stationary at the 0° location while the speech masker was placed at different angles but identical distance to the microphone pair (see Fig. 6a).



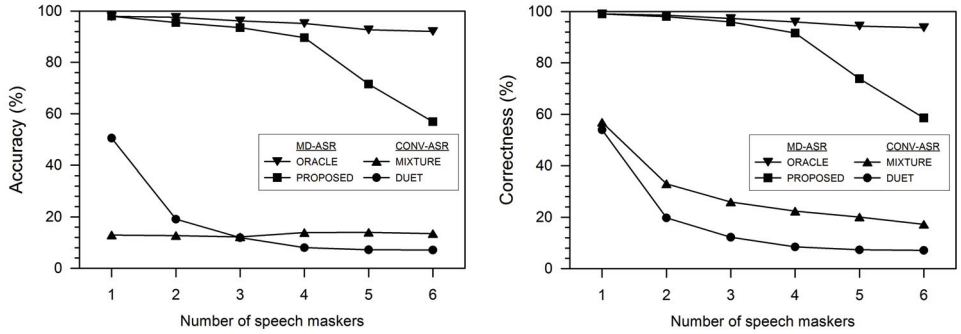
(a) (b)

Fig. 7. Speech recognition performance in terms of (a) accuracy and (b) correctness score for different masker positions. The target remained stationary at the 0° location. A conventional decoder using MFCC features was used to score on the speech mixtures. The spectral missing data system performed decoding with the proposed soft reliability mask (DUET+post-processing+mel-scale conversion) and the binary oracle mask.

The recognition performance was evaluated for a conventional recognizer and the missing data system using the oracle and estimated soft masks (Fig. 7).

Not surprisingly, the oracle mask performed best marking the upper performance bound for the missing data system while the conventional recognizer represented the lower bound. When the speech masker was placed between 45° to 180° angle relative to the target speaker, the estimated mask almost perfectly matched the oracle mask and hence achieved very high recognition accuracy. However, once the spatial separation between masker and target fell below 30° the accuracy score rapidly started to deteriorate falling below that of the cepstral baseline at the lowest separation angles (0°-5°). The correctness score followed the same trend as the accuracy score but performed better than the baseline for closely spaced sources. For these small angular separations the assumption that the sources possess distinct spatial signatures becomes increasingly violated and the DUET histogram localization starts to fail. The more the sources move together the less spatial information is available to estimate the oracle mask leading to large mask estimation errors. Nevertheless, the oracle masks (7) still exist even when target and masker are placed at identical positions because they depend on the local SNR rather than spatial locations.

(B) Number of concurrent speech maskers: The second experiment recorded the recognition performance when the target speaker was corrupted by up to six simultaneous TIMIT maskers (Fig. 8). Accuracy and correctness score were measured for the conventional recognizer using as input the speech mixture or the demixed target speaker as generated by DUET. As before, the missing data recognizer used the oracle and estimated soft masks. The number of simultaneously active speech maskers was increased by successively adding one masker after another according to the order shown in Fig. 6b.



(a) (b)

Fig. 8. Speech recognition performance in terms of (a) accuracy and (b) correctness score for different numbers of concurrent speech maskers. A conventional decoder using MFCC features was used to score on the speech mixtures and DUET's reconstructed target signal. The spectral missing data system performed decoding with the proposed soft reliability mask (DUET+post-processing+mel-scale conversion) and the binary oracle mask.

As expected, the conventional recognizer performed very poorly when scoring on the speech mixture. Performance dropped from 99% in clean conditions to 13% for the single speech masker case. Clearly, state-of-the-art cepstral feature extraction alone provides no protection against additive noise intrusions. For all but the single masker case, it also failed to produce significant improvements for the demixed DUET speech signal. In fact, for most conditions scoring on the speech mixture was better than decoding with the demixed DUET output. As discussed in Section 2.4 and 3.1, conventional speech recognizers require complete data and can not deal with masked spectra such as produced by DUET.

In contrast, the missing data system is able to handle missing feature components and provided the upper performance bound when using the oracle mask. Performance degraded very gradually with only a 6% decrease between clean conditions and corruption with six speech maskers. The estimated soft missing data masks closely matched the performance of the oracle masks for up to three simultaneously active speech maskers before starting to fall behind. The more speakers are present in the mixture the more the sparseness assumption (5) becomes invalid making an accurate peak detection in the attenuation-delay histogram increasingly difficult. Indeed, closer inspection of the 5 & 6 masker scenarios revealed that often peaks were overlapping and the peak detection algorithm failed to identify the locations correctly. For example, once the fifth masker was added, we observed in some cases that the histogram showed only four distinct peaks instead of five. This occasionally led the peak detection algorithm to place the fifth peak near the target speaker location. Due to DUET's minimum distance classification the wrongly detected speaker location absorbed some of the T-F points actually belonging to the target speaker. Consequently, performance dropped significantly for the 5 & 6 masker configurations, as evident from Fig. 8. Results can be improved somewhat by using soft assignments (Araki et al., 2006a; Kühne et al., 2007a) instead of the winner-takes-it-all concept utilized for the mask construction in (12).

(C) Mask post-processing: The last experiment investigated the influence of the proposed mask post-processing for a four speaker configuration (three maskers). To underline the importance of the mask smoothing the recognition performance with and without the

proposed two-dimensional median filtering was measured (see Table 2). In order to eliminate the effect of the histogram peak detection the true mixing parameters were directly passed to the mask construction and no source localization was performed.

Mask type	COR %	ACC %	DEL	SUB	INS
Without mask smoothing	88.62	75.37	17	92	127
With mask smoothing	94.57	93.53	12	40	10

Table 2. Recognition results in terms of HTK correctness (COR) and accuracy (ACC) score for missing data masks with and without median smoothing. The number of insertions (INS), deletions (DEL) and substitutions (SUB) is also given.

Clearly, if no median smoothing is applied to the DUET masks the recognized digit hypotheses contained a high number of insertion and substitution errors. Over 70% of the observed insertions were caused by the digit models "oh" and "eight". With the proposed median smoothing technique both the insertion and substitution errors were dramatically reduced resulting in an improved recognition performance.

5. Discussion

The experimental results reported here suggest that DUET might be used as an effective front-end for missing data speech recognition. Its simplicity, robustness and easy integration into existing ASR architecture are the main compelling arguments for the proposed model. It also fundamentally differs from other multi-channel approaches in the way it makes use of spatial information. Instead of filtering the corrupted signal to retrieve the sources (McCowan et al., 2000; Low et al., 2004, Seltzer et al., 2004a) the time-frequency plane is partitioned into disjoint regions each assigned to a particular source.

A key aspect of the model is the histogram peak detection. Here, we assumed prior knowledge about the number of speakers which should equal the number of peaks in the histogram. However, for a high number of simultaneous speakers the sparseness assumption becomes increasingly unrealistic and as a consequence sometimes histogram peaks are not pronounced enough in the data set. Forcing the peak detection algorithm to find an inadequate number of peaks will produce false localization results. Ultimately, the algorithm should be able to automatically detect the number of sources visible in the data which is usually denoted as unsupervised clustering. This would indeed make the source separation more autonomous and truly blind. However, unsupervised clustering is a considerably more difficult problem and is still an active field of research (Grira et al., 2004). Other attempts to directly cluster the attenuation and delay distributions using a statistical framework have been reported elsewhere (Araki et al., 2007; Mandel et al., 2006) and would lead to probabilistic mask interpretations.

A point of concern is the microphone distance d_{mic} that was kept very small to avoid phase ambiguities (Yilmaz & Rickard, 2004). Clearly, this limits the influence of the attenuation parameter (see Fig. 2a). Rickard (2007) has offered two extensions to overcome the small sensor spacing by using phase differentials or tiled histograms. Another option to consider

is the use of multiple microphone pairs or sensor arrays allowing for full three-dimensional source localization (Araki et al., 2006; Araki et al., 2007).

While the proposed median smoothing was highly successful in reducing spurious points in the time-frequency masks the filter was applied as a post-processing step only. Other more sophisticated methods that incorporate neighborhood information already into the mask assignment or the peak detection itself might be more appropriate. In particular, Markov Random Fields (Li, 2001) have been quite successful in the field of image processing but tend to be more complex and demanding in terms of computational resources. Other schemes for incorporating neighborhood information into clustering or mixture model learning are also readily available (Ambroise et al., 1997; Chuang et al., 2006). The advantage of the proposed post-processing scheme lies in its simplicity and relatively fast computation. Nevertheless, careful selection of the size of the median filter is required as otherwise the filter tends to remove too much energy of the target signal.

In regards to related work the overall architecture of our system is in line with previously proposed binaural CASA models. However, the DUET separation framework differs in some key aspects as it models human hearing mechanisms to a much lesser degree. Whereas Harding et al. (2006) and Roman et al. (2003) perform mask estimation for each critical band using supervised learning techniques, DUET blindly estimates these masks based on a simple frequency independent classification of attenuation and delay parameters. The spatial cues are extracted from STFT ratios which offer significant speedups over computationally expensive cross-correlation functions commonly used to compute binaural ITDs (see also Kim & Kil (2007) for an efficient method of binaural ITD estimation using zero-crossings). More importantly, Roman et al. (2003) need to recalibrate their system for each new spatial source configuration which is not required in our model. DUET also directly operates on the mixture signals and does not employ Head-Related-Transfer-Functions (HRTFs) or gammatone filterbanks for spectral analysis. However, we expect supervised source localization schemes to outperform DUET's simple histogram peak detection when angular separation angles between sources are small (0°-15°).

In terms of ASR performance we achieved comparable results to Roman et al. (2003), in that the estimated masks matched the performance of the oracle masks. Recognition accuracy remained close to the upper bound for up to three simultaneous speech maskers. While other studies (Roman et al., 2003; Mandel et al., 2006) have reported inferior localization performance of DUET even for anechoic, two or three source configurations we can not confirm these observations based on the experimental results discussed here. Mandel et al. (2006) offer a possible explanation for this discrepancy by stating that DUET was designed for a closely spaced omni-directional microhone pair and not the dummy head recordings used in binaural models.

Finally, we acknowledge that the results presented here were obtained under ideal conditions that met most of the requirements of the DUET algorithm. In particular the noise-free and anechoic environment can be considered as strong simplifications of real acoustic scenes and it is expected that under more realistic conditions the parameter estimation using DUET will fail. Future work is required to make the estimators more robust in hostile environments. To this extent, it is also tempting to combine the DUET parameters with other localization methods (Kim & Kil, 2007) or non-spatial features such as harmonicity cues (Hu & Wang, 2004). However, the integration of additional cues into the framework outlined here remains a topic for future research.

6. Conclusion

This chapter has investigated the DUET blind source separation technique as a front-end for missing data speech recognition in anechoic multi-talker environments. Using the DUET attenuation and delay estimators time-frequency masks were constructed by exploiting the sparseness property of speech in the frequency domain. The obtained masks were then smoothed with a median filter to remove spurious points that can cause insertion errors in the speech decoder. Finally, the frequency resolution was reduced by applying a triangular mel-filter weighting which makes the masks more suitable for speech recognition purposes. The experimental evaluation showed that the proposed model is able to retain high recognition performance in the presence of multiple competing speakers. For up to three simultaneous speech maskers the estimated soft masks closely matched the recognition performance of the oracle masks designed with a priori knowledge of the source spectra. In our future work we plan to extend the system to handle reverberant environments through the use of multiple sensor pairs and by combining the T-F masking framework with spatial filtering techniques that can enhance the speech signal prior to recognition.

7. Acknowledgments

This work was supported in part by The University of Western Australia, Australia and in part by National ICT Australia (NICTA). NICTA is funded through the Australian Government's Backing Australia's Ability Initiative, in part through the Australian Research Council.

8. References

Ambroise, C. ; Dang, V. & Govaert, G. (1997). Clustering of Spatial Data by the EM Algorithm, In: *geoENV I - Geostatistics for Environmental Applications,* Vol. 9, Series: Quantitative Geology and Geostatistics, pp. 493-504, Kluwer Academic Publisher

Araki, S.; Sawada, H.; Mukai, R. & Makino, S. (2005). A novel blind source separation method with observation vector clustering, *International Workshop on Acoustic Echo and Noise Control*, Eindhoven, The Netherlands, 2005

Araki, S.; Sawada, H.; Mukai, R. & Makino, S. (2006). DOA Estimation for Multiple Sparse Sources with Normalized Observation Vector Clustering, *IEEE International Conference on Acoustics, Speech and Signal Processing,* Toulouse, France, 2006

Araki, S.; Sawada, H.; Mukai, R. & Makino, S. (2006a). Blind Sparse Source Separation with Spatially Smoothed Time-Frequency Masking, *International Workshop on Acoustic Echo and Noise Control,* Paris, France, 2006

Araki, S.; Sawada, H.; Mukai, R. & Makino, S. (2007). Underdetermined blind sparse source separation for arbitrarily arranged multiple sensors, *Signal Processing,* Vol. 87, No. 8, pp. 1833-1847

Barker, J.; Josifovski, L.; Cooke, M. & Green, P. (2000). Soft decisions in missing data techniques for robust automatic speech recognition, *Proccedings of the 6th International Conference of Spoken Language Processing,* Beijing, China, 2000

Bregman, A. (1990). Auditory Scene Analysis, MIT Press, Cambridge MA., 1990

Brown, G. & Cooke, M. (1994). Computational auditory scene analysis, *Computer Speech and Language,* Vol. 8, No. 4, pp. 297-336

Cerisara, C.; Demangea, S. & Hatona, J. (2007). On noise masking for automatic missing data speech recognition: A survey and discussion, *Speech Communication*, Vol. 21, No. 3, (2007), pp. 443-457

Cherry, E. (1953). Some experiments on the recognition of speech, with one and with two ears, *Journal of Acoustical Society of America,* Vol. 25, No. 5, (1953), pp. 975-979

Chuang, K.; Tzeng, H.; Chen, S.; Wu, J. & Chen, T. (2006). Fuzzy c-means clustering with spatial information for image segmentation, *Computerized Medical Imaging and Graphics,* Vol. 30, No. 1, pp. 9-15

Cooke, M.; Green, P.; Josifovski, L. & Vizinho, A. (2001). Robust Automatic Speech Recognition with missing and unreliable acoustic data, *Speech Communication,* Vol. 34, No. 3, (2001), pp. 267-285

de Veth, J. ; de Wet, F., Cranen, B. & Boves, L. (2001). Acoustic features and a distance measure that reduces the impact of training-set mismatch in ASR, *Speech Communication,* Vol. 34, No. 1-2, (2001), pp. 57-74

El-Maliki, M. & Drygajlo, A. (1999). Missing Features Detection and Handling for Robust Speaker Verification, *Proceedings of Eurospeech,* Budapest, Hungary, 1999

Garofolo, J.; Lamel, L.; Fisher, W.; Fiscus, J.; Pallett, D.; Dahlgren, N. & Zue, V. (1993). TIMIT Acoustic-Phonetic Continuous Speech Corpus, *Linguistic Data Consortium,* Philadelphia, USA

Grira, N.; Crucianu, M. & Boujemaa, N. (2004). Unsupervised and Semi-supervised Clustering: a Brief Survey, In: *A Review of Machine Learning Techniques for Processing Multimedia Content,* MUSCLE European Network of Excellence, 2004

Harding, S.; Barker, J. & Brown, G. (2005). Mask Estimation Based on Sound Localisation for Missing Data Speech Recognition, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, USA, 2005

Harding, S.; Barker, J. & Brown, G. (2006). Mask estimation for missing data speech recognition based on statistics of binaural interaction, *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, No. 1, (2006), pp. 58-67

Harte, N.; Hurley, N.; Fearon, C. & Rickard, S. (2005). Towards a Hardware Realization of Time-Frequency Source Separation of Speech, *European Conference on Circuit Theory and Design,* Cork, Ireland, 2005

Hu, G. & Wang, D. (2004). Monaural Speech Segregation Based on Pitch Tracking and Amplitude Modulation, *IEEE Transactions on Neural Networks,* Vol. 15, No. 5, (2004), pp. 1135-1150

Hyvärinen, H. (1999). Survey on Independent Component Analysis, *Neural Computing Surveys,* Vol. 2, (1999), pp. 94-128

Kim, Y. & Kil, R. (2007). Estimation of Interaural Time Differences Based on Zero-Crossings in Noisy Multisource Environments, *IEEE Transactions on Audio, Speech, and Language Processing,* Vol. 15, No. 2, (2007), pp. 734-743

Kolossa, D.; Sawada, H.; Astudillo, R.; Orglmeister, R. & Makino, S. (2006). Recognition of Convolutive Speech Mixtures by Missing Feature Techniques for ICA, *Asilomar Conference on Signals, Systems and Computers,* Pacific Grove, CA, 2006

Kühne, M.; Togneri, R. & Nordholm, S. (2007). Mel-Spectrographic Mask Estimation for Missing Data Speech Recognition using Short-Time-Fourier-Transform Ratio Estimators, *IEEE International Conference on Acoustics, Speech and Signal Processing,* Honolulu, USA, 2007

Kühne, M.; Togneri, R. & Nordholm, S. (2007a). Smooth soft mel-spectrographic masks based on blind sparse source separation, *Proceedings of Interspeech 2007*, Antwerp, Belgium, 2007

Leonard, R. (1984). A database for speaker-independent digit recognition, *IEEE International Conference on Acoustics, Speech, and Signal Processing,* San Diego, USA, 1984

Li, S. (2001). Markov Random Field Modeling in Image Analysis, Springer-Verlag, 2001

Low, S.; Togneri, R. & Nordholm, S. (2004). Spatio-temporal processing for distant speech recognition, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, 2004

Mandel, M.; Ellis, D. & Jebara, T. (2006). An EM Algorithm for Localizing Multiple Sound Sources in Reverberant Environments, *Twentieth Annual Conference on Neural Information Processing Systems,* Vancouver, B.C., Canada, 2006

McCowan, I.; Marro, C. & Mauuary, L. (2000). Robust Speech Recognition Using Near-Field Superdirective Beamforming with Post-Filtering, *IEEE International Conference on Acoustics, Speech and Signal Processing,* Istanbul, Turkey, 2000

Moore, B. (2003). An introduction to the psychology of hearing, Academic Press, San Diego, CA

Morris, A.; Barker, J. & Bourlard, H. (2001). From missing data to maybe useful data: soft data modelling for noise robust ASR, *WISP,* Stratford-upon-Avon, England, 2001

Raj, B. & Stern, R. (2005). Missing-feature approaches in speech recognition, *IEEE Signal Processing Magazine,* Vol. 22, No. 5, (2005), pp. 101-116

Rickard, S. (2007). The DUET Blind Source Separation Algorithm, In: *Blind Speech Separation,* Makino, S.; Lee, T.-W.; Sawada, H., (Eds.), Springer-Verlag, pp. 217-237

Roman, N.; Wang, D. & Brown, G. (2003). Speech segregation based on sound localization, *Journal of the Acoustical Society of America,* Vol. 114, No. 4, (2003), pp. 2236-2252

Russ, J. (1999). The Image Processing Handbook, CRC & IEEE, 1999

Seltzer, M.; Raj, B. & Stern, R. (2004). A Bayesian classifier for spectrographic mask estimation for missing feature speech recognition, *Speech Communication*, Vol. 43, No. 4, (2004), pp. 379-393

Seltzer, M.; Raj, B. & Stern, R. (2004a). Likelihood-Maximizing Beamforming for Robust Hands-Free Speech Recognition, *IEEE Transactions on Speech and Audio Processing,* Vol. 12, No. 5, (2004), pp. 489-498

Van Hamme, H. (2004). Robust speech recognition using cepstral domain missing data techniques and noisy masks, *IEEE International Conference on Acoustics, Speech, and Signal Processing,* Montreal, Canada, 2004

Wang, D. (2005). On Ideal Binary Mask As the Computational Goal of Auditory Scene Analysis, In: *Speech Separation by Humans and Machine,* Divenyi, P., pp. 181-197, Kluwer Academic

Yilmaz, Ö. & Rickard, S. (2004). Blind Separation of Speech Mixtures via Time-Frequency Masking, *IEEE Transactions on Signal Processing,* Vol. 52, No. 7, (2004), pp. 1830-1847

Young, S.; Evermann, G.; Gales, M.; Hain, T.; Kershaw, D.; Liu, X.; Moore, G.; Odell. J.; Ollason, D.; Povey, D.; Valtchev, V. & Woodland, P. (2006). The HTK Book, *Cambridge University Engineering Department*, 2006

5

Dereverberation and Denoising Techniques for ASR Applications

Fernando Santana Pacheco and Rui Seara
Federal University of Santa Catarina
Brazil

1. Introduction

Over the last few years, advances in automatic speech recognition (ASR) have motivated the development of several commercial applications. Automatic dictation systems and voice dialing applications, for instance, are becoming ever more common. Despite significant advances, one is still far from the goal of unlimited speech recognition, i.e., recognition of any word, spoken by any person, in any place, and by using any acquisition and transmission system. In real applications, the speech signal can be contaminated by different sources of distortion. In hands-free devices, for instance, effects of reverberation and background noise are significantly intensified with the larger distance between the speaker and the microphone. If such distortions are uncompensated, the accuracy of ASR systems is severely hampered (Droppo & Acero, 2008). In the open literature, several research works have been proposed aiming to cope with the harmful effects of reverberation and noise in ASR applications (de la Torre et al., 2007; Huang et al., 2008). Summarizing, current approaches focusing on ASR robustness to reverberation and noise can be classified as model adaptation, robust parameterization, and speech enhancement.

The goal of this chapter is to provide the reader with an overview of the current state of the art about ASR robustness to reverberation and noise, as well as to discuss the use of a particular speech enhancement approach trying to circumvent this problem. For such, we choose to use spectral subtraction, which has been proposed in the literature to enhance speech degraded by reverberation and noise (Boll, 1979; Lebart & Boucher, 1998; Habets, 2004). Moreover, taking into consideration that ASR systems share similar concerns about this problem, such an approach has also been applied successfully as a preprocessing stage in these applications.

This chapter is organized as follows. Section 2 characterizes the reverberation and noise effects over speech parameters. An overview of methods to compensate reverberation and noise in ASR systems is briefly discussed in Section 3, including classification and comparison between different approaches. A discussion of spectral subtraction applied to reverberation reduction is presented in Section 4. In that section we examine how to adjust the parameters of the algorithm; we also analyze the sensitivity to estimation errors and changes in the room response. The combined effect of reverberation and noise is also assessed. Finally, concluding remarks are presented in Section 5.

2. Reverberation and noise

Speech communication is so natural to humans that we usually do not perceive some effects. Before reaching a microphone or the listener's ears, speech signals may be modified by the medium in which they are propagating (enclosure). In an ideal anechoic chamber, the signal follows only one path from the source to the receiver. But in typical rooms, surfaces (walls and furniture) reflect the emitted sound; the microphone receives a stream of reflected signals from multiple propagation paths. The whole set of reflections is termed reverberation. Although in this chapter we shall discuss methods to reduce this effect, reverberation is not detrimental at all times. It may give the listener the spatial impression of the enclosure (Everest, 2001); it also increases both the "liveness" and "warmth" of the room, especially important in music. On the other hand, reverberation in excess causes loss of intelligibility and clarity, harming communication or musical performance.

The effect of reverberation can be modeled as the processing of a signal by a linear time invariant system. This operation is represented by the convolution between the room impulse response (RIR) and the original signal, expressed as

$$y(n) = x(n) * h(n) \tag{1}$$

where $y(n)$ represents the degraded speech signal, $x(n)$, the original (without degradation) speech signal, $h(n)$ denotes the room impulse response, and $*$ characterizes the linear convolution operation.

In this approach, room reverberation is completely characterized by the RIR. Fig. 1 shows a typical impulse response measured in a room. A RIR can be usually separated into three parts: the direct response, initial reflections, and late reverberation. The amount of energy and delay of each reflection causes different psychoacoustic effects. Initial reflections (or early reverberation) are acoustically integrated by the ears, reinforcing the direct sound. Since initial reflections do not present a flat spectrum, a coloration of the speech spectrum occurs (Huang et al., 2008). Late reverberation (or reverberation tail) causes a different effect called overlap masking. Speech signals exhibit a natural dynamics with regions presenting noticeably different energy levels, as occurs between vowels and consonants. Reverberation tail reduces this dynamics, smearing the energy over a large interval and masking lower energy sounds.

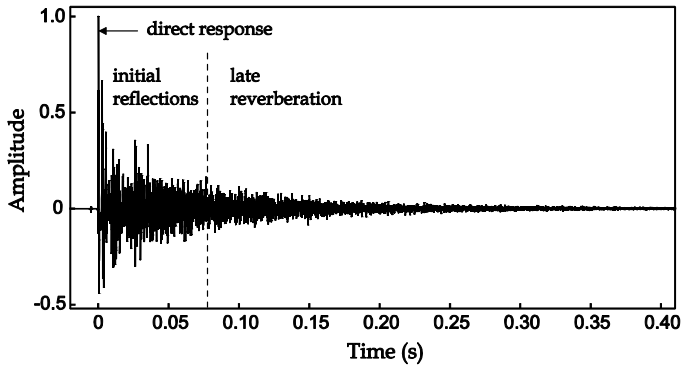


Fig. 1. A typical room impulse response.

It is worth to examine here a real-world example. Fig. 2(a) and (b) illustrate, respectively, the speech signal corresponding to the utterance "enter fifty one" and the associated spectrogram. Notice in these figures the mentioned dynamics in time and the clear harmonic structure with speech resonances marked by darker lines in the spectrogram [see Fig. 2(b)]. Reverberation is artificially incorporated to the original speech signal, by convolving this speech segment with the RIR displayed in Fig. 1. Fig. 2(c) and (d) show, respectively, the reverberant version and the corresponding spectrogram. Now, observe in Fig. 2(c) that the signal is smeared in time, with virtually no gap between phonemes. In addition, notice the difficulty to identify the resonances in Fig. 2(d).

So, how to measure the level of reverberation or how to assess the acoustic quality of a room? Much research has been carried out to define objective parameters correlated with the overall quality and subjective impression exhibited by a room. In this chapter, we present two important parameters used to measure the level of reverberation of an enclosure: reverberation time and early to late energy ratio.

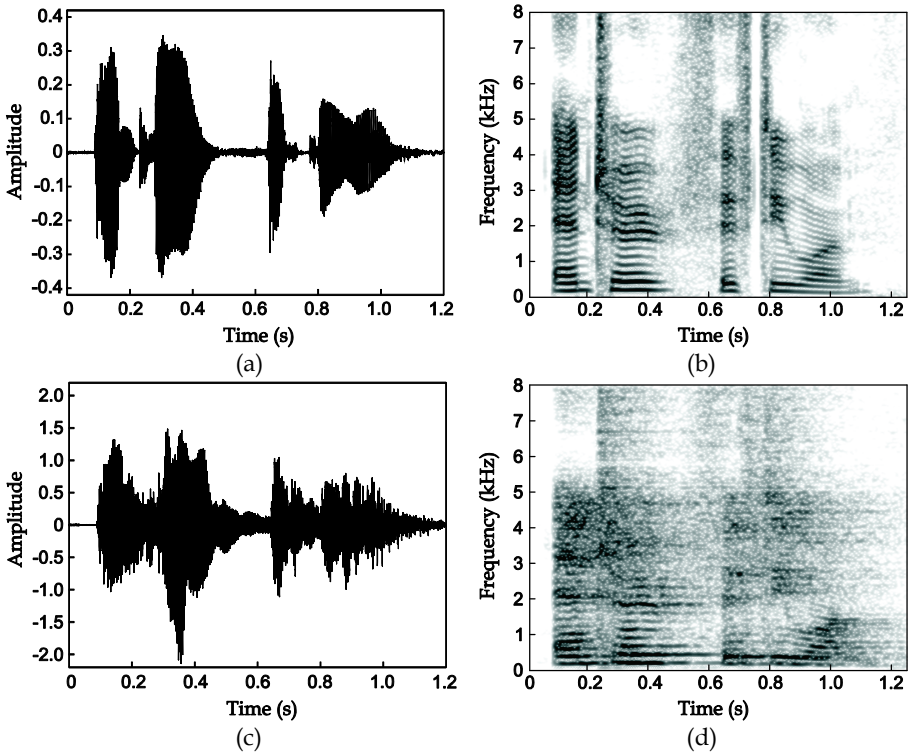


Fig. 2. Reverberation effect over a speech signal. (a) Original speech signal corresponding to the utterance "enter fifty one" and (b) associated spectrogram. (c) Reverberated version of the same previous signal and (d) corresponding spectrogram.

Reverberation time (T_{60}, RT_{60} or $RT60$) is defined as the time interval required for the reverberation to decay 60 dB from the level of a reference sound. It is physically associated with the room dimensions as well as with the acoustic properties of wall materials. The

measurement of the reverberation time is computed through the decay curve obtained from the RIR energy (Everest, 2001). The result can be expressed in terms of either a broadband measure or a set of values corresponding to frequency-dependent reverberation times (for example, RT_{500} corresponds to the reverberation time at the frequency band centered in 500 Hz). To give the reader an idea of typical values, office rooms present T_{60} between 200 ms and 600 ms while large churches can exhibit T_{60} in the order of 3 s (Everest, 2001).

Another objective indicator of speech intelligibility or music clarity is called early to late energy ratio (speech) or clarity index (music) (Chesnokov & SooHoo, 1998), which is defined as

$$C_T = 10\log_{10}\frac{\int_0^T p^2(t)}{\int_T^\infty p^2(t)} \tag{2}$$

where $p(t)$ denotes the instantaneous acoustic pressure and T is the time instant considered as the threshold between early and late reverberation. For speech intelligibility evaluation, it is usual to consider C_{50} ($T = 50$ ms), while C_{80} is a measure for music clarity (Chesnokov & SooHoo, 1998).

Now, considering the separation between early and late reverberation, $h(n)$ can be expressed as

$$h(n) = \begin{cases} 0, & n < 0 \\ h_{\mathrm{d}}(n), & 0 \le n \le N_{\mathrm{d}} \\ h_{\mathrm{r}}(n), & n > N_{\mathrm{d}} \end{cases} \tag{3}$$

where $h_{\mathrm{d}}(n)$ denotes the part of the impulse response corresponding to the direct response plus early reverberation, $h_{\mathrm{r}}(n)$, the other part of the response relating to the late reverberation, $N_{\mathrm{d}} = f_{\mathrm{s}} T$ is the number of samples of the response $h_{\mathrm{d}}(n)$, and f_{s}, the considered sampling rate.

Besides reverberation, sound is also subjected to degradation by additive noise. Noise sources, such as fans, motors, among others, may compete with the signal source in an enclosure. Thus, we should also include the noise effect over the degraded signal model $y(n)$, rewriting (1) now as

$$y(n) = x(n) * h(n) + v(n) \tag{4}$$

where $v(n)$ represents additive noise.

3. Overview of dereverberation and denoising methods

Before introducing methods to tackle reverberation and noise, we present a brief overview of current state-of-the-art ASR technology. The current generation of ASR systems is based on a statistical framework (Rabiner & Juang, 2008). It means that, before system deployment (or test), a first phase of training is mandatory. During training, a set of models is estimated

considering text-labeled utterances (speech corpus and associated transcriptions). Thereby, each model represents a reference pattern of each base unit (word or phoneme, for instance). In order to recognize a given speech signal, the system evaluates the similarity (likelihood score) between the signal and each previously trained model. The most likely word sequence is obtained as an outcome of the process.

During training, models also incorporate acoustic characteristics from the recording, such as reverberation and noise levels. If the system is deployed under similar conditions, one says that training and test are matched and high recognition rate may be expected. Unfortunately, these conditions can differ from the training phase to the effective use, leading to an important acoustic mismatch, and impairing the ASR performance. Considering a real case, if models are trained with clean speech, recorded in a studio, and the system is used for dictation at a noisy office room, the recognition rate may be degraded. Therefore, to improve the system robustness, the mismatch problem between training and test must be tackled. Over the last few decades, a considerable research effort has been directed for reducing the mismatch caused by reverberation and additive noise.

There are several ways to classify existing approaches to the reverberation and noise problems in ASR systems. In this chapter, we choose to group methods based on their location in the speech recognition chain. ASR processing can be roughly separated into two parts: front-end and back-end. Speech parameters are extracted at the front-end module, whereas the likelihood between the input signal and acoustic models is computed at the back-end (or decoder). Considering this classification, the mismatch caused by reverberation and noise can be reduced either before the front-end, or during the front-end processing or even at the back-end module, as shown in Fig. 3. Therefore, methods are grouped into the following classes: speech enhancement, robust parameterization, and model adaptation. Each group is discussed in the following.

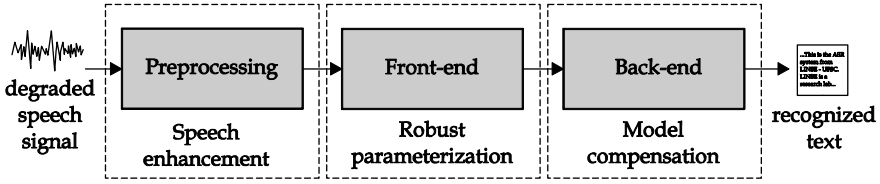


Fig. 3. Classification of methods used in speech recognition applications for reverberation and noise compensation.

3.1 Speech enhancement

Speech enhancement methods attempt to cope with reverberation and noise problems before the signal reaches the front-end. They work as a preprocessing stage in ASR systems.

Methods in this category can be broadly classified by the number of microphones they need to operate, leading to two classes: single and multi-microphone methods (the latter is also termed microphone array).

We briefly describe here some current techniques: beamforming, inverse filtering, kurtosis-based adaptive filtering, harmonicity-based dereverberation, and spectral subtraction.

Beamforming is a classical microphone array approach (Darren et al., 2001). Signals from each microphone are accurately delayed and combined (by a simple sum or a filtering algorithm). As a consequence, the involved algorithm directs the array to the speech source,

reinforcing the speech signal and reducing reverberation and noise from other directions. Although an increase in recognition rate is achieved for noisy speech, the same good effect is not attained for reverberant speech, because conventional microphone array algorithms assume that the target and the undesired signals are uncorrelated (not true for reverberation). In a recent approach, called likelihood maximizing beamforming (LIMABEAM) (Seltzer et al., 2004), the beamforming algorithm is driven by the speech recognition engine. This approach has demonstrated a potential advantage over standard beamforming techniques for ASR applications.

Methods based on inverse filtering have two processing stages: estimation of the impulse responses between the source and each microphone and application of a deconvolution operation. Among other approaches, estimation can be carried out by cepstral techniques (Bees et al., 1991) or a grid of zeros (Pacheco & Seara, 2005); however, some practical difficulties have been noted in real applications, impairing the correct working of these techniques. Regarding the inversion of the RIR, an efficient approach is proposed by Radlović & Kennedy (2000), which overcomes the drawbacks due to nonminimum phase characteristics present in real-world responses.

Another interesting approach is presented by Gillespie et al. (2001), in which characteristics of the speech signal are used for improving the dereverberation process. There, the authors have demonstrated that the residue from a linear prediction analysis of clean speech exhibits peaks at each glottal pulse while those ones are dispersed in reverberant speech. An adaptive filter can be used for minimizing this dispersion (measured by kurtosis), reducing the reverberation effect. The same algorithm is also used as a first stage of processing by Wu & Wang (2006), showing satisfactory results for reducing reverberation effects when T_{60} is between 0.2 and 0.4 s.

The harmonic structure of the speech signal can be used in harmonicity based dereveberation (HERB) (Nakatani et al., 2007). In this approach, it is assumed that the original signal is preserved at multiples of the fundamental frequency, and so an estimate of room response can be obtained. The main drawback of this technique is the amount of data needed for achieving a good estimate.

Spectral subtraction is another speech enhancement technique, which will be discussed in details in Section 4.

3.2 Robust acoustic features

In this class of techniques, the central idea is to represent the signal with parameters less sensitive to changes in acoustic conditions as reverberation and noise.

A very simple and widespread approach is called cepstral mean normalization (CMN). In this technique, a mean of the parameter vectors is initially obtained. The resulting mean vector is subtracted from each parameter vector. Therefore, the normalized parameters present a long-term average equal to zero. It is possible to demonstrate that this approach improves the robustness with respect to the linear filtering effect introduced by microphones and transmission channels over the speech signal (Droppo & Acero, 2008). It has also been verified experimentally that the CMN reduces the additive noise effect, even though it is ineffective for dereverberation (Droppo & Acero, 2008).

Regarding the reverberation problem, some authors suggest that it cannot be identified within short frames of analysis (in the order of 25 ms), since RIRs usually exhibit large lengths. Two approaches attempt to overcome this problem: relative spectra (RASTA) (Hermansky & Morgan, 1994) and modulation spectrogram (MSG) (Kingsbury, 1998). They

consider slow variations in spectrum, which is verified when a frame size in the order of 200 ms is used. ASR assessments have shown that such approaches improve the recognition accuracy for moderately reverberant conditions (Kingsbury, 1998).
An alternative parameterization technique, named missing feature approach (Palomäki et al., 2004; Raj & Stern, 2005), suggests representing the input signal in a time-frequency grid. Unreliable or missing cells (due to degradation) are identified and discarded or even replaced by an estimate of the clean signal. In the case of reverberation, reliable cells are those in which the direct signal and initial reflections are stronger. Training is carried out with clean speech and there is no need to keep retraining acoustic models for each kind of degradation. So, the identification of unreliable cells is performed only during recognition. A considerable improvement in the recognition rate may be attained; however, to obtain such identification of cells is a very hard task in practice.

3.3 Model adaptation

The main objective of model adaptation approaches is to minimize the mismatch between training and test phases by applying some kind of compensation over the reference model.
The first approach is to include reverberation and noise during training, i.e., contaminating the training material with the same kind of degradation expected for deployment. Reverberation and noise can be recorded during the acquisition of speech corpora or even to be artificially included. International projects have recorded training material in different conditions, such as inside cars in the SpeechDat-Car project (Moreno et al., 2000) or in different environment in the SpeeCon project (Iskra et al., 2002).
Artificial inclusion of reverberation allows generating models with different levels of reverberation (Couvreur & Couvreur, 2004), permitting thus to select the best model match during deployment.
As an alternative to retrain models for each noise condition, the parallel model combination (PMC) technique can be applied. This approach attempts to estimate a noisy speech model from two other models: a previously trained one, based on clean speech, and a noise model, obtained by an on-line estimate from noise segments (Gales & Young, 1995). Promising adaptation results can be achieved by using a small amount of data, whereas the main drawback of the PMC approach is a large computational burden.
A better adjustment can also be accomplished with a set of adaptation data in a maximum *a posteriori* estimation approach (Omologo et al., 1998). A significant increase in recognition rate is achieved, even though a single microphone is used for signal acquisition; however, the robustness to changes in the environmental conditions is still a challenging issue (Omologo et al., 1998).

4. Spectral subtraction

Spectral subtraction is a well-known speech enhancement technique, which is part of the class of short-time spectral amplitude (STSA) methods (Kondoz, 2004). What makes spectral subtraction attractive is its simplicity and low computational complexity, being advantageous for platforms with limited resources (Droppo & Acero, 2008).

4.1 Algorithm

Before introducing spectral subtraction as a dereverberation approach, we shall review its original formulation as a noise reduction technique. Disregarding the effect of reverberation, a noisy signal in (4) can be expressed in frequency domain as

$$Y(k) = X(k) + V(k) \tag{5}$$

where $Y(k)$, $X(k)$, and $V(k)$ denote the short-time discrete Fourier transform (DFT) of $y(n)$, $x(n)$ and $v(n)$, respectively. The central idea of spectral subtraction is to recover $x(n)$ modifying only the magnitude of $Y(k)$. The process can be described as a spectral filtering operation

$$\left|\hat{X}(k)\right|^{\nu} = G(k)\left|Y(k)\right|^{\nu} \tag{6}$$

where ν denotes the spectral order, $\hat{X}(k)$ is the DFT of the enhanced signal $\hat{x}(n)$, and $G(k)$ is a gain function.

Fig. 3 shows a block diagram of a general procedure of spectral subtraction. The noisy signal $y(n)$ is windowed and its DFT is computed. The gain function is then estimated by using the current noisy magnitude samples, the previous enhanced magnitude signal and the noise statistics. Note that the phase of $Y(k)$ [represented by $\angle Y(k)$] remains unchanged, being an input to the inverse DFT (IDFT) block. The enhanced signal is obtained associating the enhanced magnitude and the phase of $Y(k)$, processing them by the IDFT block along with an overlap-and-add operation; the latter to compensate for the windowing.

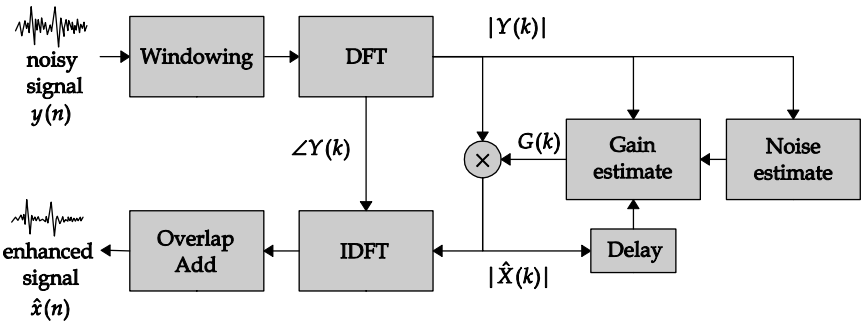


Fig. 3. Block diagram of a general procedure of spectral subtraction.

The blocks of gain and noise estimates are the most critical part in the process and the success of this technique is strongly dependent on determining adequate gains. In the following, we shall discuss this approach considering a power spectral subtraction example. Processing signals in the power spectral domain, i.e., $\nu = 2$, and assuming that signal and noise are uncorrelated, we have

$$\left|\hat{X}(k)\right|^2 = \left|Y(k)\right|^2 - \left|\hat{V}(k)\right|^2 \tag{7}$$

or even

$$\left|\hat{X}(k)\right|^2 = G(k)\left|Y(k)\right|^2 \tag{8}$$

for which the most simple estimate of the gain $G(k)$ is given by

$$G(k)=\begin{cases}1-\dfrac{1}{\mathrm{SNR}(k)}, & \mathrm{SNR}(k)>1\\ 0, & \text{otherwise}\end{cases} \tag{9}$$

with

$$\mathrm{SNR}(k)=\frac{|Y(k)|^2}{|\hat{V}(k)|^2} \tag{10}$$

where $\mathrm{SNR}(k)$ is the *a posteriori* signal-to-noise ratio and $\hat{V}(k)$ is the noise estimate. Although necessary to prevent $|\hat{X}(k)|$ from being negative, the clamping introduced by the conditions in (9) causes some drawbacks. Note that gains are estimated for every frame and at each frequency index independently. Observing the distribution of these gains in a time-frequency grid, one notes that neighbor cells may display varying levels of attenuation. This irregularity over the gain gives rise to tones at random frequencies that appear and disappear rapidly (Droppo & Acero, 2008), leading to an annoying effect called musical noise.
More elaborate estimates for $G(k)$ are proposed in the literature, aiming to reduce musical noise. An improved approach to estimate the required gain is introduced by Berouti et al. (1979), which is given by

$$G(k)=\max\left\{\left[1-\alpha\left(\frac{1}{\mathrm{SNR}(k)}\right)^{\frac{\nu}{2}}\right]^{\frac{1}{\nu}},\ \beta\right\} \tag{11}$$

where α and β are, respectively, the oversubtraction and spectral floor factors. The oversubtraction factor controls the reduction of residual noise. Lower levels of noise are attained with higher α; however, if α is too large, the speech signal will be distorted (Kondoz, 2004). The spectral floor factor works to reduce the musical noise, smearing it over a wider frequency band (Kondoz, 2004). A trade-off in β choice is also required. If β is too large, other undesired artifacts become more evident.
It is important to point out that speech distortion and residual noise cannot be reduced simultaneously. Moreover, parameter adjustment is dependent on the application. It has been determined experimentally that a good trade-off between noise reduction and speech quality is achieved with power spectral subtraction $(\nu=2)$ by using α between 4 and 8, and $\beta=0.1$ (Kondoz, 2004). This set-up is considered adequate for human listeners, since, as a general rule, human beings can tolerate some distortion, but are sensitive to fatigue caused by noise. We shall show in Section 4.4 that ASR systems usually are more susceptible to speech distortion, and so $\alpha<1$ could be a better choice for reducing the recognition error rate.

4.2 Application of spectral subtraction for dereverberation

An adaptation of spectral subtraction has been recently proposed to enhance speech degraded by reverberation (Lebart & Boucher, 1998; Habets, 2004). It will be discussed in details later on.

In order to tackle room reverberation by using spectral subtraction, some fundamental relations must be established. Firstly, the autocorrelation $r_y(\ell)$ of the reverberant signal is defined. Therefore, disregarding the additive noise effect, we get

$$r_y(\ell) \equiv r_y(n, n+\ell) = \mathrm{E}[y(n)y(n+\ell)] = \mathrm{E}[\sum_{k=-\infty}^{n} x(k)h(n-k) \sum_{m=-\infty}^{n+\ell} x(m)h(n+\ell-m)]. \tag{12}$$

Given the nature of the speech signal and of the RIR, one can consider $x(n)$ and $h(n)$ as independent statistical processes. Thus,

$$r_y(\ell) = \sum_{k=-\infty}^{n} \sum_{m=-\infty}^{n+\ell} \mathrm{E}[x(k)x(m)]\mathrm{E}[h(n-k)h(n+\ell-m)]. \tag{13}$$

Considering a RIR modeled by modulating a zero-mean random sequence with a decaying exponential (Lebart & Boucher, 1998), one can write

$$h(n) = w(n)\,\mathrm{e}^{-\tau n}u(n) \tag{14}$$

where $w(n)$ represents a white zero-mean Gaussian noise with variance σ_w^2, $u(n)$ denotes the unit step function, and τ is a damping constant related to the reverberation time, which is expressed as (Lebart & Boucher, 1998)

$$\tau = \frac{3\ln 10}{T_{60}}. \tag{15}$$

Thus, the second r.h.s. term in (13) is written as

$$\mathrm{E}[h(n-k)h(n+\ell-m)] = \mathrm{e}^{-2\tau n}\,\sigma_w^2\,\mathrm{e}^{\tau(k+m-\ell)}\,\delta(k-m+\ell) \tag{16}$$

where $\delta(n)$ represents the unit sample sequence.
Then, substituting (16) into (13), we obtain

$$r_y(\ell) = \mathrm{e}^{-2\tau n} \sum_{k=-\infty}^{n} \mathrm{E}[x(k)x(k+\ell)]\sigma_w^2\,\mathrm{e}^{2\tau k}. \tag{17}$$

Now, considering the threshold N_d, defined in (3), one can split the summation in (17) into two parts. Thereby,

$$r_y(\ell) = \mathrm{e}^{-2\tau n} \sum_{k=-\infty}^{n-N_\mathrm{d}} \mathrm{E}[x(k)x(k+\ell)]\sigma_w^2\,\mathrm{e}^{2\tau k} + \mathrm{e}^{-2\tau n} \sum_{k=n-N_\mathrm{d}+1}^{n} \mathrm{E}[x(k)x(k+\ell)]\sigma_w^2\,\mathrm{e}^{2\tau k}. \tag{18}$$

In addition, the autocorrelation of the $y(n)$ signal, computed between the samples $n-N_\mathrm{d}$ and $n-N_\mathrm{d}+\ell$, can be written as

$$r_y(n-N_\mathrm{d}, n-N_\mathrm{d}+\ell) = \mathrm{e}^{-2\tau(n-N_\mathrm{d})} \sum_{k=-\infty}^{n-N_\mathrm{d}} \mathrm{E}[x(k)x(k+\ell)]\sigma_w^2\,\mathrm{e}^{2\tau k}. \tag{19}$$

Then, from (18), the autocorrelation between the samples n and $n+\ell$ is given by

$$r_y(n,n+\ell) = r_{y_\mathrm{r}}(n,n+\ell) + r_{y_\mathrm{d}}(n,n+\ell) \tag{20}$$

with

$$r_{y_\mathrm{r}}(n,n+\ell) = \mathrm{e}^{-2\tau N_\mathrm{d}}\, r_y(n-N_\mathrm{d}, n-N_\mathrm{d}+\ell) \tag{21}$$

and

$$r_{y_\mathrm{d}}(n,n+\ell) = \mathrm{e}^{-2\tau n} \sum_{k=n-N_\mathrm{d}+1}^{n} \mathrm{E}[x(k)x(k+\ell)]\sigma_w^2\, \mathrm{e}^{2\tau k} \tag{22}$$

where $r_{y_\mathrm{r}}(n,n+\ell)$ and $r_{y_\mathrm{d}}(n,n+\ell)$ are the autocorrelation functions associated with the signals $y_\mathrm{r}(n)$ and $y_\mathrm{d}(n)$, respectively. Signal $y_\mathrm{r}(n)$ is related to the late reverberation, as a result of the convolution of $h_\mathrm{r}(n)$ and $x(n)$. Variable $y_\mathrm{d}(n)$ is associated with the direct signal and initial reflections, being obtained through the convolution of $h_\mathrm{d}(n)$ and $x(n)$.
Now, from (20), the short-time power spectral density (PSD) of the degraded signal $S_y(n,k)$ is expressed as

$$S_y(n,k) = S_{y_\mathrm{r}}(n,k) + S_{y_\mathrm{d}}(n,k) \tag{23}$$

where $S_{y_\mathrm{r}}(n,k)$ and $S_{y_\mathrm{d}}(n,k)$ are the PSDs corresponding to the signals $y_\mathrm{r}(n)$ and $y_\mathrm{d}(n)$, respectively. From (21), the estimated value $S_{y_\mathrm{r}}(n,k)$ is obtained by weighting and delaying the PSD of the degraded speech signal. Thus,

$$S_{y_\mathrm{r}}(n,k) = \mathrm{e}^{-2\tau N_\mathrm{d}}\, S_y(n-N_\mathrm{d},k). \tag{24}$$

Then, assuming that $y_\mathrm{d}(n)$ and $y_\mathrm{r}(n)$ are uncorrelated, the late reverberant signal can be treated as an additive noise, and the direct signal can be recovered through spectral subtraction.

4.3 Estimation of reverberation time

In order to implement the previously presented procedure, one initially must obtain the parameter τ, since it is used for estimating the power spectral density of the late reverberant signal (24). Given that τ is related to the reverberation time, one estimates T_{60} from the captured signal.
Some approaches have been proposed recently for blind estimation of the reverberation time. In this case, blind means that only the captured signal is available. Maximum-likelihood (ML) approaches are proposed for T_{60} estimation by Ratnam et al. (2003) and Couvreur & Couvreur (2004). The main difficulty to estimate T_{60} is the requirement of silence regions between spoken words. Particularly in short utterances, this condition may not be fulfilled, leading to a considerable error in the T_{60} estimate.

In this chapter, instead of evaluating a specific algorithm we opt to assess the sensitivity of an ASR system to errors in the estimate of T_{60}. Experimental results showing the performance of spectral subtraction algorithm under such errors are presented in the next section.

4.4 Performance assessment in ASR systems

We have used the spectral subtraction approach as a preprocessing stage in an ASR system. The chosen task here consists of recognizing digit strings representing phone numbers in the Brazilian Portuguese language. Speech data recorded through telephone and sampled at 8 kHz are used as the original signal. In this experiment, we use 250 recordings, taken with several speakers. The reverberant speech signal is generated through a linear convolution between the original speech data and a given RIR. We have considered three different impulse responses, which are obtained by using the well-known image method to model acoustic room responses (Allen & Berkley, 1979). Room configurations used in the simulation experiments are given in Table 1.

In the spectral subtraction stage, the degraded signal is segmented into 25 ms frames, with an overlapping of 15 ms, and weighted by a Hamming window. The threshold T is fixed in 40 ms. We have considered magnitude subtraction ($\nu = 1$), since previous research works have obtained very good results with this configuration (Habets, 2004). From the modified magnitude spectrum and (original) phase signal, the enhanced signal is recovered by an overlap-and-add algorithm.

Parameter		Room #1	Room #2	Room #3
Dimensions (m)		7×7×3.5	6×8×3	9×8×3
Speaker position		(2.5, 3.8, 1.3)	(2.0, 3.0, 1.5)	(4.5, 5.0, 1.0)
Microphone position		(3.3, 3.0, 0.7)	(3.0, 3.5, 0.6)	(5.5, 6.5, 0.5)
Reflection coefficients	Walls	0.9	0.9	0.9
	Floor and ceiling	0.6	0.6	0.9
Resulting T_{60}(s)		0.68	0.73	0.83

Table 1. Parameters used for obtaining the room impulse responses.

Assessments have been carried out by using a speaker-independent HMM-based speech recognition system. Experiments are performed with word-based models, one for each of 11 digits (0 to 9 in Brazilian Portuguese plus the word "meia"[1]).

Acoustic features are extracted by a mel-cepstrum front-end developed for distributed speech recognition (DSR) (ETSI, 2002). This front-end includes a preprocessing stage of noise reduction using a Wiener filter (ETSI, 2002). Feature extraction is also carried out at each 25 ms frame, with an overlapping of 15 ms.

From each segment, 12 mel-frequency cepstral coefficients (MFCC) and the energy are computed, along with the first- and second-order derivatives. Thus, the final parameter vector is composed of 39 elements.

Recognition is performed by a Viterbi decoder with beam searching and word-end pruning (Young et al., 2002).

[1] In Brazilian Portuguese, it is common to speak "meia" for representing the number six. It is short for "meia dúzia" (half a dozen).

The results of the speech recognition task are presented in terms of the sentence error rate (SER), defined as

$$\mathrm{SER}(\%) = \frac{N_e}{N_s} 100 \tag{25}$$

where N_e is the number of sentences incorrectly recognized, and N_s is the total number of sentences in the test (250 in this evaluation). We have decided to use SER since for digit string recognition (phone numbers, in our case) an error in a single digit renders ineffective the result for the whole string. Note that SER is always greater than or equal to the word error rate (WER).

For the original speech data, SER is equal to 4%. For the reverberant data, obtained by the convolution of the original speech with the RIRs, SER increases to 64.4%, 77.6%, and 93.6% for Room #1, Room #2 and Room #3, respectively. This result reinforces the importance of coping with reverberation effects in ASR systems.

In order to evaluate spectral subtraction applied to reducing reverberation in ASR systems, we present the following simulation experiments:

i) Selection of oversubtraction factor α and spectral floor factor β. Here, we verify the best combination of parameters considering a speech recognition application.

ii) Sensitivity to errors in the estimate of T_{60}. Since an exact estimation of reverberation time could be difficult, we assess here the sensitivity of ASR to such errors.

iii) Effect of RIR variation. We evaluate the effect of speaker movement, which implies changes in the RIR.

iv) Effect of both reverberation and noise over ASR performance. In real enclosures, reverberation is usually associated with additive noise. We also assess this effect here.

4.4.1 Selection of oversubtraction factor and spectral floor

The first parameter we have evaluated is the oversubtraction factor α. Previous research works (Lebart & Boucher, 1998; Habets, 2004) assume α equal to 1. In contrast to them, we use the general formulation given by (11). We have evaluated α for different values of β and here we show the best results obtained using β = 0.2 and the particular value of T_{60} for each room (see Table 1). Fig. 4 shows the SER as a function of the oversubtraction factor between 0.4 and 1.3.

For Room #1 and Room #2, the best result is obtained with α = 0.7, which corresponds to an undersubtraction level. For Room #3, the best result is also obtained for α <1.

These particular results for reverberation reduction are in accordance with those obtained in studies about noise reduction discussed by Virag (1999) and Chen et al. (2006). Virag (1999) has verified that the oversubtraction parameter should be lower in ASR systems than for human listeners. Chen et al. (2006) have used a Wiener filter for denoising considering the factor α less than unity, leading to a satisfactory reduction in the distortion level over the resulting signal.

The influence of the spectral floor factor β, parameter that controls the masking level of musical noise, is shown in Fig. 5. For the three assessed room responses, the best result is obtained for β = 0.2, i.e., suggesting that it is important to maintain a certain level of masking noise. Note also that by not using any spectral flooring (β = 0) the SER increases. These results point out that ASR systems tolerate better residual noise than the inherent distortion provoked by the spectral subtraction processing, provided the noise level is not too high.

4.4.2 Sensitivity to errors in the estimation of the reverberation time

As discussed in Section 4.2, reverberation time must be estimated by the spectral subtraction algorithm. Since this estimate is subject to errors, it is important to evaluate the effect of such errors over ASR performance. The sensitivity to errors in the estimation of T_{60} has been assessed at the operating point $\alpha = 0.7$ and $\beta = 0.2$. We use the same set of RIRs as in Table 1. In the spectral subtraction algorithm, errors over T_{60} are introduced by varying the

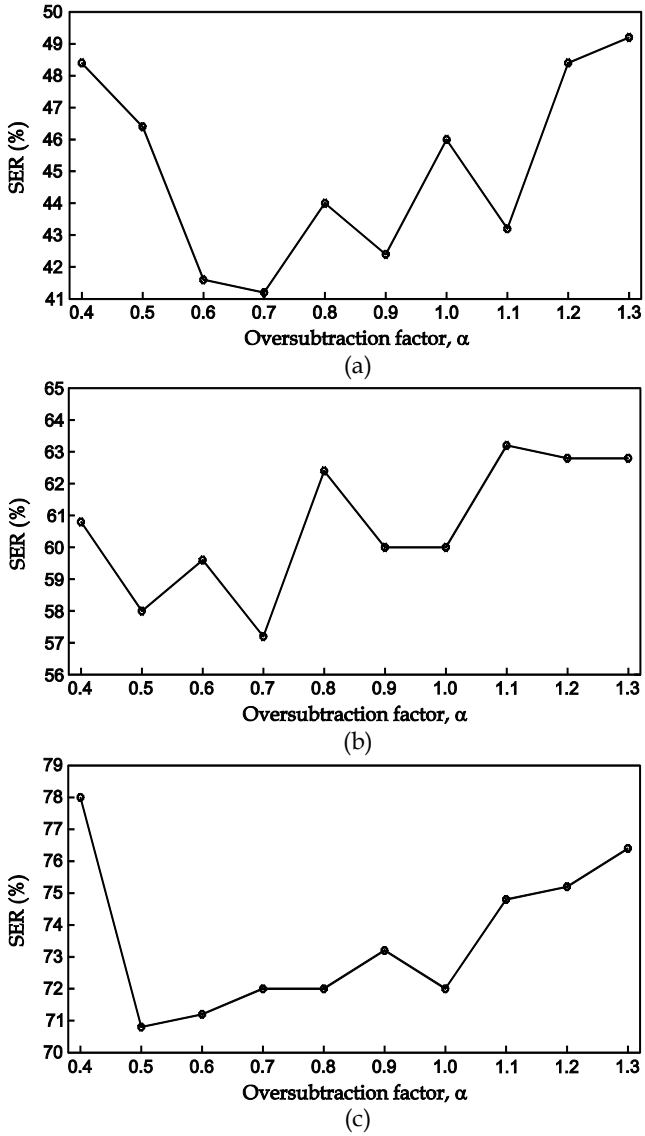


Fig. 4. Variation in SER as a function of α for $\beta = 0.2$ and the corresponding T_{60}.
(a) Room #1. (b) Room #2. (c) Room #3.

value from 0.3 to 1.3 s using steps of 0.2 s. Fig. 6 presents the SER in terms of such a variation. Ideally, the method should be less sensitive to errors in the estimation of T_{60}, since a blind estimate is very cost demanding in practice. Achieved results point out that even for an inaccurate estimate of T_{60}, the performance degradation is still tolerable.

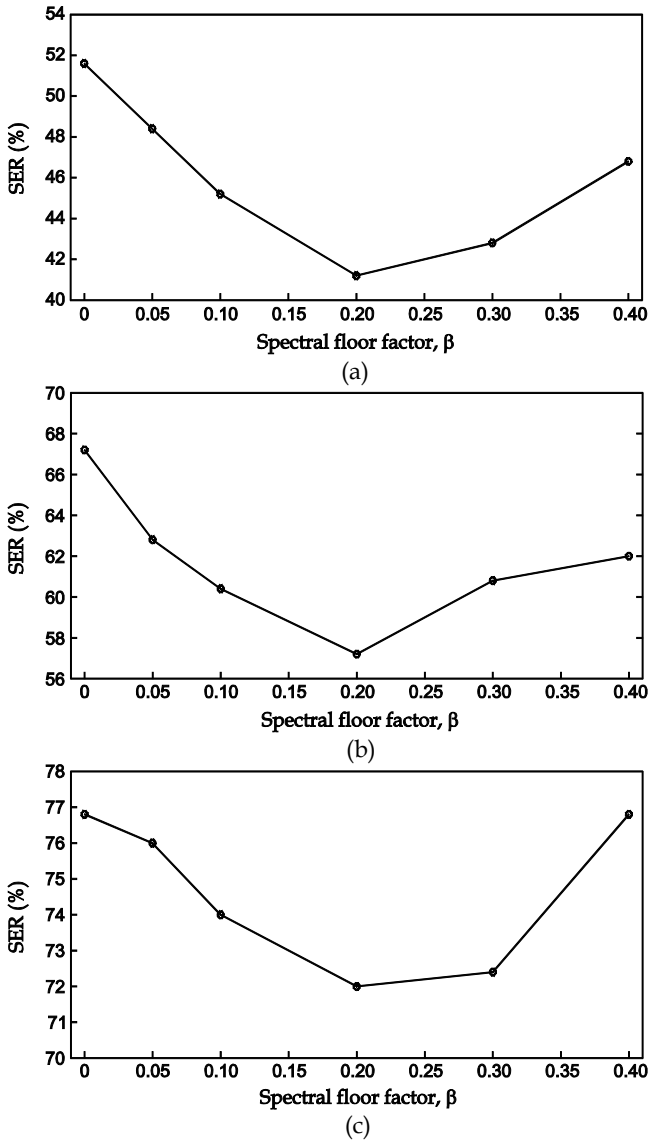


Fig. 5. Variation in SER as a function of β, keeping $\alpha = 0.7$ and the corresponding T_{60}. (a) Room #1. (b) Room #2. (c) Room #3.

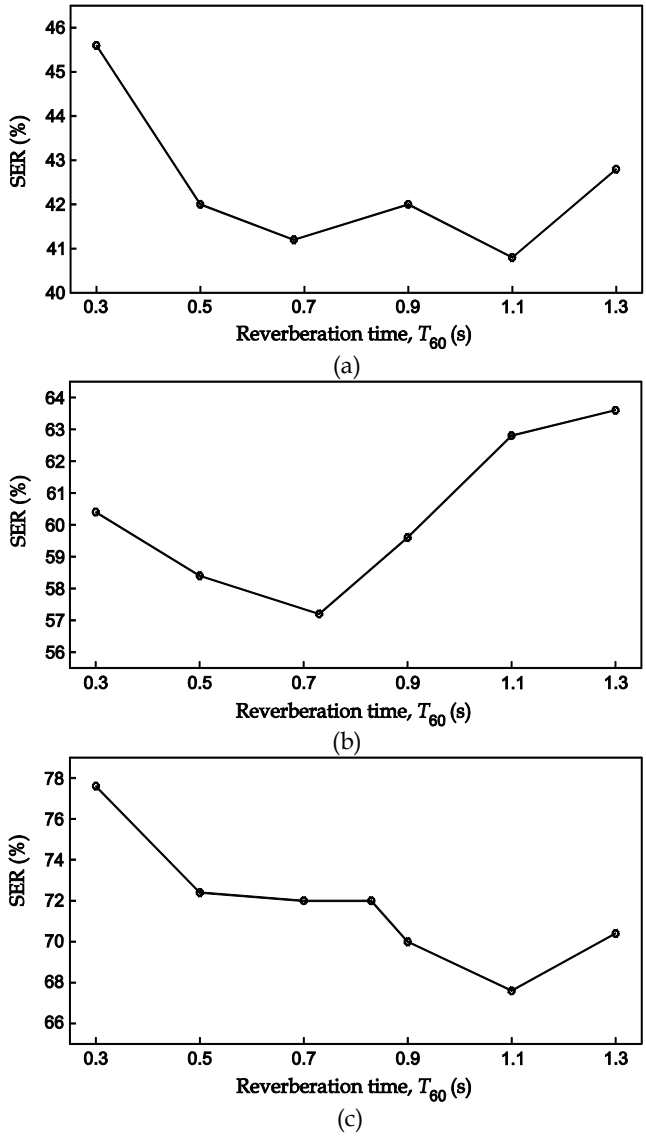


Fig. 6. Variation of SER as a function of T_{60} using $\alpha = 0.7$ and $\beta = 0.2$. (a) Room #1. (b) Room #2. (c) Room #3.

4.4.3 Effect of room impulse response variation caused by a moving speaker

The variation of the RIR in a particular enclosure is analyzed considering Room #1 (Table 1) and a moving speaker. The reference position of the speaker shown in Table 1 is shifted by 0.5 m (with a 0.25 m step) in both dimensions (length and width). Fig. 8 shows the ground plan of the enclosure, marking the positions of microphone and (moving) speaker. By using

this configuration, eight different RIRs are obtained. A set of reverberated audio signals is determined convolving each room response with the input signals from the test set. The spectral subtraction algorithm is configured with $\alpha = 0.7$, $\beta = 0.2$, and $T_{60} = 0.68$ s.

Results are presented in Table 2. Regarding the column "without processing", we observe that even small changes in the speaker position affect the ASR performance.

Test condition		SER (%)	
		Without processing	Spectral subtraction
Reference response		64.4	41.2
Speaker position shifted in the x axis by	−0.50 m	64.4	44.8
	−0.25 m	60.8	45.6
	+0.25 m	47.2	26.8
	+0.50 m	40.0	29.6
Speaker position shifted in the y axis by	−0.50 m	46.8	29.6
	−0.25 m	52.0	33.6
	+0.25 m	65.6	48.8
	+0.50 m	69.2	51.2

Table 2. SER as a function of room impulse response changes.

Although some performance reduction was expected, the effect of changes in the speaker position over the recognition rate is still considerable. In general, we verify that the larger the distance between speaker and microphone, the larger the error rate. These results confirm the need for making use of robust dereverberation techniques to cope with impulse response changes.

Spectral subtraction improves the recognition rate for all considered conditions. Error rates are reduced between 10 and 20 percentage points with respect to the standard front-end. Ideally, error rates should be less than or equal to the reference error rate (see Table 2). Although this is not verified, no instability is observed in the technique discussed here in contrast to some approaches presented in the open literature (Bees et al., 1991).

4.4.4 Combined effect of reverberation and noise

The combined effect of reverberation and additive noise is evaluated considering the addition of noise to the reverberant audio signals of Room #1 (Table 1). Samples of noise are obtained from the set available in Hansen & Arslan, (1995). We have considered two types of noise: the first one is named large city noise (LCI) and the other is white Gaussian noise (WGN), with three signal-to-noise ratio (SNR) levels: 5, 10, and 15 dB.

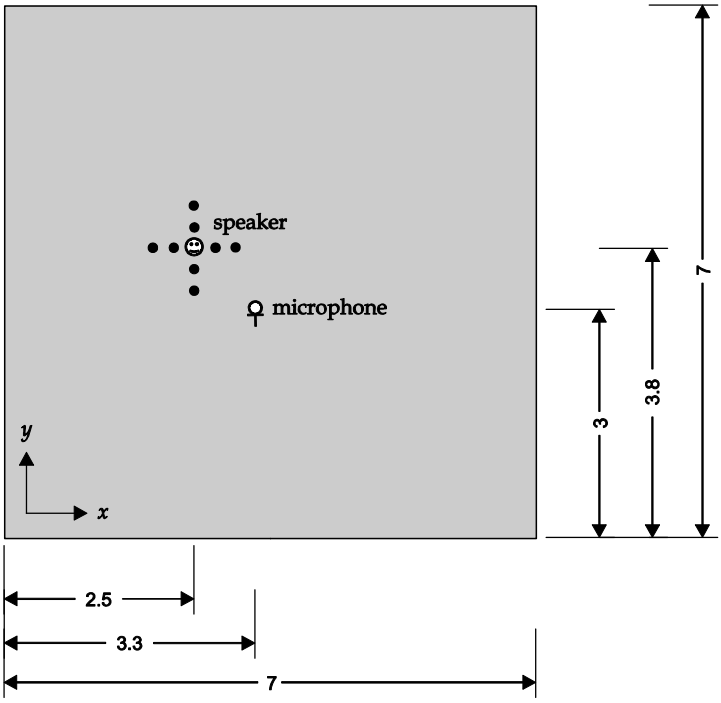


Fig. 7. Ground plan of the room showing speaker and microphone positions (dimensions in m). Speaker position is shifted with a 0.25 m step.

Test condition	SER (%)	
	Without processing	Spectral subtraction
Only reverberation	64.4	41.2
Reverberation + large city noise at SNR 15 dB	75.6	59.6
Reverberation + large city noise at SNR 10 dB	85.6	75.2
Reverberation + large city noise at SNR 5 dB	97.6	92.4
Reverberation + white Gaussian noise at SNR 15 dB	84.0	66.8
Reverberation + white Gaussian noise at SNR 10 dB	91.6	80.0
Reverberation + white Gaussian noise at SNR 5 dB	98.4	95.6

Table 3. Combined effects of reverberation and noise.

Table 3 shows the SER values. Column "without processing" presents the deleterious effect of reverberation and noise over the speech recognition performance. Error rate increases significantly as SNR decreases.
With spectral subtraction, the error is reduced for all considered situations, although it is still high for the worst noise settings. Apart from that, we do not observe any kind of instability, seen in some other approaches.

5. Concluding remarks

This chapter has characterized effects of reverberation and noise over ASR system performance. We have shown the importance of coping with such degradations in order to improve ASR performance in real applications. A brief overview of current dereverberation and denoising approaches has been addressed, classifying methods according to the point of operation in the speech recognition chain. The use of spectral subtraction applied to dereverberation and denoising in ASR systems has been discussed, giving rise to a consistent formulation to treat this impacting problem. We assessed the used approach considering the sentence error rate over a digit string recognition task, showing that the recognition rate can be significantly improved by using spectral subtraction. The impact on the choice of algorithm parameters has been assessed under different environmental conditions for performance. Finally, it is important to mention that reverberation and noise problems in ASR systems continue to be a challenging subject for the signal processing community.

6. References

ETSI (2002). Speech processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms, European Telecommunications Standards Institute (ETSI) Std. ES 202 050 V.1.1.1, Oct. 2002.

Allen, J. B. & Berkley, D. A. (1979). Image method for efficiently simulating small-room acoustics. *Journal of the Acoustical Society of America*, Vol. 65, No. 4, Apr. 1979, pp. 943-950.

Bees, D.; Blostein, M. & Kabal, P. (1991). Reverberant speech enhancement using cepstral processing. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'91)*, Vol. 2, pp. 977–980, Toronto, Canada, Apr. 1991.

Berouti, M.; Schwartz, R. & Makhoul, J. Enhancement of speech corrupted by acoustic noise. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'79)*, Vol. 4, pp. 208-211, Washington, USA, Apr. 1979.

Boll, S. F. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 27, No. 2, Apr. 1979, pp. 113-120.

Chen, J.; Benesty, J.; Huang, Y. & Doclo, S. (2006). New insights into the noise reduction Wiener filter. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, No. 4, July 2006, pp. 1218–1234.

Chesnokov, A. & SooHoo, L. (1998). Influence of early to late energy ratios on subjective estimates of small room acoustics. *Proceedings of the 105th AES Convention*, pp. 1–18, San Francisco, USA, Sept. 1998.

Couvreur, L. & Couvreur, C. (2004). Blind model selection for automatic speech recognition in reverberant environments. *Journal of VLSI Signal Processing*, Vol. 36, No. 2-3, Feb./Mar. 2004, pp. 189-203.

de la Torre, A.; Segura, J. C.; Benitez, C.; Ramirez, J.; Garcia, L. & Rubio, A. J. (2007). Speech recognition under noise conditions: Compensation methods. In: *Speech Recognition and Understanding*, Grimm, M. & Kroschel, K. (Eds.), pp. 439-460, I-Tech, ISBN 978-3-902-61308-0. Vienna, Austria.

Droppo, J. & Acero, A. (2008). Environmental robustness. In: *Springer Handbook of Speech Processing*, Benesty, J.; Sondhi, M. M. & Huang, Y. (Eds.), pp. 653-679, Springer, ISBN 978-3-540-49125-5, Berlin, Germany.

Everest, F. A. (2001). *The Master Handbook of Acoustics.* 4 ed., McGraw-Hill, ISBN 978-0-071-36097-5, New York, USA.

Gales, M. J. F. & Young, S. J. (1995). A fast and flexible implementation of parallel model combination. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'95)*, Vol. 1, pp. 133–136. Detroit, USA, May 1995.

Gillespie, B. W.; Malvar, H. S. & Florêncio, D. A. F. (2001). Speech dereverberation via maximum-kurtosis subband adaptive filtering. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, Vol. 6, pp. 3701–3704. Salt Lake City, USA, May 2001.

Habets, E. A. P. (2004). Single-channel speech dereverberation based on spectral subtraction. *Proceedings of the 15th Annual Workshop on Circuits, Systems and Signal Processing (ProRISC'04)*, pp. 250–254, Veldhoven, Netherlands, Nov. 2004.

Hansen, J. H. L. & Arslan, L. (1995). Robust feature-estimation and objective quality assessment for noisy speech recognition using the credit-card corpus. *IEEE Transactions on Speech and Audio Processing*, Vol. 3, No. 3, May 1995, pp. 169–184.

Hermansky, H. & Morgan, N. (1994). RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, Vol. 2, No. 4, Oct. 1994, pp. 578–589.

Huang, Y.; Benesty, J. & Chen, J. (2008). Dereverberation. In: *Springer Handbook of Speech Processing*, Benesty, J.; Sondhi, M. M. & Huang, Y. (Eds.), pp. 929-943, Springer, ISBN 978-3-540-49125-5, Berlin, Germany.

Iskra, D.; Grosskopf, B.; Marasek, K.; van den Heuvel, H.; Diehl, F. & Kiessling, A. (2002). SPEECON - Speech databases for consumer devices: Database specification and validation. *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'2002)*, pp. 329-333, Las Palmas, Spain, May 2002.

Kingsbury, B. E. D. (1998). *Perceptually Inspired Signal-processing Strategies for Robust Speech Recognition in Reverberant Environments*. PhD Thesis, University of California, Berkeley.

Kondoz, A. M. (2004). *Digital Speech: Coding for Low Bit Rate Communication Systems*. 2 ed, Wiley, ISBN 978-0-470-87008-2, Chichester, UK.

Lebart, K. & Boucher, J. M. (1998). A new method based on spectral subtraction for the suppression of late reverberation from speech signals. *Proceedings of the 105th AES Convention*, pp. 1–13, San Francisco, USA, Sept. 1998.

Moreno, A.; Lindberg, B.; Draxler, C.; Richard, G.; Choukri, K.; Euler, S. & Allen, J. (2000). SPEECHDAT-CAR. A large speech database for automotive environments. *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'2000)*, Vol. 2, pp. 895–900, Athens, Greece, May/June 2000.

Nakatani, T.; Kinoshita, K. & Miyoshi, M. (2007). Harmonicity-based blind dereverberation for single-channel speech signals. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 15, No. 1, Jan. 2007, pp. 80–95.

Omologo, M.; Svaizer, P. & Matassoni, M. (1998). Environmental conditions and acoustic transduction in hands-free speech recognition. *Speech Communication*, Vol. 25, No. 1-3, Aug. 1998, pp. 75–95.

Pacheco, F. S. & Seara, R. (2005). A single-microphone approach for speech signal dereverberation. *Proceedings of the European Signal Processing Conference (EUSIPCO'05)*, pp. 1–4, Antalya, Turkey, Sept. 2005.

Palomäki, K. J.; Brown, G. J. & Barker, J. P. (2004). Techniques for handling convolutional distortion with 'missing data' automatic speech recognition. *Speech Communication*, Vol. 43, No. 1-2, June 2004, pp. 123–142.

Rabiner, L. & Juang, B.-H. (2008). Historical perspectives of the field of ASR/NLU. In: *Springer Handbook of Speech Processing,* Benesty, J.; Sondhi, M. M. & Huang, Y. (Eds.), pp. 521-537, Springer, ISBN 978-3-540-49125-5, Berlin, Germany.

Radlović, B. D. & Kennedy, R. A. (2000). Nonminimum-phase equalization and its subjective importance in room acoustics. *IEEE Transactions on Speech and Audio Processing*, Vol. 8, No. 6, Nov. 2000, pp. 728–737.

Raj, B. & Stern, R. M. (2005). Missing-feature approaches in speech recognition. *IEEE Signal Processing Magazine*, Vol. 22, No. 5, Sept. 2005, pp. 101–116.

Ratnam, R.; Jones, D. L.; Wheeler, B. C.; O'Brien Jr., W. D.; Lansing, C. R. & Feng, A. S. (2003). Blind estimation of reverberation time. *Journal of the Acoustical Society of America*, Vol. 114, No. 5, Nov. 2003, pp. 2877–2892.

Seltzer, M. L.; Raj, B. & Stern, R. M. (2004). Likelihood-maximizing beamforming for robust hands-free speech recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 12, No. 5, Sept. 2004, pp. 489–498.

Virag, N. (1999). Single channel speech enhancement based on masking properties of the human auditory system. *IEEE Transactions on Speech and Audio Processing*, Vol. 7, No. 2, Mar. 1999, pp. 126-137.

Ward, D. B.; Kennedy, R. A. & Williamson, R. C. (2001). Constant directivity beamforming. In: *Microphone Arrays: Signal Processing Techniques and Applications,* Brandstein, M. & Ward, D. (Eds.), pp. 3-17, Springer, ISBN 978-3-540-41953-2, Berlin, Germany.

Wu, M. & Wang, D. (2006). A two-stage algorithm for one-microphone reverberant speech enhancement. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, No. 3, May 2006, pp. 774–784.

Young, S.; Kershaw, D.; Odell, J.; Ollason, D.; Valtchev, V. & Woodland, P. (2002). *The HTK Book (for HTK Version 3.2)*. Cambridge University, Cambridge, UK.

6

Feature Transformation Based on Generalization of Linear Discriminant Analysis

Makoto Sakai[1,2], Norihide Kitaoka[2] and Seiichi Nakagawa[3]
[1]*DENSO CORP.*,
[2]*Nagoya University*,
[3]*Toyohashi University of Technology*
Japan

1. Introduction

Hidden Markov models (HMMs) have been widely used to model speech signals for speech recognition. However, they cannot precisely model the time dependency of feature parameters. In order to overcome this limitation, several researchers have proposed extensions, such as segmental unit input HMM (Nakagawa & Yamamoto, 1996). Segmental unit input HMM has been widely used for its effectiveness and tractability. In segmental unit input HMM, the immediate use of several successive frames as an input vector inevitably increases the number of dimensions. The concatenated vectors may have strong correlations among dimensions, and may include nonessential information. In addition, high-dimensional data require a heavy computational load. Therefore, to reduce dimensionality, a feature transformation method is often applied. Linear discriminant analysis (LDA) is widely used to reduce dimensionality and a powerful tool to preserve discriminative information. LDA assumes each class has the same class covariance. However, this assumption does not necessarily hold for a real data set. In order to remove this limitation, several methods have been proposed. Heteroscedastic linear discriminant analysis (HLDA) could deal with unequal covariances because the maximum likelihood estimation was used to estimate parameters for different Gaussians with unequal covariances. Heteroscedastic discriminant analysis (HDA) was proposed as another objective function, which employed individual weighted contributions of the classes. The effectiveness of these methods for some data sets has been experimentally demonstrated. However, it is difficult to find one particular criterion suitable for any kind of data set. In this chapter we show that these three methods have a strong mutual relationship, and provide a new interpretation for them. Then, we present a new framework that we call power linear discriminant analysis (PLDA) (Sakai et al., 2007), which can describe various criteria including the discriminant analyses with one control parameter. Because PLDA can describe various criteria for dimensionality reduction, it can flexibly adapt to various environments such as a noisy environment. Thus, PLDA can provide robustness to a speech recognizer in realistic environments. Moreover, the presented technique can combine a discriminative training, such as maximum mutual information (MMI) and minimum phone error (MPE). Experimental results show the effectiveness of the presented technique.

2. Notations

This chapter uses the following notation: capital bold letters refer to matrices, e.g., **A**, bold letters refer to vectors, e.g., **b**, and scalars are not bold, e.g., *c*. Where submatrices are used they are indicated, for example, by $\mathbf{A}_{[p]}$, this is an $n \times p$ matrix. $\mathbf{A}^T$ is the transpose of the matrix, $|\mathbf{A}|$ is the determinant of the matrix, and tr(**A**) is the trace of the matrix.
We let the function f of a symmetric positive definite matrix $\mathbf{A}$ equal $\mathbf{U}\,diag\left(f(\lambda_1),\cdots,f(\lambda_n)\right)\mathbf{U}^T = \mathbf{U}\left(f(\mathbf{\Lambda})\right)\mathbf{U}^T$, where $\mathbf{A} = \mathbf{U}\Lambda\,\mathbf{U}^T$, $\mathbf{U}$ denotes the matrix of n eigenvectors, and $\mathbf{\Lambda}$ denotes the diagonal matrix of eigenvalues, λ_i's. We may define the function f as some power or the logarithm of **A**.

3. Segmental Unit Input HMM

For an input symbol sequence $\mathbf{o} = (\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T)$ and a state sequence $\mathbf{q} = (q_1, q_2, \cdots, q_T)$, the output probability of segmental unit input HMM is given by the following equations (Nakagawa & Yamamoto, 1996):

$$P(\mathbf{o}_1, \cdots, \mathbf{o}_T) = \sum_{\mathbf{q}} \prod_i P(\mathbf{o}_i \mid \mathbf{o}_1, \cdots, \mathbf{o}_{i-1}, q_1, \cdots, q_i) \times P(q_i \mid q_1, \cdots, q_{i-1}) \tag{1}$$

$$\approx \sum_{\mathbf{q}} \prod_i P(\mathbf{o}_i \mid \mathbf{o}_{i-(d-1)}, \cdots, \mathbf{o}_{i-1}, q_i) P(q_i \mid q_{i-1}) \tag{2}$$

$$\approx \sum_{\mathbf{q}} \prod_i P(\mathbf{o}_{i-(d-1)}, \cdots, \mathbf{o}_i \mid q_i) P(q_i \mid q_{i-1}), \tag{3}$$

where T denotes the length of input sequence and d denotes the number of successive frames used in probability calculation at a current frame. The immediate use of several successive frames as an input vector inevitably increases the number of parameters. When the number of dimensions increases, several problems generally occur: heavier computational load and larger memory are required, and the accuracy of parameter estimation degrades. Therefore, to reduce dimensionality, feature transformation methods, e.g., principal component analysis (PCA), LDA, HLDA or HDA, are often used (Nakagawa & Yamamoto, 1996; Haeb-Umbach & Ney, 1992; Kumar & Andreou, 1998; Saon et al., 2000).
Here, we briefly review LDA, HLDA and HDA, and then investigate the effectiveness of these methods for some artificial data sets.

3.1 Linear discriminant analysis

Given n-dimensional feature vectors $\mathbf{x}_j \in \Re^n$ $(j = 1,2,\cdots,N)$, e.g., $\mathbf{x}_j = \left[\mathbf{o}_{j-(d-1)}^T, \cdots, \mathbf{o}_j^T\right]^T$, let us find a transformation matrix $\mathbf{B}_{[p]} \in \Re^{n \times p}$ that projects these feature vectors to p-dimensional feature vectors $\mathbf{z}_j \in \Re^p$ $(j = 1,2,\cdots,N)$ $(p < n)$, where $\mathbf{z}_j = \mathbf{B}_{[p]}^T \mathbf{x}_j$, and N denotes the number of all features.
Within-class and between-class covariance matrices are defined as follows (Fukunaga, 1990):

$$\Sigma_w = \frac{1}{N}\sum_{k=1}^{c}\sum_{\mathrm{x}_j \in D_k}\left(\mathrm{x}_j - \mu_k\right)\left(\mathrm{x}_j - \mu_k\right)^T$$

$$= \sum_{k=1}^{c} P_k \Sigma_k, \tag{4}$$

$$\Sigma_b = \sum_{k=1}^{c} P_k \left(\mu_k - \mu\right)\left(\mu_k - \mu\right)^T, \tag{5}$$

where c denotes the number of classes, D_k denotes the subset of feature vectors labeled as class k, μ is the mean vector of all features, μ_k is the mean vector of the class k, Σ_k is the covariance matrix of the class k, and P_k is the class weight, respectively.

There are several ways to formulate objective functions for multi-class data (Fukunaga, 1990). Typical objective functions are the following:

$$J_{LDA}\left(\mathrm{B}_{[p]}\right) = \frac{\left|\mathrm{B}_{[p]}^T \Sigma_b \mathrm{B}_{[p]}\right|}{\left|\mathrm{B}_{[p]}^T \Sigma_w \mathrm{B}_{[p]}\right|}, \tag{6}$$

$$J_{LDA}\left(\mathrm{B}_{[p]}\right) = \frac{\left|\mathrm{B}_{[p]}^T \Sigma_t \mathrm{B}_{[p]}\right|}{\left|\mathrm{B}_{[p]}^T \Sigma_w \mathrm{B}_{[p]}\right|}, \tag{7}$$

where Σ_t denotes the covariance matrix of all features, namely a total covariance, which equals $\Sigma_b + \Sigma_w$.

LDA finds a transformation matrix $\mathbf{B}_{[p]}$ that maximizes Eqs. (6) or (7). The optimum transformations of (6) and (7) result in the same transformation.

3.2 Heteroscedastic extensions

LDA is not the optimal transformation when the class distributions are heteroscedastic. Campbell has shown that LDA is related to the maximum likelihood estimation of parameters for a Gaussian model with an identical class covariance (Campbell, 1984). However, this condition is not necessarily satisfied for a real data set.

In order to overcome this limitation, several extensions have been proposed. This chapter focuses on two heteroscedastic extensions called heteroscedastic linear discriminant analysis (HLDA) (Kumar & Andreou, 1998) and heteroscedastic discriminant analysis (HDA) (Saon et al., 2000).

3.2.1 Heteroscedastic linear discriminant analysis

In HLDA, the full-rank linear transformation matrix $\mathbf{B} \in \Re^{n \times n}$ is constrained as follows: the first p columns of $\mathbf{B}$ span the p-dimensional subspace in which the class means and variances are different and the remaining n-p columns of $\mathbf{B}$ span the (n-p)-dimensional subspace in which the class means and variances are identical. Let the parameters that describe the class means and covariances of $\mathbf{B}^T\mathbf{x}$ be $\hat{\mu}_k$ and $\hat{\Sigma}_k$, respectively:

$$\hat{\mu}_k = \begin{bmatrix} \mathrm{B}_{[p]}^T \mu_k \\ \mathrm{B}_{[n-p]}^T \mu \end{bmatrix}, \tag{8}$$

$$\hat{\Sigma}_k = \begin{bmatrix} \mathrm{B}_{[p]}^T \Sigma_k \mathrm{B}_{[p]} & 0 \\ 0 & \mathrm{B}_{[n-p]}^T \Sigma_t \mathrm{B}_{[n-p]} \end{bmatrix}, \tag{9}$$

where $\mathbf{B} = \left[\mathbf{B}_{[p]} | \mathbf{B}_{[n-p]}\right]$ and $\mathbf{B}_{[n-p]} \in \Re^{n \times (n-p)}$.

Kumar et al. incorporated the maximum likelihood estimation of parameters for differently distributed Gaussians. An HLDA objective function is derived as follows (Kumar & Andreou, 1998):

$$J_{HLDA}\left(\mathrm{B}\right) = \frac{\left|\mathrm{B}\right|^{2N}}{\left|\mathrm{B}_{[n-p]}^T \Sigma_t \mathrm{B}_{[n-p]}\right|^N} \frac{1}{\prod_{k=1}^{c} \left|\mathrm{B}_{[p]}^T \Sigma_k \mathrm{B}_{[p]}\right|^{N_k}}. \tag{10}$$

N_k denotes the number of features of class *k*. The solution to maximize Eq. (10) is not analytically obtained. Therefore, its maximization is performed using a numerical optimization technique.

3.2.2 Heteroscedastic discriminant analysis

HDA uses the following objective function, which incorporates individual weighted contributions of the class variances (Saon et al., 2000):

$$J_{HDA}\left(\mathrm{B}_{[p]}\right) = \prod_{k=1}^{c} \left(\frac{\left|\mathrm{B}_{[p]}^T \Sigma_b \mathrm{B}_{[p]}\right|}{\left|\mathrm{B}_{[p]}^T \Sigma_k \mathrm{B}_{[p]}\right|} \right)^{N_k} \tag{11}$$

$$= \frac{\left|\mathrm{B}_{[p]}^T \Sigma_b \mathrm{B}_{[p]}\right|^N}{\prod_{k=1}^{c} \left|\mathrm{B}_{[p]}^T \Sigma_k \mathrm{B}_{[p]}\right|^{N_k}}. \tag{12}$$

In contrast to HLDA, this function is not considered (*n*-*p*) dimensions. Only a transformation matrix $\mathbf{B}_{[p]}$ is estimated. There is no closed-form solution to obtain transformation matrix $\mathbf{B}_{[p]}$ similar to HLDA.

3.3 Dependency on data set

In Fig. 1, two-dimensional, two- or three-class data features are projected onto one-dimensional subspaces by LDA and HDA. Here, HLDA projections were omitted because they were close to HDA projections. Fig. 1 (a) shows that HDA has higher separability than LDA for the data set used in (Saon et al., 2000). On the other hand, as shown in Fig. 1(b), LDA has higher separability than HDA for another data set. Fig. 1 (c) shows the case with another data set where both LDA and HDA have low separabilities. Thus, LDA and HDA

do not always classify the given data set appropriately. All results show that the separabilities of LDA and HDA depend significantly on data sets.

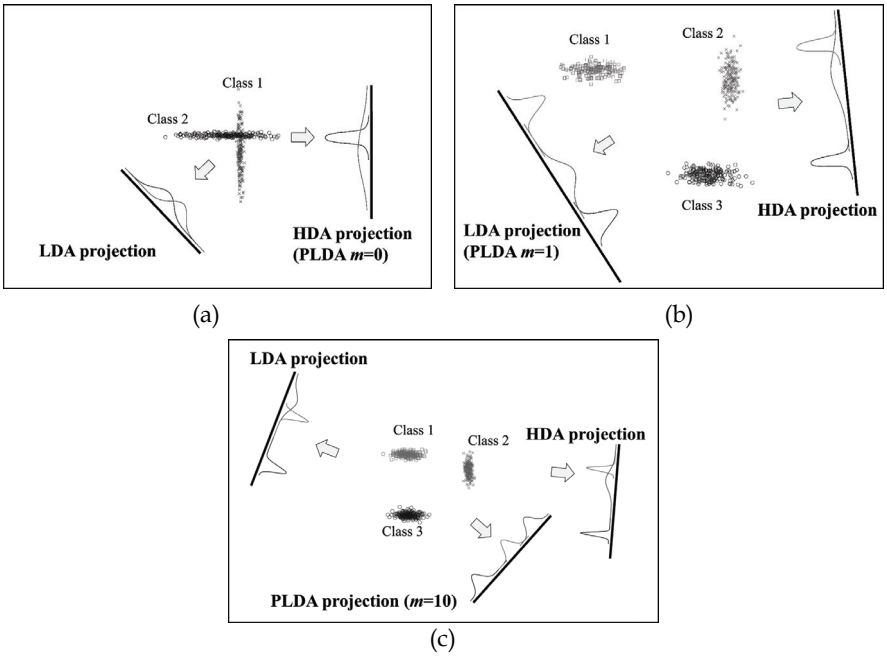


Fig. 1. Examples of dimensionality reduction by LDA, HDA and PLDA.

4. Generalization of discriminant analysis

As shown above, it is difficult to separate appropriately every data set with one particular criterion such as LDA, HLDA, or HDA. Here, we concentrate on providing a framework which integrates various criteria.

4.1 Relationship between HLDA and HDA

By using Eqs. (8) and (9), let us rearrange $\mathbf{B}^T\Sigma_t\mathbf{B}$ as follows:

$$\mathrm{B}^T\Sigma_t\mathrm{B} = \mathrm{B}^T\Sigma_b\mathrm{B} + \mathrm{B}^T\Sigma_w\mathrm{B} \tag{13}$$

$$= \sum_k P_k\left(\hat{\mu}_k - \hat{\mu}\right)\left(\hat{\mu}_k - \hat{\mu}\right)^T + \sum_k P_k\hat{\Sigma}_k \tag{14}$$

$$= \begin{bmatrix} \mathrm{B}_{[p]}^T\Sigma_t\mathrm{B}_{[p]} & 0 \\ 0 & \mathrm{B}_{[n-p]}^T\Sigma_t\mathrm{B}_{[n-p]} \end{bmatrix}, \tag{15}$$

where $\hat{\mu} = \mathbf{B}^T\mu$.

The determinant of this is

$$\left|\mathrm{B}^T\Sigma_t\mathrm{B}\right| = \left|\mathrm{B}_{[p]}^T\Sigma_t\mathrm{B}_{[p]}\right|\left|\mathrm{B}_{[n-p]}^T\Sigma_t\mathrm{B}_{[n-p]}\right|, \tag{16}$$

Inserting this in (10) and removing a constant term yields

$$J_{HLDA}\left(\mathrm{B}_{[p]}\right) \propto \frac{\left|\mathrm{B}_{[p]}^T\Sigma_t\mathrm{B}_{[p]}\right|^N}{\prod_{k=1}^{c}\left|\mathrm{B}_{[p]}^T\Sigma_k\mathrm{B}_{[p]}\right|^{N_k}}. \tag{17}$$

From (12) and (17), the difference between HLDA and HDA lies in their numerators, i.e., the total covariance matrix versus the between-class covariance matrix. This difference is the same as the difference between the two LDAs shown in (6) and (7). Thus, (12) and (17) can be viewed as the same formulation.

4.2 Relationship between LDA and HDA

The LDA and HDA objective functions can be rewritten as

$$J_{LDA}\left(\mathrm{B}_{[p]}\right) = \frac{\left|\mathrm{B}_{[p]}^T\Sigma_b\mathrm{B}_{[p]}\right|}{\left|\mathrm{B}_{[p]}^T\Sigma_w\mathrm{B}_{[p]}\right|} = \frac{\left|\tilde{\Sigma}_b\right|}{\left|\sum_{k=1}^{c}P_k\tilde{\Sigma}_k\right|}, \tag{18}$$

$$J_{HDA}\left(\mathrm{B}_{[p]}\right) = \frac{\left|\mathrm{B}_{[p]}^T\Sigma_b\mathrm{B}_{[p]}\right|^N}{\prod_{k=1}^{c}\left|\mathrm{B}_{[p]}^T\Sigma_k\mathrm{B}_{[p]}\right|^{N_k}} \propto \frac{\left|\tilde{\Sigma}_b\right|}{\left|\prod_{k=1}^{c}\tilde{\Sigma}_k^{P_k}\right|}, \tag{19}$$

where $\tilde{\Sigma}_b = \mathbf{B}_{[p]}^T\Sigma_b\mathbf{B}_{[p]}$ and $\tilde{\Sigma}_k = \mathbf{B}_{[p]}^T\Sigma_k\mathbf{B}_{[p]}$ are between-class and class k covariance matrices in the projected p-dimensional space, respectively.

Both numerators denote determinants of the between-class covariance matrix. In Eq. (18), the denominator can be viewed as a determinant of *the weighted arithmetic mean* of the class covariance matrices. Similarly, in Eq. (19), the denominator can be viewed as a determinant of *the weighted geometric mean* of the class covariance matrices. Thus, the difference between LDA and HDA is the definitions of the mean of the class covariance matrices. Moreover, to replace their numerators with the determinants of the total covariance matrices, the difference between LDA and HLDA is the same as the difference between LDA and HDA.

4.3 Power linear discriminant analysis

As described above, Eqs. (18) and (19) give us a new integrated interpretation of LDA and HDA. As an extension of this interpretation, their denominators can be replaced by a determinant of *the weighted harmonic mean*, or a determinant of *the root mean square*.

In the econometric literature, a more general definition of a mean is often used, called *the weighted mean of order m* (Magnus & Neudecker, 1999). We have extended this notion to a determinant of a matrix mean and have proposed a new objective function as follows (Sakai et al., 2007):

$$J_{PLDA}\left(\mathrm{B}_{[p]}, m\right) = \frac{\left|\tilde{\Sigma}_n\right|}{\left|\left(\sum_{k=1}^{c}P_k\tilde{\Sigma}_k^m\right)^{1/m}\right|}, \tag{20}$$

where $\tilde{\Sigma}_n \in \{\tilde{\Sigma}_b, \tilde{\Sigma}_t\}, \tilde{\Sigma}_t = \mathbf{B}_{[p]}^T \Sigma_t \mathbf{B}_{[p]}$, and m is a control parameter. By varying the control parameter m, the proposed objective function can represent various criteria. Some typical objective functions are enumerated below.

- m=2 (root mean square)

$$J_{PLDA}\left(\mathrm{B}_{[p]}, 2\right) = \frac{\left|\tilde{\Sigma}_n\right|}{\left|\left(\sum_{k=1}^{c} P_k \tilde{\Sigma}_k^2\right)^{1/2}\right|}. \tag{21}$$

- m=1 (arithmetic mean)

$$J_{PLDA}\left(\mathrm{B}_{[p]}, 1\right) = \frac{\left|\tilde{\Sigma}_n\right|}{\left|\sum_{k=1}^{c} P_k \tilde{\Sigma}_k\right|} = J_{LDA}\left(\mathrm{B}_{[p]}\right). \tag{22}$$

- $m \to 0$ (geometric mean)

$$J_{PLDA}\left(\mathrm{B}_{[p]}, 0\right) = \frac{\left|\tilde{\Sigma}_n\right|}{\prod_{k=1}^{c} \left|\tilde{\Sigma}_k\right|^{P_k}} \propto J_{HDA}\left(\mathrm{B}_{[p]}\right). \tag{23}$$

- m=-1 (harmonic mean)

$$J_{PLDA}\left(\mathrm{B}_{[p]}, -1\right) = \frac{\left|\tilde{\Sigma}_n\right|}{\left|\left(\sum_{k=1}^{c} P_k \tilde{\Sigma}_k^{-1}\right)^{-1}\right|}. \tag{24}$$

The following equations are also obtained under a particular condition.

- $m \to \infty$

$$J_{PLDA}\left(\mathrm{B}_{[p]}, \infty\right) = \frac{\left|\tilde{\Sigma}_n\right|}{\max_k \left|\tilde{\Sigma}_k\right|}. \tag{25}$$

- $m \to -\infty$

$$J_{PLDA}\left(\mathrm{B}_{[p]}, -\infty\right) = \frac{\left|\tilde{\Sigma}_n\right|}{\min_k \left|\tilde{\Sigma}_k\right|}. \tag{26}$$

Intuitively, as m becomes larger, the classes with larger variances become dominant in the denominator of Eq. (20). Conversely, as m becomes smaller, the classes with smaller variances become dominant.

We call this new discriminant analysis formulation *Power Linear Discriminant Analysis* (PLDA). Fig. 1 (c) shows that PLDA with m=10 can have a higher separability for a data set

with which LDA and HDA have lower separability. To maximize the PLDA objective function with respect to **B**, we can use numerical optimization techniques such as the Nelder-Mead method or the SANN method. These methods need no derivatives of the objective function. However, it is known that these methods converge slowly. In some special cases below, using a matrix differential calculus, the derivatives of the objective function are obtained. Hence, we can use some fast convergence methods, such as the quasi-Newton method and conjugate gradient method.

4.3.1 Order *m* constrained to be an integer

Assuming that a control parameter m is constrained to be an integer, the derivatives of the PLDA objective function are formulated as follows:

$$\frac{\partial}{\partial \mathrm{B}_{[p]}} \log J_{PLDA}\left(\mathrm{B}_{[p]}, m\right) = 2\Sigma_n \mathrm{B}_{[p]} \tilde{\Sigma}_n^{-1} - 2\mathrm{D}_m, \tag{27}$$

where

$$\mathbf{D}_m = \begin{cases} \dfrac{1}{m}\displaystyle\sum_{k=1}^{c} P_k \Sigma_k \mathbf{B}_{[p]} \sum_{j=1}^{m} \mathbf{X}_{m,j,k}, & \text{if m} > 0 \\ \displaystyle\sum_{k=1}^{c} P_k \Sigma_k \mathbf{B}_{[p]} \tilde{\Sigma}_k^{-1}, & \text{if m} = 0 \\ -\dfrac{1}{m}\displaystyle\sum_{k=1}^{c} P_k \Sigma_k \mathbf{B}_{[p]} \sum_{j=1}^{|m|} \mathbf{Y}_{m,j,k}, & \text{otherwise} \end{cases}$$

$$\mathbf{X}_{m,j,k} = \tilde{\Sigma}_k^{m-j} \left(\sum_{l=1}^{c} P_l \tilde{\Sigma}_l^{m} \right)^{-1} \tilde{\Sigma}_k^{j-1},$$

and

$$\mathbf{Y}_{m,j,k} = \tilde{\Sigma}_k^{m+j-1} \left(\sum_{l=1}^{c} P_l \tilde{\Sigma}_l^{m} \right)^{-1} \tilde{\Sigma}_k^{-j}.$$

This equation can be used for acoustic models with full covariance.

4.3.2 $\tilde{\Sigma}_k$ constrained to be diagonal

Because of computational simplicity, the covariance matrix of class k is often assumed to be diagonal (Kumar & Andreou, 1998; Saon et al., 2000). Since a diagonal matrix multiplication is commutative, the derivatives of the PLDA objective function are simplified as follows:

$$J_{PLDA}\left(\mathrm{B}_{[p]}, m\right) = \frac{\left|\tilde{\Sigma}_n\right|}{\left|\left(\sum_{k=1}^{c} P_k diag(\tilde{\Sigma}_k)^m \right)^{1/m}\right|}, \tag{28}$$

$$\frac{\partial}{\partial \mathrm{B}_{[p]}} \log J_{PLDA}\left(\mathrm{B}_{[p]}, m\right) = 2\Sigma_n \mathrm{B}_{[p]} \tilde{\Sigma}_n^{-1} - 2\mathrm{F}_m \mathrm{G}_m, \tag{29}$$

where

$$\mathrm{F}_m = \sum_{k=1}^{c} P_k \Sigma_k \mathrm{B}_{[p]} diag(\tilde{\Sigma}_k)^{m-1}, \tag{30}$$

$$\mathrm{G}_m = \left(\sum_{k=1}^{c} P_k diag(\tilde{\Sigma}_k)^m \right)^{-1}, \tag{31}$$

and *diag* is an operator which sets zero for off-diagonal elements. In Eq. (28), the control parameter m can be any real number, unlike in Eq. (27).
When m is equal to zero, the PLDA objective function corresponds to the diagonal HDA (DHDA) objective function introduced in (Saon et al., 1990).

5. Selection of an optimal control parameter

As shown in the previous section, PLDA can describe various criteria by varying its control parameter m. One way of obtaining an optimal control parameter m is to train HMMs and test recognition performance on a development set changing m and to choose the m with the smallest error. Unfortunately, this raises a considerable problem in a speech recognition task. In general, to train HMMs and to test recognition performance on a development set for finding an optimal control parameter requires several dozen hours. PLDA requires considerable time to select an optimal control parameter because it is able to choose a control parameter within a real number.
In this section we focus on a class separability error of the features in the projected space instead of using a recognition error on a development set. Better recognition performance can be obtained under the lower class separability error of projected features. Consequently, we measure the class separability error and use it as a criterion for the recognition performance comparison. We define a class separability error of projected features.

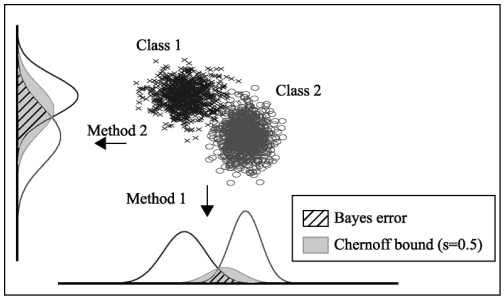


Fig. 2. Comparison of Bayes error and Chernoff bound.

5.1 Two-class problem

This section focuses on the two-class case. We first consider the Bayes error of the projected features on training data as a class separability error:

$$\varepsilon = \int \min\left[P_1 p_1(\mathrm{x}), P_2 p_2(\mathrm{x})\right] d\mathrm{x}, \tag{32}$$

where P_i denotes a prior probability of class i and $p_i(\mathbf{x})$ is a conditional density function of class i. The Bayes error ε can represent a classification error, assuming that training data and evaluation data come from the same distributions. However, it is difficult to directly measure the Bayes error. Instead, we use the Chernoff bound between class 1 and class 2 as a class separability error (Fukunaga, 1990):

$$\varepsilon_u^{1,2} = P_1^s P_2^{1-s} \int P_1^s(\mathrm{x}) P_2^{1-s}(\mathrm{x})\, d\mathrm{x} \qquad \text{for} \quad 0 \le \mathrm{s} \le 1 \tag{33}$$

where ε_u indicates an upper bound of ε. In addition, when the $p_i(\mathbf{x})$'s are normal with mean vectors $\boldsymbol{\mu}_i$ and covariance matrices $\boldsymbol{\Sigma}_i$, the Chernoff bound between class 1 and class 2 becomes

$$\varepsilon_u^{1,2} = P_1^s P_2^{1-s} \exp(-\eta^{1,2}(s)), \tag{34}$$

where

$$\eta^{1,2}(s) = \frac{s(1-s)}{2}(\mu_1 - \mu_2)^T \Sigma_{12}^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\ln\frac{|\Sigma_{12}|}{|\Sigma_1|^s |\Sigma_2|^{1-s}}, \tag{35}$$

where $\boldsymbol{\Sigma}_{12} \equiv s\boldsymbol{\Sigma}_1 + (1-s)\boldsymbol{\Sigma}_2$. In this case, ε_u can be obtained analytically and calculated rapidly. In Fig. 2, two-dimensional two-class data are projected onto one-dimensional subspaces by two methods. To compare with their Chernoff bounds, the lower class separability error is obtained from the projected features by Method 1 as compared with those by Method 2. In this case, Method 1 preserving the lower class separability error should be selected.

5.2 Extension to multi-class problem

In Section 5.1, we defined a class separability error for two-class data. Here, we extend a two-class case to a multi-class case. Unlike the two-class case, it is possible to define several error functions for multi-class data. We define an error function as follows:

$$\tilde{\varepsilon}_u = \sum_{i=1}^{c}\sum_{j=1}^{c} I(i,j)\varepsilon_u^{i,j} \tag{36}$$

where $I(\cdot)$ denotes an indicator function. We consider the following three formulations as an indicator function.

5.2.1 Sum of pairwise approximated errors

The sum of all the pairwise Chernoff bounds is defined using the following indicator function:

$$I(i,j) = \begin{cases} 1, & \text{if} \quad j > i, \\ 0, & \text{otherwise.} \end{cases} \tag{37}$$

5.2.2 Maximum pairwise approximated error

The maximum pairwise Chernoff bound is defined using the following indicator function:

$$I(i,j)=\begin{cases}1, & \text{if} \quad j>i \quad and \quad (i,j)=(\hat{i},\hat{j}),\\ 0, & \text{otherwise},\end{cases} \tag{38}$$

where $(\hat{i},\hat{j})\equiv \arg\max_{i,j} \varepsilon_u^{i,j}$.

5.2.3 Sum of maximum approximated errors in each class

The sum of the maximum pairwise Chernoff bounds in each class is defined using the following indicator function:

$$I(i,j)=\begin{cases}1, & \text{if} \quad j=\hat{j}_i,\\ 0, & \text{otherwise},\end{cases} \tag{39}$$

where $\hat{j}_i \equiv \arg\max_{j} \varepsilon_u^{i,j}$.

6. Combination of feature transformation and discriminative training

Feature transformation aims to transform high dimensional features to low dimensional features in a feature space while separating different classes such as monophones. Discriminative trainings, such as maximum mutual information (MMI) (Bahl et al., 1986) and minimum phone error (MPE) (Povey & Woodland, 2002), estimate the acoustic models discriminatively in a model space (Fig. 3). Because feature transformation and discriminative training are adopted at different levels, a combination of them can have a complementary effect on speech recognition.

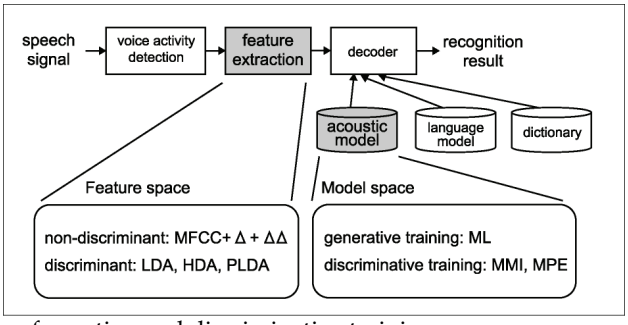


Fig. 3. Feature transformation and discriminative training.

6.1 Maximum mutual information (MMI)

The MMI criterion is defined as follows (Bahl et al., 1986):

$$F_{MMI}(\lambda)=\sum_{r=1}^{R}\log\frac{p_\lambda(\mathrm{O}_r\mid s_r)^\kappa P(s_r)}{\sum_s p_\lambda(\mathrm{O}_r\mid s)^\kappa P(s)}, \tag{40}$$

where λ is the set of HMM parameters, $\mathbf{O}_r$ is the r'th training sentence, R denotes the number of training sentences, κ is an acoustic de-weighting factor which can be adjusted to improve the test set performance, $p_\lambda(\mathbf{O}_r|s)$ is the likelihood given sentence s, and $P(s)$ is the language model probability for sentence s. The MMI criterion equals the multiplication of the posterior probabilities of the correct sentences s_r.

6.2 Minimum phone error (MPE)

MPE training aims to minimize the phone classification error (or maximize the phone accuracy) (Povey & Woodland, 2002). The objective function to be maximized by the MPE training is expressed as

$$F_{MPE}(\lambda) = \sum_{r=1}^{R} \frac{\sum_s p_\lambda(\mathrm{O}_r \mid s)^\kappa P(s) A(s, s_r)}{\sum_{s'} p_\lambda(\mathrm{O}_r \mid s')^\kappa P(s')}, \tag{41}$$

where $A(s,s_r)$ represents the raw phone transcription accuracy of the sentence s given the correct sentence s_r, which equals the number of correct phones minus the number of errors.

7. Experiments

We conducted experiments on CENSREC-3 database (Fujimoto et al., 2006), which is designed as an evaluation framework for Japanese isolated word recognition in real in-car environments. Speech data were collected using two microphones: a close-talking (CT) microphone and a hands-free (HF) microphone. The data recorded with an HF microphone tend to have higher noise than those recorded with a CT microphone because the HF microphone is attached to the driver's sun visor. For training of HMMs, a driver's speech of phonetically-balanced sentences was recorded under two conditions: while idling and driving on city streets under a normal in-car environment. A total of 28,100 utterances spoken by 293 drivers (202 males and 91 females) were recorded with both microphones. We used all utterances recorded with CT and HF microphones for training. For evaluation, we used driver's speech of isolated words recorded with CT and HF microphones under a normal in-car environment and evaluated 2,646 utterances spoken by 18 speakers (8 males and 10 females) for each microphone. The speech signals for training and evaluation were both sampled at 16 kHz.

7.1 Baseline system

In the CENSREC-3, the baseline scripts are designed to facilitate HMM training and evaluation by HTK (available at http://htk.eng.cam.ac.uk/). The acoustic models consisted of triphone HMMs. Each HMM had five states and three of them had output distributions. Each distribution was represented with 32 mixture diagonal Gaussians. The total number of states with the distributions was 2,000. The feature vector consisted of 12 MFCCs and log-energy with their corresponding delta and acceleration coefficients (total 39 dimensions). Frame length and frame shift were 20 msec and 10 msec, respectively. In the Mel-filter bank analysis, a cut-off was applied to frequency components lower than 250 Hz. The decoding process was performed without any language model. The vocabulary size was 100 words, which included the original fifty words and another fifty similar-sounding words.

7.2 Dimensionality reduction procedure

The dimensionality reduction was performed using PCA, LDA, HDA, DHDA (Saon et al., 2000), and PLDA for concatenated features. Eleven successive frames (143 dimensions) were reduced to 39 dimensions. In (D)HDA and PLDA, to optimize Eq. (28), we assumed that projected covariance matrices were diagonal and used the limited-memory BFGS algorithm as a numerical optimization technique. The LDA transformation matrix was used as the initial gradient matrix. To assign one of the classes to every feature after dimensionality reduction, HMM state labels were generated for the training data by a state-level forced alignment algorithm using a well-trained HMM system. The class number was 43 corresponding to the number of the monophones.

7.3 Experimental results

Tables 1 and 2 show the word error rates and class separability errors according to Eqs. (37)-(39) for each dimensionality reduction criterion. The evaluation sets used in Tables 1 and 2 were recorded with CT and HF microphones, respectively. For the evaluation data recorded with a CT microphone, Table 1 shows that PLDA with $m = -0.5$ yields the lowest WER. For the evaluation data recorded with a HF microphone, the lowest WER is obtained by PLDA with a different control parameter ($m = -1.5$) in Table 2. In both cases with CT and HF microphones, PLDA with the optimal control parameters consistently outperformed the other criteria. Two data sets recorded with different microphones had different optimal control parameters. The analysis on the training data revealed that the voiced sounds had larger variances while the unvoiced sounds had smaller ones. As described in Section 4.3, PLDA with a smaller control parameter gives greater importance to the discrimination of classes with smaller variances. Thus, PLDA with a smaller control parameter has better ability to discriminate unvoiced sounds. In general, under noisy environment as with an HF microphone, discrimination of unvoiced sounds becomes difficult. Therefore, the optimal control parameter m for an HF microphone is smaller than with a CT microphone. In comparing dimensionality reduction criteria without training HMMs nor testing recognition performance on a development set, we used $s = 1/2$ for the Chernoff bound computation because there was no *a priori* information about weights of two class distributions. In the case of $s = 1/2$, Eq. (33) is called the Bhattacharyya bound. Two covariance matrices in Eq. (35) were treated as diagonal because diagonal Gaussians were used to model HMMs. The parameter selection was performed as follows: To select the optimal control parameter for the data set recorded with a CT microphone, all the training data with a CT microphone were labeled with monophones using a forced alignment recognizer. Then, each monophone was modeled as a unimodal normal distribution, and the mean vector and covariance matrix of each class were calculated. Chernoff bounds were obtained using these mean vectors and covariance matrices. The optimal control parameter for the data set with an HF microphone was obtained using all of the training data with an HF microphone through the same process as a CT microphone. Both Tables 1 and 2 show that the results of the proposed method and relative recognition performance agree well. There was little difference in the parameter selection performances among Eqs. (37)-(39) in parameter selection accuracy. The proposed selection method yielded sub-optimal performance without training HMMs nor testing recognition performance on a development set, although it neglected time information of speech feature sequences to measure a class

separability error and modeled a class distribution as a unimodal normal distribution. In addition, the optimal control parameter value can vary with different speech features, a different language, or a different noise environment. The proposed selection method can adapt to such variations.

	m	WER	Eq. (37)	Eq. (38)	Eq. (39)
MFCC+ Δ + ΔΔ	-	7.45	2.31	0.0322	0.575
PCA	-	10.58	3.36	0.0354	0.669
LDA	-	8.78	3.10	0.0354	0.641
HDA	-	7.94	2.99	0.0361	0.635
PLDA	-3.0	6.73	2.02	0.0319	0.531
PLDA	-2.0	7.29	2.07	0.0316	0.532
PLDA	-1.5	6.27	**1.97**	0.0307	0.523
PLDA	-1.0	6.92	1.99	0.0301	**0.521**
PLDA	-0.5	**6.12**	2.01	**0.0292**	0.525
DHDA (PLDA)	- (0.0)	7.41	2.15	0.0296	0.541
PLDA	0.5	7.29	2.41	0.0306	0.560
PLDA	1.0	9.33	3.09	0.0354	0.641
PLDA	1.5	8.96	4.61	0.0394	0.742
PLDA	2.0	8.58	4.65	0.0404	0.745
PLDA	3.0	9.41	4.73	0.0413	0.756

Table 1. Word error rates (%) and class separability errors according to Eqs. (37)-(39) for the evaluation set with a CT microphone. The best results are highlighted in bold.

	m	WER	Eq. (37)	Eq. (38)	Eq. (39)
MFCC+ Δ + ΔΔ	-	15.04	2.56	0.0356	0.648
PCA	-	19.39	3.65	0.0377	0.738
LDA	-	15.80	3.38	0.0370	0.711
HDA	-	17.16	3.21	0.0371	0.697
PLDA	-3.0	15.04	2.19	0.0338	0.600
PLDA	-2.0	12.32	2.26	0.0339	0.602
PLDA	-1.5	**10.70**	**2.18**	0.0332	**0.5921**
PLDA	-1.0	11.49	2.23	**0.0327**	0.5922
PLDA	-0.5	12.51	2.31	0.0329	0.598
DHDA (PLDA)	- (0.0)	14.17	2.50	0.0331	0.619
PLDA	0.5	13.53	2.81	0.0341	0.644
PLDA	1.0	16.97	3.38	0.0370	0.711
PLDA	1.5	17.31	5.13	0.0403	0.828
PLDA	2.0	15.91	5.22	0.0412	0.835
PLDA	3.0	16.36	5.36	0.0424	0.850

Table 2. Word error rates (%) and class separability errors according to Eqs. (37)-(39) for the evaluation set with an HF microphone.

7.4 Discriminative training results

We also conducted the same experiments using MMI and MPE by HTK and compared a maximum likelihood (ML) training, MMI, approximate MPE and exact MPE. The approximate MPE assigns approximate correctness to phones while the exact MPE assigns exact correctness to phones. The former is faster in computation for assigning correctness, and the latter is more precise in correctness. The results are shown in Tables 3 and 4. By combining PLDA and the discriminative training techniques, we obtained better performance than the PLDA with a maximum likelihood criterion training. There appears to be no consistent difference between approximate and exact MPE as reported in a discriminative training study (Povey, 2003).

	ML	MMI	MPE (approx.)	MPE (exact)
MFCC+ Δ + $\Delta\Delta$	7.45	7.14	6.92	6.95
PLDA	6.12	5.71	5.06	4.99

Table 3. Word error rates (%) using a maximum likelihood training and three discriminative trainings for the evaluation set with a CT microphone.

	ML	MMI	MPE (approx.)	MPE (exact)
MFCC+ Δ + $\Delta\Delta$	15.04	14.44	18.67	15.99
PLDA	10.70	10.39	9.44	10.28

Table 4. Word error rates (%) using a maximum likelihood training and three discriminative trainings for the evaluation set with an HF microphone.

7.5 Computational costs

The computational costs for the evaluation of recognition performance versus the proposed selection method are shown in Table 5. Here, the computational cost involves the optimization procedure of the control parameter. In this experiment, we evaluate the computational costs on the evaluation data set with a Pentium IV 2.8 GHz computer. For every dimensionality reduction criterion, the evaluation of recognition performance required 15 hours for training of HMMs and five hours for test on a development set. In total, 220 hours were required for comparing 11 feature transformations (PLDAs using 11 different control parameters). On the other hand, the proposed selection method only required approximately 30 minutes for calculating statistical values such as mean vectors and covariance matrices of each class in the original space. After this, 2 minutes were required to calculate Eqs. (37)-(39) for each feature transformation. In total, only 0.87 hour was required for predicting the sub-optimal feature transformation among the 11 feature transformation described above. Thus, the proposed method could perform the prediction process much faster than a conventional procedure that included training of HMMs and test of recognition performance on a development set.

conventional	220 h = (15 h (training) + 5 h (test)) $\times$ 11 conditoins
proposed	0.87 h = 30 min (mean and variance calculations) + 2 min (Chernoff bound calculation) $\times$11 conditions

Table 5. Computational costs with the conventional and proposed methods.

8. Conclusions

In this chapter we presented a new framework for integrating various criteria to reduce dimensionality. The framework, termed power linear discriminant analysis, includes LDA, HLDA and HDA criteria as special cases. Next, an efficient selection method of an optimal PLDA control parameter was introduced. The method used the Chernoff bound as a measure of a class separability error, which was the upper bound of the Bayes error. The experimental results on the CENSREC-3 database demonstrated that segmental unit input HMM with PLDA gave better performance than the others and that PLDA with a control parameter selected by the presented efficient selection method yielded sub-optimal performance with a drastic reduction of computational costs.

9. References

Bahl, L., Brown, P., de Sousa, P. & Mercer, R. (1986). Maximul mutual information estimation of hidden Markov model parameters for speech recognition, *Proceedings of IEEE Int. Conf. on Acoustic Speech and Signal Processing*, pp. 49-52.

Campbell, N. A. (1984). Canonical variate analysis – A general model formulation, *Australian Journal of Statistics*, Vol.4, pp. 86-96.

Fujimoto, M., Takeda, K. & Nakamura, S. (2006). CENSREC-3 : An evaluation framework for Japanese speech recognition in real driving-car environments, *IEICE Trans. Inf. & Syst.*, Vol. E89-D, pp. 2783-2793.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Academic Press, New York.

Haeb-Umbach, R. & Ney, H. (1992). Linear discriminant analysis for improved large vocabulary continuous speech recognition. *Proceedings of IEEE Int. Conf. on Acoustic Speech and Signal Processing*, pp. 13-16.

Kumar, N. & Andreou, A. G. (1998). Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition, *Speech Communication*, pp. 283-297.

Magnus, J. R. & Neudecker, H. (1999). *Matrix Differential Calculus with Applications in Statistics and Econometrics*, John Wiley & Sons.

Nakagawa, S. & Yamamoto, K. (1996). Evaluation of segmental unit input HMM. *Proceedings of IEEE Int. Conf. on Acoustic Speech and Signal Processing*, pp. 439-442.

Povey, D. & Woodland, P. (2002). Minimum phone error and I-smoothing for improved discriminative training, *Proceedings of IEEE Int. Conf. on Acoustic Speech and Signal Processing*, pp. 105-108.

Povey, D. (2003). *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. Thesis, Cambridge University.

Sakai, M., Kitaoka, N. & Nakagawa, S. (2007). Generalization of linear discriminant analysis used in segmental unit input HMM for speech recognition, *Proceedings of IEEE Int. Conf. on Acoustic Speech and Signal Processing*, pp. 333-336.

Saon, G., Padmanabhan, M., Gopinath, R. & Chen, S. (2000). Maximum likelihood discriminant feature spaces, *Proceedings of IEEE Int. Conf. on Acoustic Speech and Signal Processing*, pp. 129-132.

Acoustic Modelling

7

Algorithms for Joint Evaluation of Multiple Speech Patterns for Automatic Speech Recognition

Nishanth Ulhas Nair and T.V. Sreenivas
Department of Electrical Communication Engineering,
Indian Institute of Science, Bangalore - 560012,
India

1. Introduction

Improving speech recognition performance in the presence of noise and interference continues to be a challenging problem. Automatic Speech Recognition (ASR) systems work well when the test and training conditions match. In real world environments there is often a mismatch between testing and training conditions. Various factors like additive noise, acoustic echo, and speaker accent, affect the speech recognition performance. Since ASR is a statistical pattern recognition problem, if the test patterns are unlike anything used to train the models, errors are bound to occur, due to feature vector mismatch. Various approaches to robustness have been proposed in the ASR literature contributing to mainly two topics: (i) reducing the variability in the feature vectors or (ii) modify the statistical model parameters to suit the noisy condition. While some of the techniques are quite effective, we would like to examine robustness from a different perspective. Considering the analogy of human communication over telephones, it is quite common to ask the person speaking to us, to repeat certain portions of their speech, because we don't understand it. This happens more often in the presence of background noise where the intelligibility of speech is affected significantly. Although exact nature of how humans decode multiple repetitions of speech is not known, it is quite possible that we use the combined knowledge of the multiple utterances and decode the unclear part of speech. Majority of ASR algorithms do not address this issue, except in very specific issues such as pronunciation modeling. We recognize that under very high noise conditions or bursty error channels, such as in packet communication where packets get dropped, it would be beneficial to take the approach of repeated utterances for robust ASR. We have formulated a set of algorithms for both joint evaluation/decoding for recognizing noisy test utterances as well as utilize the same formulation for selective training of Hidden Markov Models (HMMs), again for robust performance. Evaluating the algorithms on a speaker independent confusable word Isolated Word Recognition (IWR) task under noisy conditions has shown significant improvement in performance over the baseline systems which do not utilize such joint evaluation strategy.

A simultaneous decoding algorithm using multiple utterances to derive one or more allophonic transcriptions for each word was proposed in [Wu & Gupta, 1999]. The goal of a

simultaneous decoding algorithm is to find one optimal allophone sequence W^* for all input utterances U_1, U_2,. . ., U_n. Assuming independence among U_i, according to the Bayes criterion, W^* can be computed as

$$\begin{aligned} W^* &= \arg\max_W P(W/U_1, U_2, \ldots, U_n) \\ &= \arg\max_W P(U_1, U_2, \ldots, U_n/W)P(W) \\ &= \arg\max_W P(U_1/W)P(U_2/W)\ldots P(U_n/W)P(W) \end{aligned} \quad (1)$$

where $P(X)$, stands for the probability of the event X occurring.

From an information theoretic approach, consider two speech sequences $U1$ and $U2$. The joint entropy $H(U1, U2)$ will be higher than either of their individual entropies $H(U1)$ or $H(U2)$ [Shannon, 1948]. We know that if $U1$ and $U2$ are completely independent of each other then the joint entropy $H(U1,U2)$ will be equal to $H(U1) + H(U2)$. If they are completely dependent $H(U1,U2) = H(U1) = H(U2)$. When $U1$ and $U2$ come from the same class, there is a degree of correlation between them. Particularly when parts of $U1$ or $U2$ is corrupted, then the joint entropy would have a higher difference with respect to either of the individual entropies. This is because the noise is more random in nature. This applies to > 2 sequences also.. The goal of the pattern recognition task is to exploit this higher information entropy in a maximum likelihood (ML) framework for better recognition.

One direct approach to simultaneous decoding is to use the N-best criteria [Nilsson, 1971, Schwartz & Chow, 1990, Soong & Hung, 1991]. In this, an individual N-best list for each input utterance is generated independently using the N-best search algorithm of statistical decoding. These individual N-best lists are merged and rescored using all the input utterances [Haeb-Umbach et al., 1995]; based on their joint likelihoods the transcriptions are re-ordered. However this solution is suboptimal unless N is very large [Wu & Gupta, 1999].

Simultaneous decoding for multiple input utterances can be done using a modified version of the tree-trellis search algorithm [Soong & Hung, 1991] (the same algorithm was used in [Holter et al., 1998]). A forward Viterbi beam search for each utterance is performed independently, and then a combined backward A^* search [Bahl et al., 1983] for all the utterances is applied simultaneously. A word-network-based algorithm is also developed for simultaneous decoding. This algorithm has been shown to be computationally very efficient [Wu & Gupta, 1999].

Multiple utterances of same speech unit has been typically used in pronunciation estimation. Pronunciation determined using only one recording of a word can be very unreliable. So for more reliability, modeling multiple recordings of a word is used. However commonly used decoding algorithms are not suited to discover a phoneme sequence that jointly maximizes the likelihood of all the inputs.To arrive at the same solution, various alternative techniques have been proposed. One method is to produce recognition lattices individually from each of the inputs, and identify the most likely path in the intersection of these lattices. Another generates N-best hypotheses from each of the audio inputs and re-scores the cumulative set jointly with all the recordings [Singh et al., 2002, Svendson, 2004]. Alternately, the pronunciations may be derived by voting amongst the recognition outputs from the individual recordings [Fiscus, 1997]. While all of these procedures result in outputs that are superior to what might be obtained using only one recorded instance of the word, they nevertheless do not truly identify the most likely pronunciation for the given set of recordings, and thus remain suboptimal. So it is important to jointly estimate the pronunciation from multiple recordings.

Dealing with multiple speech patterns occurs naturally during the training stage. In most, the patterns are considered as just independent exemplars of a random process, whose parameters are being determined. There is some work in the literature to make the ML training process of statistical model, such as HMM, more robust or better discriminative. For example, it is more difficult to discriminate between the words "rock" and "rack", than between the words "rock" and "elephant". To address such issues, there has been attempts to increase the separability among similar confusible classes, using multiple training patterns.

In discriminative training, the focus is on increasing the separable distance between the models, generally their means. Therefore the model is changed. In selective training the models are not forced to fit the training data, but deemphasizes the data which does not fit the models well. In [Arslan & Hansen, 1996, Arslan & Hansen, 1999], each training pattern is selectively weighted by a confidence measure in order to control the influence of outliers, for accent and language identification application. Adaptation methods for selective training, where the training speakers close to the test speaker are chosen based on the likelihood of speaker Gaussian Mixture Models (GMMs) given the adaptation data, is done in [Yoshizawa et al., 2001]. By combining precomputed HMM sufficient statistics for the training data of the selected speakers, the adapted model is constructed. In [Huang et. al, 2004], cohort models close to the test speaker are selected, transformed and combined linearly. Using the methods in [Yoshizawa et al., 2001, Huang et. al, 2004], it is not possible to select data from a large data pool, if the speaker label of each utterance is unknown or if there are only few utterances per speaker. This can be the case when data is collected automatically, e.g. the dialogue system for public use such as Takemaru-kun [Nishimura et al., 2003]. Selective training of acoustic models by deleting single patterns from a data pool temporarily or alternating between successive deletion or addition of patterns has been proposed in [Cincarek et al., 2005].

In this chapter, we formulate the problem of increasing ASR performance given multiple utterances (patterns) of the same word. Given K test patterns ($K \geq 2$) of a word, we would like to improve the speech recognition accuracy over a single test pattern, for the case of both clean and noisy speech. We try to jointly recognize multiple speech patterns such that the unreliable or corrupt portions of speech are given less weight during recognition while the clean portions of speech are given a higher weight. We also find the state sequence which best represents the K patterns. Although the work is done for isolated word recognition, it can also be extended to connected word and continuous speech recognition. To the best of our knowledge, the problem that we are formulating has not been addressed before in speech recognition.

Next, we propose a new method to selectively train HMMs by jointly evaluating multiple training patterns. In the selective training papers, the outlier patterns are considered unreliable and are given a very low (or zero) weighting. But it is possible that only some portions of these outlier data are unreliable. For example, if some training patterns are affected by burst/transient noise (e.g. bird call) then it would make sense to give a lesser weighting to only the affected portion. Using the above joint formulation, we propose a new method to train HMMs by selectively weighting regions of speech such that the unreliable regions in the patterns are given a lower weight. We introduce the concept of "virtual training patterns" and the HMM is trained using the virtual training patterns instead of the

original training data. We thus address all the three main tasks of HMMs by jointly evaluating multiple speech patterns.

The outline of the chapter is as follows: sections 2 and 3 gives different approaches to solve the problem of joint recognition of multiple speech patterns. In section 4, the new method of selectively training HMMs using multiple speech patterns jointly is proposed. Section 5 gives the experimental evaluations for the proposed algorithms, followed by conclusions in section 6.

2. Multi Pattern Dynamic Time Warping (MPDTW)

The Dynamic Time Warping (DTW) [Rabiner & Juang, 1993, Myers et al., 1980, Sakoe & Chiba, 1978] is a formulation to find a warping function that provides the least distortion between any two given patterns; the optimum solution is determined through the dynamic programming methodology. DTW can be viewed as a pattern dissimilarity measure with embedded time normalization and alignment. We extend this formulation to multiple patterns greater than two, resulting in the multi pattern dynamic time warping (MPDTW) algorithm [Nair & Sreenivas, 2007, Nair & Sreenivas, 2008 b]. The algorithm determines the optimum path in the multi-dimensional discrete space to optimally warp all the K number of patterns jointly, leading to the minimum distortion path, referred to as MPDTW path. As in standard DTW, all K patterns are warped with respect to each other. The MPDTW algorithm finds the least distortion between the K patterns and the MPDTW algorithm reduces to the DTW algorithm for K = 2. To find the MPDTW path, we need to traverse through the Kdimensional grid along the K time axes. Let the K patterns be $\mathbf{O}^1_{1:T_1}$, $\mathbf{O}^2_{1:T_2}$,. . ., $\mathbf{O}^K_{1:T_K}$,, of lengths T_1, T_2,. . ., T_K, respectively, where $\mathbf{O}^i_{1:T_i} = (O^i_1, O^i_2, \ldots, O^i_{T_i})$ is the observation vector sequence of the i^{th} pattern and $O^i_{t_i}$ is the feature vector at time frame t_i.

Fig. 1 shows an example MPDTW path for K = 2; it is the least distortion path between the two patterns. We define a path P of grid traversing as a sequence of steps in the grid, each specified by a set of coordinate *increments* [Rabiner & Juang, 1993], i.e.,

$$P \rightarrow (p^1_1, p^2_1, \ldots, p^K_1), (p^1_2, p^2_2, \ldots, p^K_2), \ldots, (p^1_T, p^2_T, \ldots, p^K_T) \tag{2}$$

where p^i_t is the increment at step t along utterance i (i^{th} dimension).

Let the t = 1 step correspond to (1, 1,. . ., 1), is the staring point in the grid where all the K patterns begin. Let us set $p^1_1 = p^2_1 = \ldots = p^K_1 = 1$. Let t = T correspond to (T_1, T_2, . . . , T_K), which is the ending point of the multi-dimensional grid. $\phi_1(t)$, $\phi_2(t)$,. . ., $\phi_K(t)$ are K warping functions such that $\phi_i(t) = t_i$ for the i^{th} pattern. Let us define:

$$\phi_l(t) = \sum_{i=1}^{t} p^l_i, \qquad l = 1, 2, \ldots, K \tag{3}$$

The coordinate *increments* satisfy the constraints:

$$\sum_{t=1}^{T} p^l_t = T_l, \qquad l = 1, 2, \ldots, K \tag{4}$$

Endpoint constraints are:

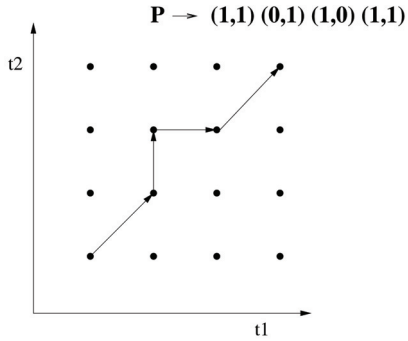


Fig. 1. An example MPDTW path for $K = 2$.

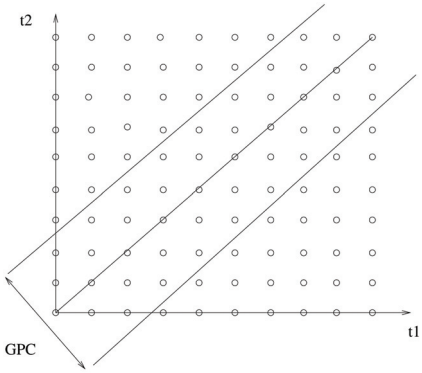


Fig. 2. An example Global Path Constraint for MPDTW for $K = 2$.

$$\phi_1(1) = 1, \ldots, \phi_K(1) = 1 \tag{5}$$

$$\phi_1(T) = T_1, \ldots, \phi_K(T) = T_K \tag{6}$$

Relaxed end pointing can also be introduced as in standard DTW. Various types of Local Continuity Constraints (LCC) and Global Path Constraints (GPC) as defined for DTW [Rabiner & Juang, 1993], can be extended to the K dimensional space. The LCC we used is similar to the simplest Type-1 LCC used in DTW, except that it has K dimensions. The point $(t_1, t_2, \ldots, t_K)$ can be reached from any one of the points $(t_1 - i_1, t_2 - i_2, \ldots, t_K - i_K)$ where $i_k = 0, 1$ for $k = 1, 2, \ldots, K$. This leads to $(2^K - 1)$ predecessor paths, excluding the all-zero combination. One type of GPC for MPDTW when $K = 2$ is shown in Fig. 2. It can be extended for any K. For e.g., if $K = 3$ the GPC will look like a square cylinder around the diagonal.

The important issue in MPDTW is the distortion measure between patterns being compared. Since the goal is to minimize an accumulated distortion along the warping path, we define a positive distortion metric at each end of the node of the grid traversing. We define a joint distance measure $d(t_1, t_2, \ldots, t_K)$ between the K vectors $O^1_{t_1}, O^2_{t_2}, \ldots, O^K_{t_K}$, as follows:

$$d(t_1, \ldots, t_K) = \sum_{i=1}^{K} d(O^i_{t_i}, C_{\phi(t)}), \tag{7}$$

where $C_{\phi(t)}$ is the centroid of the K vectors $O^1_{t_1}, O^2_{t_2}, \ldots, O^K_{t_K}$ and $d(O^i_{t_i}, C_{\phi(t)})$ is the Euclidean distance between $O^i_{t_i}$ and $C_{\phi(t)}$ and $\phi(t) = (t_1, t_2, \ldots, t_K) = (\phi_1(t), \phi_2(t), \ldots, \phi_K(t))$. This measure is generalizable to any perceptually important measures such as the Itakura-Saito measure [Itakura & Saito, 1968] and a generalized centroid as defined in Vector Quantization [Gersho & Gray, 1992]. In Fig. 3, three vectors $O1$, $O2$, $O3$ and their Euclidean centroid C is shown. The joint distance measure between $O1$, $O2$, $O3$ is $d(O1,C) + d(O2,C) + d(O3,C)$, where $d(O,C)$ is the Euclidean distance between vector O and vector C.
To discourage the optimum path close to the diagonal, the slope weighting function $m(t)$ is utilized. The final accumulated distortion is normalized using

$$M_\phi = \sum_{t=1}^{T} m(t) \tag{8}$$

Thus the total distortion to be minimized is

$$D_{total} = \sum_{t=1}^{T} m(t) \sum_{j=1}^{K} d(O^j_{t_j}, C_{\phi(t)})/M_\phi \tag{9}$$

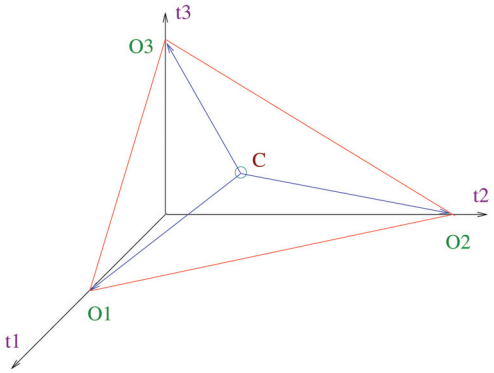


Fig. 3. 3 vectors $O1$, $O2$, $O3$ and their centroid C.

2.1 MPDTW Algorithm

Let $D_{total} = D(t_1, t_2, \ldots, t_K)$ be the accumulated cost function, which is to be minimized.

a) Initialization

$$D(1, \ldots, 1) = d(1, \ldots, 1)m(1) \tag{10}$$

b) Recursion

For $1 \le t_1 \le T_1$, $1 \le t_2 \le T_2, \ldots, 1 \le t_K \le T_K$, such that $t_1, t_2, \ldots, t_K$ lie within the allowable grid.

$$\begin{aligned} D(t_1, \ldots, t_K) \quad = \quad & \min_{(t'_1, \ldots, t'_K)} [D(t'_1, \ldots, t_K{}') + \\ & \zeta((t'_1, \ldots, t'_K)(t_1, \ldots, t_K))] \end{aligned} \tag{11}$$

where $(t'_1, \ldots, t'_K)$ are the candidate values as given by the LCC and

$$\zeta((t_1^{'},\ldots,t_K^{'})(t_1,\ldots,t_K)) \quad = \quad \sum_{l=0}^{L_s} d(\phi_1(T^{'}-l),\ldots,\phi_K(T^{'}-l))m(T^{'}-l) \tag{12}$$

$\phi_1(T^{'}) = t_1,\ldots, \phi_K(T^{'}) = t_K$ and $\phi_1(T^{'}-L_s) = t_1^{'},\ldots, \phi_K(T^{'}-L_s) = t_K^{'}$ where L_s being the number of moves in the path from (t'_1, . . . , t'_K) to (t_1, . . . , t_K) according to ϕ_1,. . ., ϕ_K. A backtracking pointer I is defined to remember the best local choice in equation 11, which will be used for the global path backtracking.

$$I(t_1,\ldots,t_K) \quad = \quad \arg \min_{(t_1^{'},\ldots,t_K^{'})} [D(t_1^{'},\ldots,t_K^{'}) + \zeta((t_1^{'},\ldots,t_K^{'})(t_1,\ldots,t_K))] \tag{13}$$

c) Termination

$$\mathbf{d}(\mathbf{O}^1_{1:T_1},\ldots,\mathbf{O}^K_{1:T_K}) = D(T_1,\ldots,T_K)/M_\phi \tag{14}$$

where $\mathbf{d}(\mathbf{O}^1_{1:T_1},\ldots,\mathbf{O}^K_{1:T_K})$ is the total distortion between the K patterns $\mathbf{O}^1_{1:T_1},\ldots,\mathbf{O}^K_{1:T_K}$; this is the best distortion measure under the constraints of the MPDTW algorithm.

d) Path Backtracking

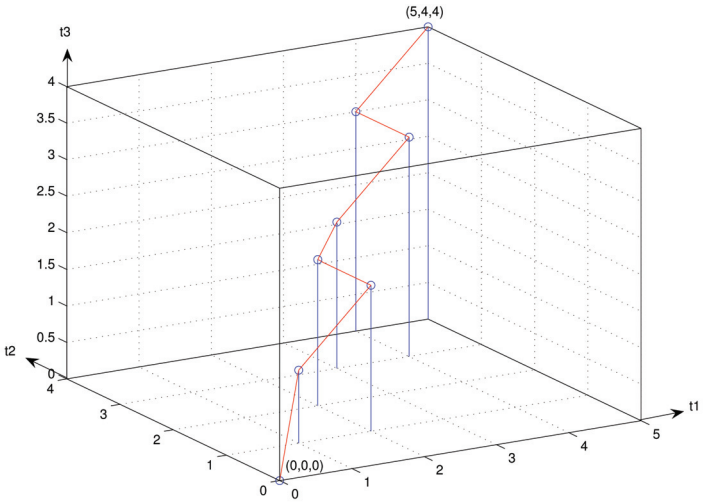


Fig. 4. An example MPDTW path for K = 3 patterns.

The optimum warping between the K patterns is obtained through backtracking, starting from the end point and "hopping" through the back-pointer grid, i.e.,

$$(t_1^*,\ldots,t_K^*) = I(t_1,\ldots,t_K) \tag{15}$$

$$(t_1,\ldots,t_K) = (t_1^*,\ldots,t_K^*) \tag{16}$$

where (t_1, . . . , t_K) = (T_1, . . . , T_K), . . . , (1, . . . , 1).

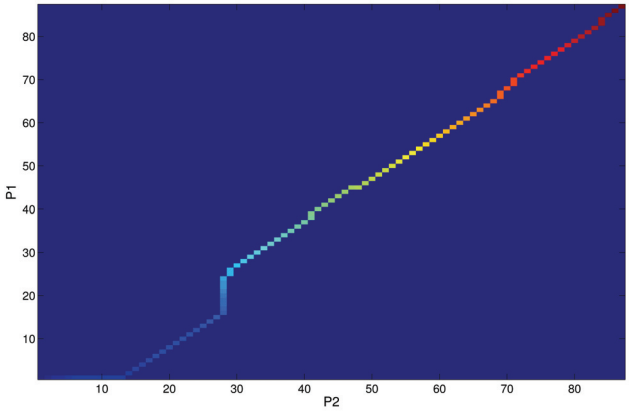


Fig. 5. MPDTW path for 3 patterns P1, P2, P3 projected on P1-P2 plane. The first 30% of the frames (frame number 1 to 27) in P2 is noisy at -15 dB SNR. P1 and P3 are clean.

The least distortion path, referred to as MPDTW path, gives us the most similar non linear time warping path between them. Let ϕ be the MPDTW path for K patterns. $\phi(t) = (t_1, \ldots, t_K)$, where $(t_1, \ldots, t_K)$ is a point on the MPDTW path. $\phi(1) = (1, \ldots, 1)$ and $\phi(T) = (T_1, \ldots, T_K)$. So $\phi = (\phi(1), \phi(2), \ldots, \phi(t), \ldots, \phi(T))$.

An example MPDTW path for $K = 3$ patterns is shown in Fig. 4. A projection of an example MPDTW path for 3 speech patterns (P1, P2, P3) on the P1-P2 plane is shown in Fig. 5, where burst noise at -15 dB Signal to Noise Ratio (SNR) is added to the first 30% in the speech pattern P2. All the three patterns belong to the word "Voice Dialer" by one female speaker. The feature vector used to represent speech was Mel Frequency Cepstral Coefficients (MFCC), Δ MFCC, Δ^2 MFCC without the energy component (36 dimension vector). Notice that the initial portion of the MPDTW path has a deviation from the diagonal path but it comes back to it. Fig. 6 shows the MPDTW path when burst noise at -5 dB SNR is added to 10% frames in the beginning portion of pattern P2. We don't see too much of a deviation from the diagonal path. This tells us that the MPDTW algorithm is relatively robust to burst noise using only 3 patterns (K= 3). This clearly shows that we can use the MPDTWalgorithm to align K patterns coming from same class even if they are affected by burst/transient noise. We will use this property to our advantage later.

2.2 MPDTW for IWR

Since MPDTW is a generalization of the DTW, it opens new alternatives to the basic problem of IWR. In a basic IWR, the test pattern and reference pattern are compared, resulting in an optimum warping path in 2-Dimension. But we have a solution in K dimensions. Hence, we can choose a variable number of test and reference patterns [Nair & Sreenivas, 2008 b]. Hence, let there be r reference patterns per class (word) and $K - r$ test patterns. We can compare the minimum distortion of the test patterns with respect to the reference patterns of different words using the MPDTW algorithm. Whichever word reference set gives the lowest distortion, it can be selected as the matched word. It should be noted that the recognition performance will be different from the 2D-DTW and likely more robust, because multiple patterns are involved both for testing and as reference templates.

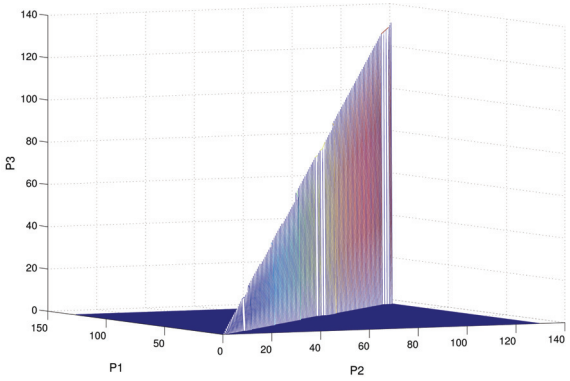


Fig. 6. MPDTW path for 3 patterns of the word "Voice Dialer". The first 10% of pattern P2 is noisy at -5 dB SNR. Patterns P1 and P3 are clean.

Case 1: Multiple Test Patterns and One Reference Pattern

We have $r = 1$ and $K - r > 1$. When the number of test patterns is more than 1, they together produce a "reinforcing" effect as there is more information. So when the $K - r$ test patterns are compared with the correct reference pattern, the distortion between the K patterns would be less, and when they are compared with the wrong reference pattern, the distortion is likely to be much higher. The discriminability between the distortions using the correct and wrong reference patterns and its robustness is likely to increase as we are using more than 1 test pattern. Therefore the recognition accuracy is likely to increase as the number of test patterns ($K - r$) increases.

Case 2: One Test Pattern and Multiple Reference Patterns

In this case we have multiple reference patterns but only one test pattern; i.e., $r > 1$ and $K - r = 1$. This MPDTWalgorithm will be repeated for different vocabulary to recognize the word. For the sake of simplicity consider an example that has only 2 reference patterns (R_1 and R_2) and one test pattern (P_1). We find the MPDTWpath (least distortion path) in the multi dimensional grid between these 3 patterns using the MPDTW algorithm. Project this MPDTW path on any of the planes containing 2 patterns, say P_1 and R_1. (We know that the optimum least distortion path between P_1 and R_1 is given by the DTW algorithm.) The projected MPDTW path on the plane containing P_1 and R_1 need not be same as least distortion path given by the DTW algorithm. Hence it is a suboptimal path and the distortion obtained is also not optimal. So taking the distortion between P_1 and R_1 (or P_1 and R_2) using the DTW algorithm, leads to lower total distortion between the 2 patterns than using a projection of the MPDTW algorithm. This sub optimality is likely to widen with increasing r, the number of reference patterns. This property holds good for incorrectly matched reference patterns also. However, since the distortion is high, the additional increase due to joint pattern matching may not be significant. So it is likely that the MPDTW algorithm will give a poorer discriminative performance than the 2-D DTW algorithm, as the number of reference patterns (r) per class increase. The use of multiple templates is common in speaker dependent ASR applications, to model the pronunciation variability. When the templates of the same class (vocabulary word) are significantly different, the joint recognition is likely to worsen the performance much more than otherwise.

Case 3: Multiple Test and Reference Patterns

This is the most general case of $r > 1$ and $K - r > 1$. From the discussions mentioned in cases 1, 2, and 3, we know that the recognition accuracy is likely to increase as $K - r$ increases and decrease when r increases. When both reference patterns and test patterns are likely to be noisy or distorted, $r > 1$ and $K - r > 1$ will likely to lead to more robust performance.

3. Joint likelihood of multiple speech patterns

Considering left to right stochastic models of speech patterns, we now propose a new method to recognize K patterns jointly by finding their joint multi pattern likelihood, i.e., $P(\mathbf{O}^1_{1:T_1}, \mathbf{O}^2_{1:T_2}, \ldots, \mathbf{O}^K_{1:T_K}/\lambda)$. We assume that the stochastic model is good, but some or all of the test patterns may be distorted due to burst/transient noise or even badly pronounced. We would like to jointly recognize them in an "intelligent" way such that the noisy or unreliable portions of speech are neglected and more weightage is given to the reliable portions of the patterns. Such a solution would be better than single pattern separate recognition.

As before, we denote the K number of observed speech sequences (patterns) $\mathbf{O}^1_{1:T_1}, \mathbf{O}^2_{1:T_2}, \ldots, \mathbf{O}^K_{1:T_K}$ of lengths $T_1, T_2, \ldots, T_K$, respectively, where $\mathbf{O}^i_{1:T_i} = (O^i_1, O^i_2, \ldots, O^i_{T_i})$ and $O^i_{t_i}$ is the feature vector of the ith pattern at time frame t_i. Let each of these K sequences belong to the same pattern class (spoken word). They are repeated patterns of the same word. In the standard HMM decoding, we have a trellis structure in $K + 1$ dimensions, where K dimensions belong to the K patterns and one dimension belongs to the HMM states. Let $\mathbf{q}$ be any HMM state sequence jointly decoding the K patterns. N is the total number of HMM states.

$$P(\mathbf{O}^1_{1:T_1}, \mathbf{O}^2_{1:T_2}, \ldots, \mathbf{O}^K_{1:T_K}/\lambda) = \sum_{\forall \mathbf{q}} P(\mathbf{O}^1_{1:T_1}, \mathbf{O}^2_{1:T_2}, \ldots, \mathbf{O}^K_{1:T_K}, \mathbf{q}/\lambda) \tag{17}$$

We can calculate the joint multi pattern likelihood only over the optimum HMM state sequence q^* for K patterns as shown:

$$P(\mathbf{O}^1_{1:T_1}, \mathbf{O}^2_{1:T_2}, \ldots, \mathbf{O}^K_{1:T_K}, \mathbf{q}^*/\lambda) = \max_{\forall \mathbf{q}} P(\mathbf{O}^1_{1:T_1}, \mathbf{O}^2_{1:T_2}, \ldots, \mathbf{O}^K_{1:T_K}, \mathbf{q}/\lambda) \tag{18}$$

We can find the optimum HMM state sequence q^* as follows:

$$\begin{aligned} \mathbf{q}^* &= \arg\max_{\mathbf{q}} log\ P(\mathbf{q}/\mathbf{O}^1_{1:T_1}, \ldots, \mathbf{O}^K_{1:T_K}, \lambda) \\ &= \arg\max_{\mathbf{q}} log\ P(\mathbf{q}, \mathbf{O}^1_{1:T_1}, \ldots, \mathbf{O}^K_{1:T_K}/\lambda) \end{aligned} \tag{19}$$

Fig. 7 shows a schematic of two patterns $O^1_{1:T_1}$ and $O^2_{1:T_2}$ and the time alignment of the two patterns along the optimum HMM state sequence $\mathbf{q}^*$ is shown. This is opposite to the case we see in [Lleida & Rose, 2000]. In [Lleida & Rose, 2000], a 3D HMM search space and a Viterbi-like decoding algorithm was proposed for Utterance Verification. In that paper, the two axes in the trellis belonged to HMM states and one axis belongs to the observational sequence. However, here (equation 19) we have K observational sequences as the K axis, and one axis for the HMM states. We would like to estimate one state sequence by jointly decoding the K patterns since we know that the K patterns come from the same class. They are conditionally independent, that is, they are independent given that they come from the

same class. But, there is a strong correlation between the K patterns because they belong to the same class. The states in a Left to Right HMM roughly correspond to the stationary phonemes of the pattern and hence use of the same sequence is well justified. The advantage is this is that we can do discriminant operations like frame based voting, etc., as we will be shown later. This formulation is more complicated for Left to Right HMMs with skips or even more general ergodic HMMs.

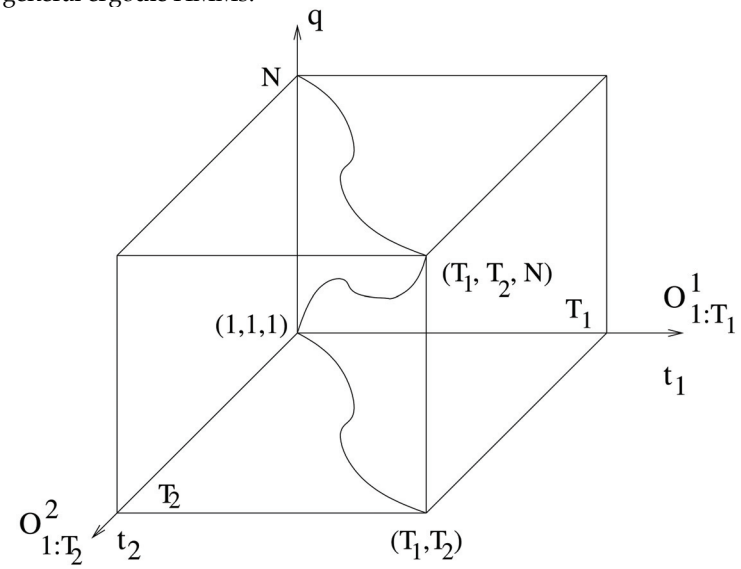


Fig. 7. A multi dimensional grid with patterns $O^1_{1:T_1}$, $O^2_{1:T_2}$ forming a trellis along the q-axis, and the optimum state sequence **q***. If N = 3 states, then an example **q***could be [1 1 1 2 2 3 3 3].

To find the total probability of K patterns given λ - $P(\mathbf{O}^1_{1:T_1}, \mathbf{O}^2_{1:T_2}, \ldots, \mathbf{O}^K_{1:T_K}/\lambda)$- we have to traverse through a trellis of K+1 dimensions. This leads to a high-dimensional Viterbi search in which both the state transition probabilities as well as local warping constraints of multiple patterns have to be accommodated. We found this to be somewhat difficult and it did not yield consistent results. Hence, the problem is simplified by recasting it as a two stage approach of joint recognition, given the optimum alignment between the various patterns. This alignment between the K patterns can be found using the Multi Pattern Dynamic Time Warping (MPDTW) algorithm. This is followed by one of the Multi Pattern Joint Likelihood (MPJL) algorithms to determine the joint multi pattern likelihood and the best state sequence for the K patterns jointly [Nair & Sreenivas, 2007, Nair & Sreenivas, 2008 a]. The twostage approach can also be viewed as a hybrid of both non-parametric ASR and parametric (stochastic) ASR, because we use both the non-parametric MPDTW and parametric HMM for speech recognition (Fig. 8). There is also a reduction in the computational complexity and search path from K + 1 dimensions to K dimensions, because of this two stage approach. We experimented with the new algorithms for both clean speech and speech with burst and other transient noises for IWR and show it to be advantageous. We note that similar formulations are possible for connected word recognition and continuous speech recognition tasks also. We thus come up with solutions to address the first two problems of HMMs using joint evaluation techniques.

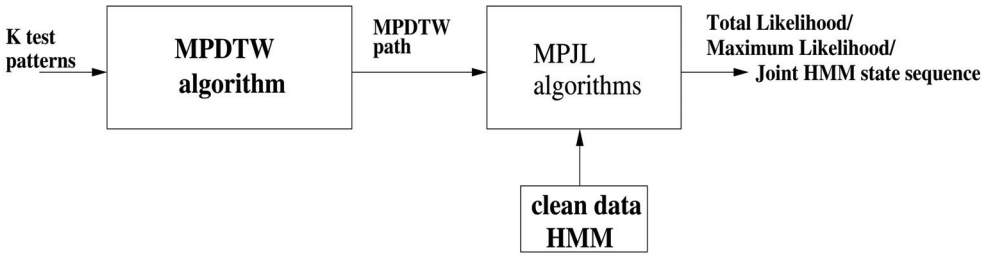


Fig. 8. Joint Likelihood of K patterns.

The MPDTW algorithm finds the least distortion mapping between the K patterns, referred to as the MPDTW path. MPDTW path gives us a handle to evaluate similar/dissimilar components of the K patterns visa-vis the HMM. Note that here all the K patterns are test patterns and there is no reference pattern (the three cases of section 2.2 does not apply). Let ϕ be the MPDTW mapping between K patterns and $\phi(t) = (t_1, \ldots, t_K)$ where $(t_1, \ldots, t_K)$ is a K tuple coordinate sequence along the MPDTW optimum path, in the K dimensional rectangular grid. Endpoint errors are not considered here since the MPDTW algorithm can accommodate relaxed end point constraints separately. Since the MPDTW path is the least distortion path between K patterns of the same class, the path groups similar sounding portions from the K patterns. Each point on the MPDTW path represents K feature vectors, each coming from one pattern.

Since our goal is to determine the joint likelihood of all the K patterns given the HMM, we can consider the MPDTW path $\phi(t)$, $1 \le t \le T$ as a 1-D evolution of the multiple patterns. Let T be the total number of distinct points in the MPDTW path and $T \ge \max_k\{T_k\}$. Thus, we can use $\phi(t)$ as a virtual path for the evolution of multiple patterns and construct the HMM trellis; the trellis comprises of N states × T sequence of K-vector sets. The joint likelihood of K vector sequences as seen along $\phi(t)$ is determined by using one of the MPJL algorithms. The MPJL algorithms are divided into two versions of determining either the total probability or the ML state sequence probability; Constrained Multi Pattern Forward Algorithm (CMPFA), Constrained Multi Pattern Backward Algorithm (CMPBA) determine total probability using either the forward or backward algorithm, and Constrained Multi Pattern Viterbi Algorithm (CMPVA) for the Viterbi score (ML state sequence). These algorithms are called "constrained" because their time path is fixed by the MPDTW algorithm.

The main feature of the MPJL algorithms is to calculate the total probability in an "intelligent" manner such that we are making use of the "best" information available and avoiding (giving less weight) to the noisy or unreliable information among multiple patterns.

3.1 Constrained Multi Pattern Forward and Backward algorithms (CMPFA and CMPBA)

The CMPFA and CMPBA are used to calculate the total joint probability of the K patterns through all possible HMM state sequences. Following the terminology of a standard HMM [Rabiner & Juang, 1993] for the forward algorithm, we define the forward variable $\alpha_{\phi(t)}(j)$ along the path $\phi(t)$; i.e.,

$$\alpha_{\phi(t)}(j) = P(\mathbf{O}^1_{1:t_1}, \mathbf{O}^2_{1:t_2}, \ldots, \mathbf{O}^K_{1:t_K}, q_{\phi(t)} = j/\lambda) \tag{20}$$

where $q_{\phi(t)}$ is the HMM state at $t \rightarrow \phi(t)$, λ is the HMM model with states $j \in 1 : N$. As in the forward algorithm, we can determine $\alpha_{\phi(t)}(j)$ recursively leading to the total probability.

3.1.1 CMPFA-1

Let us define the MPDTW path transition vector, $\Delta\phi(t) = \phi(t) - \phi(t-1) = (\Delta t^1_1, \Delta t^2_2, \ldots, \Delta t^K_K)$. Depending on the local constraints chosen in the MPDTW algorithm, $\Delta\phi(t)$ can be a K dimensional vector of only 0's and 1's; e.g., $\Delta\phi(t)$ = [0, 1, 1, 0, . . . , 1]. $\Delta\phi(t)$ will comprise of at least one non-zero value and a maximum of K non-zero values. (The [0,1] values are due to the non-skipping type of local constraints in MPDTW. The skipping-type can introduce higher range such as [0,1,2] or [0,1,2,3].) Let $S_{\phi(t)} = \{O^i_{t_i} | \Delta t^i_i \neq 0, i = 1, 2, \ldots, K\}$ be the set of vectors that have been mapped together by the MPDTW at $\phi(t)$. Let $\{O_{\phi(t)}\} = (O^m_{t_m}, \ldots, O^n_{t_n})$ such that $(O^m_{t_m}, \ldots, O^n_{t_n})$ are all the feature vectors in the set $S_{\phi(t)}$. $\{O_{\phi(t)}\}$ is a subset of the vectors $(O^1_{t_1}, O^2_{t_2}, \ldots, O^K_{t_K})$, retaining only those $O^k_{t_k}$ whose Δt^k_k are non-zero. The set $S_{\phi(t)}$ and $\{O_{\phi(t)}\}$ can have a minimum of one feature vector and a maximum of K feature vectors. Let $\phi(t)^* = \{t_i | \Delta t^i_i \neq 0, i = 1, 2, \ldots, K\}$. The recursive equation for the evaluation of $\alpha_{\phi(t)}(j)$ along all the grid points of the trellis is given by (derivation given in appendix $A1$):

$$\alpha_{\phi(t)}(j) = \sum_{i=1}^{N} \left[\alpha_{\phi(t-1)}(i).a_{ij} \right] .b_j \left(\{O_{\phi(t)}\} \right), \tag{21}$$

t = 2, 3, . . . , T , j = 1, 2, . . . , N. a_{ij} is the state transition probability from state i to state j (as in standard HMM), $b_j(\{O_{\phi(t)}\})$ is the joint likelihood of $\{O_{\phi(t)}\}$ being emitted by state j. It is the same as joint likelihood of all the vectors $(O^m_{t_m}, \ldots, O^n_{t_n})$ emitted by state j, where $(O^m_{t_m}, \ldots, O^n_{t_n})$ consist of all the feature vectors in a set $S_{\phi(t)}$. Thus, a HMM state-j can emit a variable number of vectors from the K patterns, corresponding to the number of non-zero values in the $\Delta\phi(t)$ vector. Thus, the number of feature vectors each state emits ranges from 1 to K and it is a function of the MPDTW path. But, when the recursive computation of $\alpha_{\phi(t)}$ reaches $\alpha_{\phi(T)}$, each state j would have emitted the exact number of multi-pattern feature vectors = $(T_1 + T_2 + \ldots + T_K)$, irrespective of the MPDTW path.

We examine a few alternate methods to calculate the joint likelihood $b_j(\{O_{\phi(t)}\})$, shown in section 3.3. We know from HMM theory, that a HMM state can emit one feature vector at any given time. However in our case, each HMM state emits 1 upto K feature vectors at each time coordinate $\phi(t)$ and a given state j. However, we calculate the joint likelihood $b_j(\{O_{\phi(t)}\})$ by normalizing it (shown in section 3.3) such that it is comparable to a likelihood of a single vector emitted by a HMM state. This can be interpreted as inherently performing recognition for one virtual pattern created from the K similar patterns. Thus, we can consider the values of transition probability a_{ij} and state initial probability π_i as the same as that used in single pattern recognition task.

An example of MPDTW path, when there are only two patterns (K = 2) is shown in Fig. 9. The t1 time axis is for pattern $O^1_{1:T_1}$ and t2 time axis is for pattern $O^2_{1:T_2}$. We fit a layer of HMM states (of a class) on this path. For simplicity, let us consider that there is only one state j = 1. Now we traverse along this MPDTW path. In the trellis, for CMPFA-1, at time instant (1,1) feature vectors (O^1_1, O^2_1) are emitted by state j. At time instant (2,2) state j emits

vectors O_2^1 and O_2^2. At time (3,2) state *j* emits only one vector O_3^1 and not O_2^2, as O_2^2 was already emitted at time (2,2). So we are exactly recognizing all the *K* patterns such that there is no reuse of feature vectors. The total number of feature vectors emitted at the end of the MPDTW path by each state in this example will be exactly equal to $T_1 + T_2$.

3.1.2 CMPFA-2

CMPFA-2 is an approximate solution for calculating the total probability of the *K* patterns given HMM λ but it has some advantages over CMPFA-1. In CMPFA-2, a recursive solution (equation (22)) is proposed to calculate the value of $\alpha_{\phi(t)}(j)$. This solution is based on the principle that each HMM state *j* can emit a fixed (*K*) number of feature vectors for each transition. So, it is possible that some of the feature vectors from some patterns are reused based on the MPDTWpath. This corresponds to stretching each individual pattern to a fixed length *T* (which is $\geq \max_k\{T_k\}$) and then determining the joint likelihood for the given HMM. Thus, we are creating a new virtual pattern which is a combination of all the *K* patterns (with repetitions of feature vectors possible) of length equal to that of the MPDTWpath. The HMM is used to determine the likelihood of this virtual pattern. The total number of feature vectors emitted in this case is *K.T* . Considering the forward recursion as before,

$$\alpha_{\phi(t)}(j) = \sum_{i=1}^{N} [\, \alpha_{\phi(t-1)}(i).a_{ij}].b_j(O_{t_1}^1, \ldots, O_{t_K}^K) \tag{22}$$

t = 2, 3, . . . , T , j = 1, 2, . . . , N, where N is the number of states in the HMM. a_{ij} is the state transition probability from state i to state j, π_j is the state initial probability at state j, $b_j(O_{t_1}^1, \ldots, O_{t_K}^K)$ is the joint likelihood of the observations $O_{t_1}^1, \ldots, O_{t_K}^K$ generated by state j. The various methods to calculate this joint likelihood $b_j(O_{t_1}^1, \ldots, O_{t_K}^K)$ is shown in section 3.3.

Let us consider again the example of Fig. 9. In CMPFA-2, each HMM state emits a fixed number (*K*) of feature vectors. In CMPFA-2, at time instant (1,1), feature vectors (O_1^1, O_1^2) are emitted by state *j*. At time instant (2,2), state *j* emits vectors O_2^1 and O_2^2. At time instant (3,2), state *j* emits vectors O_3^1 and O_2^2. At time instant (4,2), state *j* emits vectors O_4^1 and O_2^2. Here we see that vector O_2^2 is emitted 3 times. So some feature vectors are repeated in CMPFA-2 based on the MPDTW path. So by using CMPFA-2, the HMMs are not emitting the *K* patterns in an exact manner.

The initialization for both CMPFA-1 and CMPFA-2 is the same, i.e.,

$$\alpha_{\phi(1)}(j) = \pi_j.b_j(O_1^1, \ldots, O_1^K) \tag{23}$$

where j = 1, 2, . . . ,N, π_j is the state initial probability at state j (and it is assumed to be same as the state initial probability given by the HMM), $b_j(O_{t_1}^1, \ldots, O_{t_K}^K)$ is the joint likelihood of the observations $O_{t_1}^1, \ldots, O_{t_K}^K$ generated by state j.

The termination of CMPFA-1 and CMPFA-2 is also same:

$$P^* = \sum_{i=1}^{N} \alpha_{\phi(T)}(i), \tag{24}$$

where P^* is the total joint likelihood of *K* patterns.

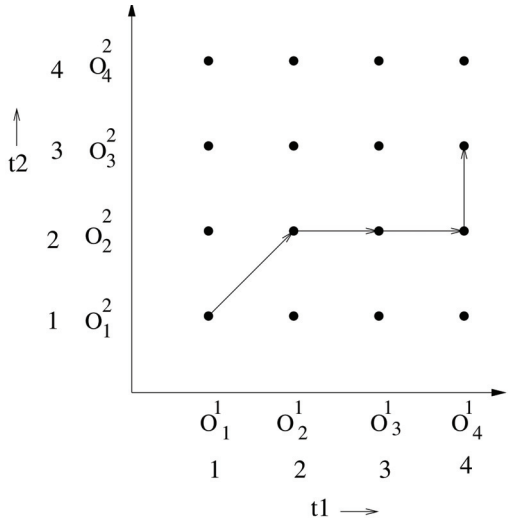


Fig. 9. An example path.

3.1.3 CMPBA-1 and CMPBA-2

We can easily construct backward recursion algorithms similar to the forward recursion of the previous section, which leads to CMPBA- 1 and CMPBA-2. We define $\beta_{\phi(t)}(j)$ as in equation (25).

$$\beta_{\phi(t-1)}(j) = P(O^1_{t_1:T_1}, \ldots, O^K_{t_K:T_K} / q_{\phi(t-1)} = j, \lambda) \tag{25}$$

For CMPBA-1 can write similar recursive equation as in CMPFA-1, by using the backward cumulative probability.

$$\beta_{\phi(t-1)}(i) = \sum_{j=1}^{N} a_{ij}.\beta_{\phi(t)}(j).b_j(\{O_{\phi(t)}\}) \tag{26}$$

t = T, . . . , 2, i = 1, 2, . . . ,N, where N is the number of states in the HMM, and the rest of the terms are as defined in section 3.1.1. Again, CMPBA-2 is an approximation to calculate $\beta_{\phi(t)}(j)$ and is similar to CMPFA-2. We define the recursive solution for calculating $\beta_{\phi(t)}(j)$ as follows.

$$\beta_{\phi(t-1)}(i) = \sum_{j=1}^{N} a_{ij}.\beta_{\phi(t)}(j).b_j(O^1_{t_1}, \ldots, O^K_{t_K}) \tag{27}$$

t = *T*, . . . , 2, *i* = 1, 2, . . . , *N*, where *N* is the number of states in the HMM.

3.2 Constrained Multi Pattern Viterbi Algorithms (CMPVA-1 and CMPVA-2)

Unlike the CMPFAs and CMPBAs which consider all state sequences through the HMM trellis, the CMPVA is evaluated for the probability along the maximum likelihood (ML) state sequence: $\mathbf{q}^* = q^*_{\phi(1)}, q^*_{\phi(2)}, \ldots, q^*_{\phi(T)}$, where $q^*_{\phi(t)}$ is the optimum state at time $\phi(t)$. We

define $\delta_{\phi(t)}(j)$ as the log likelihood of the first $\phi(t)$ observations of the K patterns through the best partial state sequence up to the position $\phi(t-1)$ and $q_{\phi(t)} = j$ along the path $\phi(t)$; i.e.,

$$\delta_{\phi(t)}(j) = \max_{q_{\phi(1):\phi(t-1)}} log\ P(\mathbf{O}^1_{1:t_1}, \ldots, \mathbf{O}^K_{1:t_K}, q_{\phi(1):\phi(t-1)}, q_{\phi(t)} = j/\lambda) \tag{28}$$

where $q_{\phi(1):\phi(T)} = q_{\phi(1)}, q_{\phi(2)}, \ldots, q_{\phi(T)}$. The recursive equation for CMPVA-1 (similar to CMPFA-1) is:

$$\delta_{\phi(t)}(j) = \max_i[\ \delta_{\phi(t-1)}(i) + log\ a_{ij}] + log\ b_j(\{O_{\phi(t)}\}) \tag{29}$$

with $t = 2, 3, \ldots, T$ and $j = 1, 2, \ldots, N$, N is the number of HMM states. The terminology is similar to that used in equation (21). The recursive solution for CMPVA-2 (similar to CMPFA-2) is:

$$\delta_{\phi(t)}(j) = \max_i[\ \delta_{\phi(t-1)}(i) + log\ a_{ij}] + log\ b_j(O^1_{t_1}, \ldots, O^K_{t_K}) \tag{30}$$

$t = (2, 3, \ldots, T)$, $j = 1, 2, \ldots, N$
Initialization for both CMPVA-1 and CMPVA-2 is done as follows:

$$\delta_{\phi(1)}(j) = log\ \pi_j + log\ b_j(O^1_1, \ldots, O^K_1), \qquad j = 1, 2, \ldots, N \tag{31}$$

The path backtracking pointer $\psi_{\phi(t)}(j)$ for CMPVA-1 and CMPVA-2 is:

$$\psi_{\phi(t)}(j) = \arg\max_i[\ \delta_{\phi(t-1)}(i) + log\ a_{ij}] \tag{32}$$

where $t = 2, 3, \ldots, T$ and $j = 1, 2, \ldots, N$.
The ML joint likelihood for both CMPVA-1 and CMPVA-2 is determined by:

$$P^* = \max_i\{\delta_{\phi(T)}(i)\} \tag{33}$$

Path Backtracking is done to find the optimum state sequence.

$$q^*_{\phi(t)} = \psi_{\phi(t+1)}(q^*_{\phi(t+1)}), \qquad t = T-1, \ldots, 1 \tag{34}$$

An example of a HMM state sequence along the MPDTW path is shown in Fig. 10.
For robust IWR, we use either CMPFA or CMPBA or CMPVA to calculate the probability P^* of the optimal sequence. For simplicity let us group CMPFA-1, CMPBA-1, CMPVA-1 as CMP?A-1 set of algorithms; and CMPFA-2, CMPBA-2, CMPVA-2 as CMP?A-2 set of algorithms.

3.3 Feature vector weighting

In missing feature theory [Cooke et al., 1994, Cooke et al., 2001], we can identify the unreliable (noisy) feature vectors, and either ignore them in subsequent processing, or they can be filled in by the optimal estimate of their putative value [Raj & Stern, 2005]. Similar to this approach we determine the joint likelihoods for any of the six algorithms discussed earlier, by differentiating as to which portions of the speech patterns are unreliable. We can

give a lesser or zero weighting to the unreliable (noisy) feature vectors and a higher weighting to the corresponding reliable ones from the other patterns. Fig. 11 shows an example of two speech patterns affected by burst noise. The strategy is to give a lesser weight to the regions of speech contaminated by burst noise and the corresponding clean speech in the other pattern should be given a higher weight. This can be interpreted as a form of voting technique which is embedded into HMM decoding. We have considered alternate criteria for weighting the feature vectors, to achieve robustness to transient, bursty noise in the test patterns.

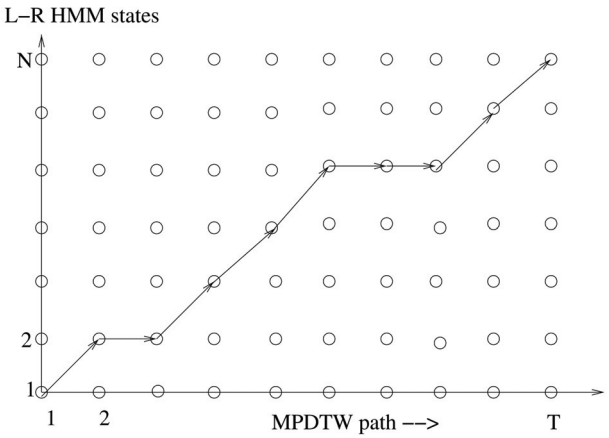


Fig. 10. Example of a HMM state sequence along MPDTW path for Left-Right HMM.

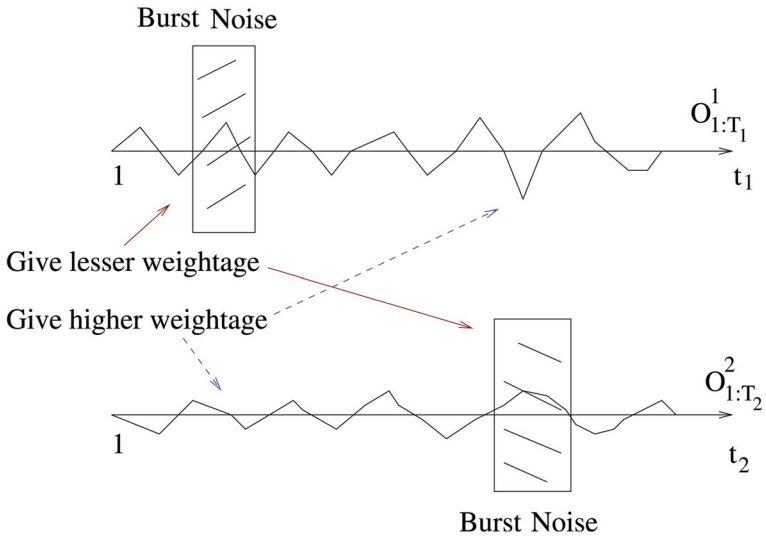


Fig. 11. Two speech patterns $O^1_{1:T_1}$ and $O^2_{1:T_2}$ affected by burst noise.

To determine the joint likelihood of emitting the set of feature vectors in a given state, we can resort to alternate formulations. Since the path traversed along the K-dimensional time

axes is already optimized, how we choose $b_j(\{O_{\phi(t)}\})$ (or $b_j(O^1_{t_1}, O^2_{t_2},\ldots, O^K_{t_K})$) affects only the total/ML joint likelihood P^* (of equations 24 and 32) and the ML state sequence (of equation 33). We define various discriminant criteria for calculating $b_j(\{O_{\phi(t)}\})$ (in equations 21, 26, 29) and $b_j(O^1_{t_1}, O^2_{t_2},\ldots, O^K_{t_K})$ (in equations 22, 27, 30) as follows:

3.3.1 Discriminant Criteria - average (DC-avg)

We know that $\mathbf{O}^1_{1:T_1}, \mathbf{O}^2_{1:T_2},\ldots, \mathbf{O}^K_{1:T_K}$ are different speech patterns which come from the same class (word) and uttered by the same speaker. Given that they come from the same class, and uttered independently, we don't use any discriminating criteria, and simplify use of the vectors as they are. Even though the feature vectors $O^1_{t_1}, O^2_{t_2},\ldots, O^K_{t_K}$ come from the same class, we can assume that they are independent if it is given that they occur from the same state j, so as to compute the joint likelihood of the vectors being emitted from the HMM. So, we can express the joint probability term in CMP?A-1

$$b_j(\{O_{\phi(t)}\}) = [b_j(O^m_{t_m})\ldots b_j(O^n_{t_n})]^{1/r} \tag{35}$$

where $(O^m_{t_m},\ldots, O^n_{t_n})$are all the feature vectors in the set $S_{\phi(t)}$ as mentioned in section 3.1.1 and $b_j(O^i_{t_i})$ is the state-j emission probability for the HMM (probability of vector $O^i_{t_i}$ emitted by state j given the HMM) and r is the cardinality of the set $S_{\phi(t)}$.
Similarly, CMP?A-2, since all the patterns are used at each time, the following equation is proposed.

$$b_j(O^1_{t_1},\ldots, O^K_{t_K}) = [b_j(O^1_{t_1}).b_j(O^2_{t_2})\ldots b_j(O^K_{t_K})]^{1/K} \tag{36}$$

The independence assumption is justified because successive vectors in a pattern are only linked through the underlying Markov model and the emission densities act only one symbol at a time. The geometric mean using power of $1/r$ or $1/K$ normalizes the use of r or K vectors emitted by a HMM state, comparable to a single vector likelihood. Therefore we can use a_{ij}'s and π_i's that are defined as in single-pattern HMM. If $O^i_{t_i}$ is emitted from its actual state j from the correct HMM model λ, we can expect that $b_j(O^i_{t_i})$ to have a higher value than that if $O^i_{t_i}$ is emitted from state j of the wrong model. And taking the product of all the $b_j(O^i_{t_i})$ brings in a kind of "reinforcing effect". Therefore, while doing IWR, the values of joint likelihood P^* using the correct model and the P^* using the mismatched models, is likely to widen. Therefore we can expect better speech recognition accuracy to improve. Even if some of the K vectors are noisy, the reinforcing effect will improve speech recognition because the rest of the vectors are clean.

3.3.2 Discriminant Criteria - maximum (DC-max)

In the first discriminant criteria DC-avg, we estimate the output probability by considering the vectors emitted at each state, in an average sense; i.e., geometric mean of the independent vector probabilities is considered. Instead of the average, we consider the maximum of the independent probabilities from the set $S_{\phi(t)}$. Of course, $S_{\phi(t)}$ itself would be different for the two classes of algorithms for grid traversing (CMP?A-1 and CMP?A-2). Thus, we can express

$$b_j(\{O_{\phi(t)}\}) = \max(b_j(O^m_{t_m}),\ldots, b_j(O^n_{t_n})) \tag{37}$$

$$b_j(O_{t_1}^1, \ldots, O_{t_K}^K) = \max(b_j(O_{t_1}^1), b_j(O_{t_2}^2), \ldots, b_j(O_{t_K}^K)) \tag{38}$$

Because of the maximum (max) operation, we can expect the likelihoods in DC-max to be higher than the respective ones using DC-avg. In terms of the virtual pattern interpretation, the sequence of the T length patterns are composed of the most probable vectors corresponding to the HMM-λ. If the speech patterns are affected by noise, we expect DC-max to give better discrimination than DCavg. However, for the case of clean speech, it is possible that DC-max will reduce speech recognition accuracy than DC-avg because the max operation will also increase the likelihood P^* for the mismatched model and bring it closer to the P^* of the correct model. Also, this reinforcing effect will be absent in DC-max. So we would prefer to use DC-max when the speech patterns are noisy or badly spoken, and for the clean speech case we would prefer DC-avg.

3.3.3 Discriminant Criteria - threshold 1 (DC-thr1)

Instead of using SNR, to determine which feature vectors are reliable or unreliable as in missing feature theory, we propose a novel distortion measure called the joint distance measure. For CMP?A-1 set of algorithms, the joint distance measure is defined as, $d(\phi(t)^*) = \sum_{O_{t_i}^i \in S_{\phi(t)}} d(O_{t_i}^i, C_{\phi(t)^*})$, where $C_{\phi(t)^*}$ is the centroid of all the vectors in $S_{\phi(t)}$ as in section 3.1, and $d(O_{t_i}^i, C_{\phi(t)^*})$ is the Euclidean distance between $O_{t_i}^i$ and $C_{\phi(t)^*}$. We define $b_j(\{O_{\phi(t)}\})$ as:

$$b_j(\{O_{\phi(t)}\}) = \begin{cases} \left[\prod_{O_{t_i}^i \in S_{\phi(t)}} b_j(O_{t_i}^i)\right]^{\frac{1}{r}} & \text{if } d(\phi(t)^*) < \gamma, r = |S_{\phi(t)}| \\ \max_{O_{t_i}^i \in S_{\phi(t)}} b_j(O_{t_i}^i) & \text{if } d(\phi(t)^*) \geq \gamma \end{cases} \tag{39}$$

where γ is a threshold, r is the cardinality of the set $S_{\phi(t)}$.

In equation (39), if we choose $\gamma = \infty$, then $b_j(\{O_{\phi(t)}\})$ is always equal to $\prod_{O_{t_i}^i \in S_{\phi(t)}} b_j(O_{t_i}^i)$ (product operation), and when $\gamma < 0$, then it is always equal to $\max_{O_{t_i}^i \in S_{\phi(t)}} \{b_j(O_{t_i}^i)\}$ (max operation). The first option of $d(\phi(t)^*) < \gamma$ is provided to take care of the statistical variation among the patterns, even without noise. If the distortion is low ($< \gamma$) it implies no noise among the patterns; then we consider all the vectors to be reliable data and set $S_{\phi(t)}$ come from the same class, we can assume that $b_j(\{O_{\phi(t)}\})$ is determined as in DC-avg. When the distortion is high ($> \gamma$), it could be due to misalignment as well as distortion in the patterns. So, we choose only one vector out of all the possible r vectors, which gives the maximum probability in state j. The rest of the $r-1$ vectors are not considered for joint likelihood. This is expressed as the max operation.

For CMP?A-2 set of algorithms, the set $S_{\phi(t)}$ includes all the vectors from the K patterns at $\phi(t)$ and accordingly, the joint distance measure is defined as, $d(t_1, \ldots, t_K) = \sum_{i=1}^{K} d(O_{t_i}^i, C_{\phi(t)^*})$, where $C_{\phi(t)^*}$ is the centroid of the K vectors $(O_{t_1}^1, O_{t_2}^2, \ldots, O_{t_K}^K)$, and $d(O_{t_i}^i, C_{\phi(t)^*})$ is the Euclidean distance between $O_{t_i}^i$ and $C_{\phi(t)^*}$. Rest are similar to equation 39.

If the speech patterns are affected by noise, we would expect that the max operation to give better recognition results and for the case of well articulated clean speech, we expect the

product operation to give better results. We need to select the threshold such that it is optimum for a particular noisy environment.

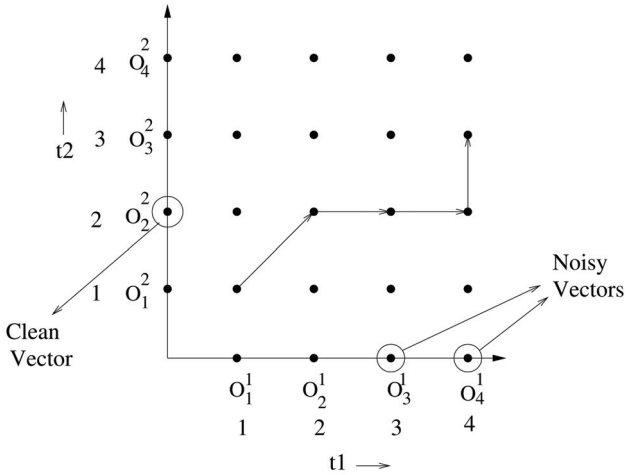


Fig. 12. MPDTW path for K = 2; vector O_2^2 is clean and vectors O_3^1 and O_4^1 are noisy

3.3.4 Discriminant Criteria - threshold 2 (DC-thr2)

In DC-thr1, while doing the max operation, we are taking only the best pattern. In practice a variable number of patterns could be noisy and we would like to use the max operation only to omit the noisy patterns and use the product operation for the rest of the patterns. So we choose only pairwise distortion between two vectors at a time and define a new criteria for the joint likelihood.

Let $1 \le a, b \le K$ be the indices of vectors belonging to the K patterns. For the CMP?A-1 set, we define the clean (undistorted) set of vectors as Z, such that $a, b \in Z$ iff $d(O_{t_a}^a, O_{t_b}^b) < \gamma$, where $d(O_{t_a}^a, O_{t_b}^b)$ is the Euclidean distance between $O_{t_a}^a$ and $O_{t_b}^b$. Let Z be the set of remaining vector indices, such that $Z \cup \overline{Z} = \{m | O_{t_m}^m \in S_{\phi(t)}\}$. Let R is the cardinality of the set $S_{\phi(t)}$ and we can search all pairs of vectors among R exhaustively, i.e., $R(R-1)/2$ combinations, since R is usually small ($R \sim 2, 3$).

$$b_j(O_{t_m}^m, \ldots, O_{t_n}^n) = \begin{cases} \prod_{\{i|i\in Z\}} \left[b_j(O_{t_i}^i)\right]^{\frac{1}{r}} \\ \text{if } Z \neq 0 \\ \max_{\{i|i\in Z\cup\overline{Z}\}} \left(b_j(O_{t_k}^k)\right) \\ \text{if } Z = 0 \end{cases} \tag{40}$$

where r is the cardinality of set Z, and 0 stands for null set.

For the CMP?A-2 set of algorithms $S_{\phi(t)} = (O_{t_1}^1, O_{t_2}^2, \ldots, O_{t_K}^K)$, be the full set of K-pattern vectors and all the steps to compute equation 40 is followed exactly.

Note that DC-thr2 becomes same as DC-thr1 if number of patterns K is equal to 2. There is one disadvantage in this criteria. Consider three vectors $O_{t_1}^1$, $O_{t_2}^2$, $O_{t_3}^3$. Let $O_{t_1}^1$ and $O_{t_2}^2$ be affected by similar kind of noise and let $O_{t_3}^3$ be clean. Then it is possible that the noisy

vectors $O^1_{t_1}$ and $O^2_{t_2}$ are pooled together in set Z and the clean vector $O^3_{t_3}$ is ejected. This can affect speech recognition performance in a negative way. Thus the clean vectors may be removed from probability computation. However this would be a very rare possibility.

3.3.5 Discriminant Criteria - weighted (DC-wtd)

The previous two criteria DC-thr1 and DC-thr2 have one difficulty of choosing a proper threshold γ. We consider a new weighting criteria without the need for the threshold. For the CMP?A-1 set of algorithms, the joint probability is defined as follows:

$$b_j(\{O_{\phi(t)}\}) = \prod_{O^i_{t_i} \in S_{\phi(t)}} b_j(O^i_{t_i})^{\frac{b_j(O^i_{t_i})}{\sum_{O^i_{t_i} \in S_{\phi(t)}} (b_j(O^i_{t_i}))}} \tag{41}$$

For CMP?A-2 set of algorithms, $b_j(O^1_{t_1}, \ldots, O^K_{t_K})$ is defined as follows:

$$b_j(O^1_{t_1}, \ldots, O^K_{t_K}) = \prod_{i=1}^{K} b_j(O^i_{t_i})^{\frac{b_j(O^i_{t_i})}{\sum_{i=1}^{K} (b_j(O^i_{t_i}))}} \tag{42}$$

In this discriminant weighting, the individual probabilities are weighted according to the proportion of their probability value compared with that of all the vectors in $S_{\phi(t)}$. Thus distinctly higher probability values are magnified and distinctly lower probabilities are penalized. The above equations are basically a weighted geometric mean of the $b_j(O^i_{t_i})$'s. Thus, when the values of $b_j(O^i_{t_i})$'s are close to each other, then DC-wtd becomes close to the product operation of DC-thr1 and if the various values $b_j(O^i_{t_i})$'s are very different, then DC-wtd becomes close to max operation of DC-thr1. DC-wtd behaves somewhat similar to DC-thr1 (when DC-thr1 is set with optimum threshold). The main advantage of using DC-wtd is that we don't need to set any threshold. We expect this type of soft-thresholding may be useful in some applications.

3.4. Analysis of CMP?A-1 versus CMP?A-2

Now we analyze which of these two set of algorithms, CMP?A-1 and CMP?A-2, is better and under what conditions. An example of MPDTW path, when there are only two patterns (K = 2) is shown in Fig. 12. The t1 time axis is for pattern $O^1_{1:T_1}$ and t2 time axis is for pattern $O^2_{1:T_2}$. We fit a layer of HMMstates (of a class) on this path. For simplicity, let us consider that there is only one state j = 1. Now we traverse along this MPDTWpath. In the example shown in Fig. 12, we assume that the vector O^2_2 is clean and the vectors O^1_3 and O^1_4 are noisy or badly articulated. Let us use DC-thr1 to calculate joint probability and choose a low value for the threshold, so that the max operation dominates. Using CMP?A-2, since O^2_2 is re-emitted (by state j) at time instants (3,2) and (4,2), the max operation will be most likely used as the joint distortion measure will most likely be above the threshold. So only the clean O^2_2 vector will be used to calculate joint probability. However in the case of CMP?A-1, since O^2_2 is emitted only once at time instant (2,2) and not emitted at time instants (3,2) and (4,2), only the noisy vectors O^1_3 and O^1_4, contribute to the calculation of joint probability. This affects P^*. So in this case the recognition accuracy is likely to decrease if we use CMP?A-1 when compared to CMP?A-2.

Now we consider an other case (Fig. 13) when we are using DC-thr1 and the value of the threshold is very high, so that the product operation dominates. Let vector *O*22 be noisy or badly articulated and vectors O_3^1 and O_4^1 be clean. Since the product operation will mostly be used, using CMP?A-2, the noisy vector O_2^2 will affect the calculation of the joint probability at time instants (3,2) and (4,2) as it is re-emitted. Now using CMP?A-1, as vector O_2^2 is not re-emitted, only the clean vectors O_3^1 and O_4^1 contribute to the calculation of joint probability. So CMP?A-1 is expected to give better speech recognition accuracy than CMP?A-2.

For the case of clean, well articulated speech, CMP?A-1 is expected to perform better than CMP?A-2 as it does not reuse any feature vector. This is true when we use DC-thr1 at lower values of threshold. At higher (sub optimal) values of threshold, CMP?A-2 could be better. If DC-wtd is used, we expect that using CMP?A-1 would give better recognition accuracy than CMP?A-2 for well articulated clean speech and worse values for speech with burst/transient noise or speech with bad articulation. This is because DC-wtd behaves similar to DC-thr1 when the threshold of DC-thr1 is optimum. Finally we conclude that if we look at the best performances of CMP?A-1 and CMP?A-2, CMP?A-2 is better than CMP?A-1 for noisy speech (burst/transient noise), and CMP?A-1 is better than CMP?A-2 for clean speech.

The recognition accuracy is expected to increase with the increase in the number of test patterns *K*.

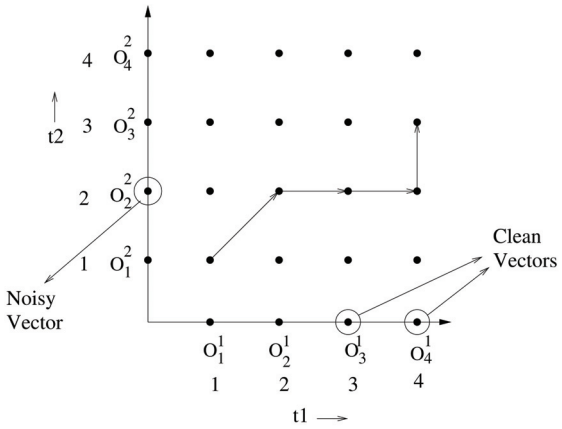


Fig. 13. MPDTW path for *K* = 2; vector O_2^2 is noisy and vectors O_3^1 and O_4^1 are clean

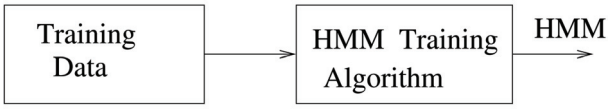


Fig. 14. Standard HMM Training.

4. Selective HMM training (SHT)

This is the next part of the joint multi-pattern formulation for robust ASR. We have first addressed evaluation and decoding tasks of HMM for multiple patterns. Now we consider the benefits of joint multi-pattern likelihood in HMM training. Thus, we would have

addressed all the three main tasks of HMM, to utilize the availability of multiple patterns belonging to the same class. In the usual HMM training, all the training data is utilized to arrive at a best possible parametric model. But, it is possible that the training data is not all genuine and therefore have labeling errors, noise corruptions, or plain outlier examples. In fact, the outliers are addressed explicitly in selective HMM training papers. We believe that the multi-pattern formulation of this chapter can provide some advantages in HMM training also.

Typically, in HMM training the Baum-Welch algorithm [Baum & Petrie, 1966, Baum & Egon, 1967, Baum & Sell, 1968, Baum et al., 1970, Baum, 1972] is used (Fig. 14). We would like to extend it to use the concepts of joint multi-pattern likelihood. Let us refer to this as selective training, in which the goal is to utilize the best portions of patterns for training, omitting any outliers. In selective training, we would like to avoid the influence of corrupted portions of the training patterns, in determining the optimum model parameters. Towards this, virtual training patterns are created to aid the selective training process. The selective training is formulated as an additional iteration loop around the HMM Baum-Welch iterations. Here the virtual patterns are actually created. The virtual patterns can be viewed as input training patterns that have been subjected to "filtering" to deemphasize distorted portions of the input patterns. The filtering process requires two algorithms that we have proposed earlier, viz., MPDTW and CMPVA. The CMPVA uses the MPDTW path as a constraint to derive the joint Viterbi likelihood of a set of patterns, given the HMM λ. CMPVA is an extension of the Viterbi algorithm [Viterbi, 1967] for simultaneously decoding multiple patterns, given the time alignment. It has been shown in [Nair & Sreenivas, 2007, Nair & Sreenivas, 2008 a] that these two algorithms provide significant improvement to speech recognition performance in noise.

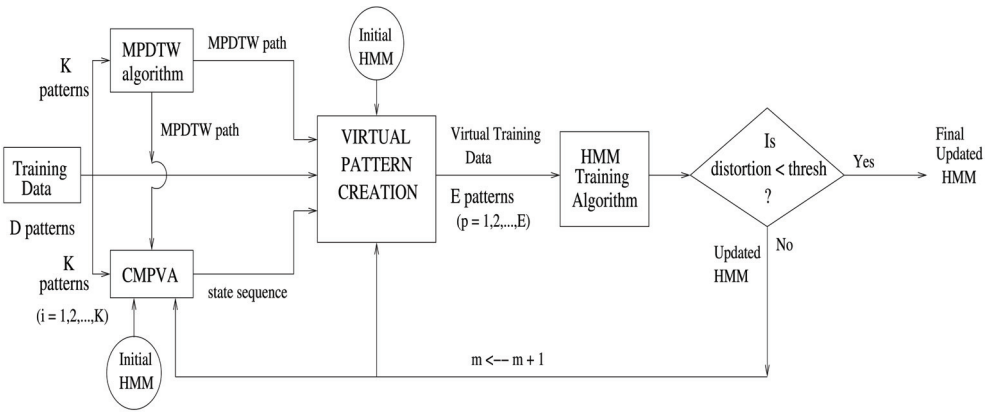


Fig. 15. Selective HMM Training.

A block diagram of the new selective HMM training scheme using virtual patterns is shown in Fig. 15. Let there be D number of training patterns (tokens) in the training data, all of them labeled as belonging to the same class (e.g. words). Let us choose K subset of patterns from these D patterns (here $2 \leq K \leq D$). We know that the K patterns come from the same class and there is strong correlation between them, although they are uttered independently. The K patterns are used to create one virtual pattern by taking advantage of the fact that they are correlated. Then the next K patterns we choose from the pool of D patterns are used

to create another virtual pattern. Let the total maximum number of virtual patterns we can create be equal to *J*. These virtual patterns are now considered independent with respect to every other virtual pattern. All these virtual patterns together constitute the training pattern set which is used for HMM training instead of the original patterns directly. The maximum number of virtual patterns (*J*) given D training patterns is equal to $\sum_{2\leq K\leq D} C_K^D$ (the number of training combinations with at least two in the set), where $C_K^D = \frac{D!}{K!\,(D-K)!}$ and *K*! stands for *K* factorial ($(K! = \prod_{i=1}^{K} i)$.). However since this value can be very high for a large database, we can choose a subset of *E* patterns from *J* patterns, in an intelligent way. These *E* patterns form the virtual training data. The higher the value of *E* the better would be the statistical stability of the estimation and also the various speech variabilities is likely to be modeled better.

Let $\mathbf{O}^1_{1:T_1}$, $\mathbf{O}^2_{1:T_2}$,…, $\mathbf{O}^K_{1:T_K}$ be the *K* patterns selected from the *D* pattern training data. We apply the MPDTW algorithm to find the MPDTW path. The MPDTW path now acts as an alignment to compare similar sounds. Using this MPDTW path we find the HMM state sequence using the CMPVA. It may be noted that even the virtual patterns have different lengths, depending on the MPDTW path.

4.1 Virtual pattern creation

The MPDTW path and the CMPVA decoded HMM state sequence are used as inputs to create a virtual pattern. An initial HMM is found out using the HMM training algorithm on the original training data and the HMM parameters are used to start the iterations. The virtual pattern is created by aligning the *K* patterns and weighting the feature vectors of the different patterns such that the more likely vectors are given a higher weighting and the less reliable vectors are given a lower weighting. The MPDTW path is a global alignment of the input patterns, which does not change through the selective iterations. Let ϕ be the MPDTW mapping between one set of *K* patterns and $\phi(t) = (t_1, \ldots, t_K) = (\phi_1(t), \phi_2(t), \ldots, \phi_K(t))$, where $(t_1, \ldots, t_K)$ is a *K* tuple coordinate sequence belonging to the *K* patterns along the MPDTW optimum path, and $\phi_i(t) = t_i$, such that $\phi_i(t)$ is the projection of the MPDTW path $\phi(t)$ onto the t_i axis. $\phi = (\phi(1), \phi(2), \ldots, \phi(T_p))$, where T_p is the number of points in the MPDTW path such that $T_p \geq \max\{T_1, T_2, \ldots, T_K\}$. Since the MPDTW path provides the least distortion alignment between the *K* patterns we are able to compare all the similar sounds from the multiple patterns and weight them appropriately to create a virtual pattern. The length of the virtual pattern is equal to the length of the MPDTW path. The HMM state sequence emitting *K* patterns jointly is given by the CMPVA. Let $\mathbf{q} = q_{\phi(1)}, q_{\phi(2)}, \ldots, q_{\phi(T_p)}$ be the jointly decoded HMM state sequence, where $q_{\phi(t)}$ is the HMM state at time ϕ (t) which is jointly emitting the feature vectors $(O^1_{t_1}, O^2_{t_2}, \ldots, O^K_{t_K})$. We define a virtual pattern to be created as below:

$$\mathbf{V}^p_{\phi(1):\phi(T_p)} = f\{\mathbf{O}^1_{1:T_1}, \mathbf{O}^2_{1:T_2}, \ldots, \mathbf{O}^K_{1:T_K}; \phi(1) : \phi(T_p)\} \tag{43}$$

where $\mathbf{V}^p_{\phi(1):\phi(T_p)}$ is the p^{th} virtual pattern of length T_p, $1 \leq p \leq E$; $f(.)$ is a function which maps the *K* patterns to one virtual pattern. $\mathbf{V}^p_{\phi(1):\phi(T_p)} = (V^p_{\phi(1)}, V^p_{\phi(2)}, \ldots, V^p_{\phi(T_p)})$ and $V^p_{\phi(t)}$ is the feature vector of the p^{th} virtual pattern at time $\phi(t)$. Each feature vector in the virtual pattern

is defined as a weighted combination of the feature vectors from the K patterns (of the subset of training data), determined through the K tuple of the MPDTW path, i.e.

$$V^p_{\phi(t)} = \sum_{i=1}^{K} w_i(\phi(t)).O^i_{\phi_i(t)} \tag{44}$$

where $\phi(t) = (t_1, \ldots, t_K) = (\phi_1(t), \phi_2(t), \ldots, \phi_K(t))$, $\phi_i(t) = t_i$, and $w_i(\phi(t))$ is the weight for $O^i_{\phi_i(t)}$. $O^i_{\phi_i(t)}$ is the feature vector of the i^{th} pattern which lies on the time frame t_i ($O^i_{\phi_i(t)}$ is same as $O^i_{t_i}$ defined before). $w_i(\phi(t))$ is itself defined as:

$$w_i(\phi(t)) = \frac{b_{q_{\phi(t)}}(O^i_{\phi_i(t)})}{\sum_{i=1}^{K} b_{q_{\phi(t)}}(O^i_{\phi_i(t)})} \tag{45}$$

where $b_{q_{\phi(t)}}(O^i_{\phi_i(t)})$ is the likelihood of feature vector $O^i_{\phi_i(t)}$ emitted from state $q_{\phi(t)}$ of the current HMM λ. The above weighting factor is similar to the geometric weighting proposed in DC-wtd, but used differently in equation 44.

Similarly, we consider the next K subset of training patterns from the database and create another virtual pattern, and so on till E virtual patterns are created. All the E virtual patterns together are used to train the HMMparameters, using the Baum-Welch algorithm. After Baum-Welch convergence, the updated model is used iteratively for the re-creation of the virtual patterns as shown in Fig. 15. For each SHT iteration, a new set of virtual training patterns are created because the weights in equation 45 get modified because of the updated HMM parameters. However, the warping path is $\phi(t)$ for each virtual pattern is not a function of HMM parameters and hence does not vary with SHT iterations. We define a distortion measure to stop the SHT iterations. The number of features in the p^{th} virtual pattern path is fixed by the MPDTWpath and does not vary with iterations. Therefore, we can define convergence of the virtual patterns themselves as a measure of convergence. The change in the virtual pattern vector $V^p_{\phi(t)}$ of the p^{th} virtual pattern, $d(V^{p,m}_{\phi(t)}, V^{p,m-1}_{\phi(t)})$, is defined as the Euclidean distance between $V^p_{\phi(t)}$ at iteration number m and $V^p_{\phi(t)}$ at iteration number $m-1$. The total change at iteration m for the p^{th} virtual pattern sequence is

$$D^m(\mathbf{V}^p_{\phi(1):\phi(T_p)}) = \frac{1}{T_p}\sum_{t=1}^{T_p} d(V^{p,m}_{\phi(t)}, V^{p,m-1}_{\phi(t)}) \tag{46}$$

where $p = 1, 2, \ldots, E$. The average distortion D^m_{avg} for all the E virtual patterns at the m^{th} iteration is calculated and if it is below a threshold the SHT iterations are stopped.

The virtual pattern has the property that it is "cleaner" (less distorted) compared to the original patterns. Let us take an example when one (or all) of the original training patterns have been distorted by burst noise. The virtual pattern is created by giving less weighting to the unreliable portions of the original data through DC-wtd and the weight parameters $w_i(\phi(t))$ of equation 45. When most of the training patterns are not outliers, the initial HMM is relatively noise free. So $b_{q_{\phi(t)}}(O^i_{\phi_i(t)})$ of equation 45 can be expected to be higher for reliable values of $O^i_{\phi_i(t)}$. Hence, $w_i(\phi(t))$ has a higher value for reliable feature vectors and lower for the unreliable ones. With each SHT iteration, the virtual patterns become more

and more noise free leading to a better HMM. In the standard HMM training, the model converges optimally to the data. In the SHT, the model and the data are both converging. Since the data is moving towards what is more likely, it is possible that the variance parameter of the Gaussian mixture model (GMM) in each HMM state gets reduced after each iteration. This could deteriorate the generalizability of HMM and hence its speech recognition performance for the unseen data, as the new HMMs might not be able to capture the variability in the test patterns. So we have chosen to clamp the variance after the initial HMM training and allow only the rest of the HMM parameters to adapt. Also we have considered some variants to the formulation of the virtual pattern given in equation 44. Through the experiments, we found that there may be significant variation of the weight parameter $w_i(\phi(t))$ for each $\phi(t)$, and also across iterations. Therefore we propose below two methods of smoothing the $w_i(\phi(t))$ variation, which leads to better convergence of HMM parameters.

A weighted averaging in time for the weights $w_i(\phi(t))$s is done by placing a window in time as shown below:

$$V^p_{\phi(t)} = \sum_{i=1}^{K} \left[\sum_{j=-P}^{P} l_i(\phi(t+j)).w_i(\phi(t+j)) \right] .O^i_{\phi_i(t)} \tag{47}$$

where $2P + 1$ is the length of the window placed over time $\phi(t - P)$ to $\phi(t + P)$; $l_i(\phi(t + j))$ is the weighting given to the weight $w_i(\phi(t + j))$ at time $\phi(t + j)$, such that $\sum_{j=-P}^{P} l_i(\phi(t+j)) = 1$. Smoothing of the weights $w_i(\phi(t))$ allows the reduction of some sudden peaks and also uses the knowledge of the neighboring vectors. This improved ASR accuracy. Weighted averaging can also be done for the weights over successive iterations:

$$V^p_{\phi(t)} = \sum_{i=1}^{K} \left[\sum_{j=0}^{P} l_i^{m-j}(\phi(t)).w_i^{m-j}(\phi(t)) \right] .O^i_{\phi_i(t)} \tag{48}$$

where m is the iteration number, and $P + 1$ is the window length over iterations for the weights. $w_i^m(\phi(t))$ is the value of weight $w_i(\phi(t))$ at iteration m. $l_i^{m-j}(\phi(t))$ is the weighting given to the weight $w_i^{m-j}(\phi(t))$ at iteration $m - j$, such that $\sum_{j=0}^{P} l_i^{m-j} = 1$.

5. Experimental evaluation

5.1 MPDTW experiments

We carried out the experiments (based on the formulation in section 2) using the IISc-BPL database[1] which comprises a 75 word vocabulary, and 36 female and 34 male adult

[1] IISc-BPL database is an Indian accented English database used for Voice Dialer application. This database consists of English isolated words, English TIMIT sentences, Native language (different for different speakers) sentences, spoken by 36 female and 34 male adult speakers recorded in a laboratory environment using 5 different recording channels: PSTN-telephone (8 kHz sampling), Cordless local phone (16 kHz sampling), Direct microphone (16 kHz sampling), Ericsson (GSM) mobile phone (8 kHz sampling), Reverberant room telephone (Sony) (8 kHz sampling).

speakers, with three repetitions for each word by the same speaker, digitized at 8 kHz sampling rate. The vocabulary consists of a good number of phonetically confusing words used in Voice Dialer application. MFCCs, Δ MFCCs, and Δ² MFCCs is used without their energy components (36 dimensions) as feature vector. The experiment is carried out for speaker dependent IWR for 20 speakers and 75 word vocabulary.

The slope weighting function $m(t)$ is set equal to 1. The global normalization factor M_ϕ (equation 8) that we used is $M_\phi = T_1 + T_2 + \ldots + T_K$. (In this section 5.1, K stands for the sum of the number of test and template patterns.) Through the experiments we found that normalization using only the sum of the total frames of reference patterns in each class gives worse recognition than the normalization we used. In an experiment where each speaker utters 2 test patterns of the same class (of lengths T_1 and T_2 frames) and 1 reference pattern (of length T_3 frames) per class per speaker for 20 speakers, the percentage ASR accuracy using $M_\phi = T_1 + T_2 + T_3$ is 96.47%. If $M_\phi = T_3$ then the percentage accuracy reduces to 89.07%.

Table 1 summarizes the results based on the formulation of section 2. In the table, the experiment-1 DTW-1 test-1 templ, corresponds to standard DTW algorithm applied when there is 1 test pattern spoken by each speaker for each word and it's distortion is compared with the reference patterns (1 reference pattern per speaker per word for 20 speakers) of each word in the vocabulary. In experiment-2 DTW-2 test-1 templ each speaker utters 2 patterns of a word. Each one of them is compared separately with the reference patterns (1 template per speaker per word). In experiment-3 DTW-2 test-1 templ (minimum of two), the minimum of the two distortions of the two test patterns (of the same word by a speaker) with the reference patterns, is considered to calculate recognition accuracy. In experiment-4 MPDTW-2 test-1 templ, each speaker utters 2 test patterns. The MPDTW algorithm is applied on the 2 test patterns and 1 reference pattern at a time (1 reference pattern per speaker per word) to find the distortion between them. In experiment-5 DTW-1 test-2 templ, 1 test pattern, each speaker speaks 1 test pattern and 2 reference patterns. The test pattern is now compared with the reference patterns (2 reference patterns per speaker per word for 20 speakers). In experiment-6 MPDTW-1 test-2 templ, the MPDTW algorithm is applied on 1 test pattern and 2 reference patterns (2 reference patterns per speaker per word) and then IWR is done. In this experiment K is equal to the sum of the number of test and template patterns.

Experiment	Clean	Noisy
1. DTW-1 test-1 templ	94.67	91.13
2. DTW-2 test-1 templ	95.17	92.47
3. DTW-2 test-1 templ (minimum of two)	95.40	89.73
4. MPDTW-2 test-1 templ	96.47	94.47
5. DTW-1 test-2 templ	96.67	94.07
6. MPDTW-1 test-2 templ	90.80	85.60

Table 1. Comparison of ASR percentage accuracy for clean and noisy test pattern. For noisy speech, burst noise is added for 10% of test pattern frames at -5 dB SNR (local). Reference patterns (templ) are always clean. IWR is done for 20 speakers.

We see from the results that when there are 2 test patterns uttered by a speaker and 1 reference pattern (case 2) and the MPDTW algorithm (for K = 3) is used, the speech recognition word error rate reduced by 33.78% for clean speech and 37.66% for noisy test speech (10% burst noise added randomly with uniform distribution at -5 dB SNR (local) in both the test patterns), compared to the DTW algorithm (same as MPDTW algorithm when

K = 2) for only 1 test pattern. Even when using the minimum distortion among two test patterns (experiment-2 DTW-2 test-1 templ (minimum of two)), we see that the MPDTW algorithm works better. However, when we use only 1 test pattern and 2 reference patterns and when the MPDTWalgorithm (for K = 3) is used, the percentage accuracy reduces as predicted in section 2.2. Hence we see that use of multiple test repetitions of a word can significantly improve the ASR accuracy whereas using multiple reference patterns can reduces the performance.

5.2 MPJL experiments

Based on the formulations in section 3, we conducted experiments - A1M, A1P, A2, A3 for speaker independent IWR along with the base line system of standard Forward Algorithm (FA) or Viterbi Algorithm (VA) for a single pattern, for the cases of both clean and noisy speech. Since the normal FA/VA uses one pattern to make a recognition decision and the proposed algorithms use K patterns to make a decision, the comparison of results may not be fair. For a fairer comparison we formulated the experiment A1M, which also uses K patterns using the standard FA/VA and the best (max) likelihood of the K patterns is chosen. Experiment A1P uses the product of the K likelihoods of the K patterns. So we compare the new algorithms (experiment A2 and A3) with the experiments A1M, A1P also. (In this section 5.2, K stands for the total number of test patterns of a word spoken by a speaker.)

The experiment A1M is as described. Given $\mathbf{O}^1_{1:T_1}$, $\mathbf{O}^2_{1:T_2}$,. . ., $\mathbf{O}^K_{1:T_K}$ as the individual patterns belonging to the same class, we can obtain the joint likelihood score as $\theta_j = \max_{1\leq i\leq K} P(\mathbf{O}^i_{1:T_i}/\lambda_j)$, where λ_j are the clean word models and the FA/VA is used to calculate $P(\mathbf{O}^i_{1:T_i}/\lambda_j)$. We select the pattern as j^* = argmax$_j$ θ_j . We are actually doing a voting, where the pattern which has the highest likelihood is chosen. Experiment A1P when the joint likelihood score $\theta_j = \prod_{1\leq i\leq K} P(\mathbf{O}^i_{1:T_i}/\lambda_j)$. For the experiments, we have restricted to two or three patterns per test speaker. When K = 2, for each word of a test speaker, A1M and A1P are done for pattern 1 and pattern 2, pattern 2 and 3, pattern 3 and 1. When K = 3, all the 3 patterns are considered. Experiment A2 is the MPDTW algorithm followed by CMP?A-2. Experiment A3 is the MPDTW algorithm followed by CMP?A-1. In all joint recognition experiments, we have restricted to two or three pattern joint recognition and compared the performance with respect to single pattern recognition. When K = 2, for each word of a test speaker, pattern 1 is jointly recognized with pattern 2, pattern 2 with 3, pattern 3 with 1. When K = 3 all the three patterns are jointly recognized. Please note that in the noisy case, all the three patterns are noisy. As the number of test patterns K = 2, for the new experiments we chose the Local Continuity Constraints for MPDTW as (1,0) or (0,1) or (1,1) and the slope weighting function $m(t)$ = 1. Similar extensions are done for K = 3.

The IISc-BPL database was used for experimentation. Left to Right HMMs are trained for clean speech using the Segmental K Means (SKM) algorithm [Rabiner et al., 1986, Juang & Rabiner, 1990]. 25 male and 25 female speakers are used for training, with three repetitions of each word by each speaker. We tested the algorithm for 20 unseen speakers (11 female and 9 male) in both clean and noisy cases. Test words are three patterns for each word by each speaker, at each SNR.

We first run the experiment for speech affected by burst noise. Burst noise was added to some percentage of the frames of each word at -5 dB, 0 dB, 5 dB SNRs (local) to all the three patterns. (The remaining frames are clean; the range of -5 dB to +5 dB indicates severe to

mild degradation of the noise affected frames.) The burst noise can occur randomly anywhere in the spoken word with uniform probability distribution. MFCCs, Δ MFCC, and Δ^2 MFCC is used without their energy components (36 dimensions). Energy components are neglected and Cepstral Mean Subtraction was done. Variable number of states are used for each word model; i.e. proportional to the average duration of the training patterns, for each second of speech, 8 HMM states were assigned, with 3 Gaussian mixtures per state.
We experimented for various values of the threshold γ in DC-thr1 and found that there is indeed an optimum value of γ where the performance is maximum. When K = 2, for the noisy patterns with burst noise added to 10% of the frames at -5 dB SNR, γ = 0.5 is found to be optimum. It is also clear that γ < 0 provides closer to optimum performance than $\gamma = \infty$, indicating that the max operation is more robust than the product operation. Using DC-wtd was shown to have similar results to using DC-thr1 with optimum threshold.

Algorithm	ASRA(Clean)	ASRA(Clean)	-5 dB ASRA	-5 dB ASRA	0 dB ASRA	0 dB ASRA	5 dB ASRA	5 dB ASRA
FA	89.61	89.61	56.87	56.87	61.24	61.24	67.13	67.13
A1M, K=2	89.81	89.81	60.07	60.07	64.00	64.00	69.40	69.40
A1P, K=2	92.29	92.29	64.44	64.44	68.84	68.84	74.44	74.44
DC-thr1, $\gamma = \infty$, K=2	91.87, A2	91.80, A3	61.31, A2	61.53, A3	66.07, A2	66.11, A3	72.42, A2	72.22, A3
DC-thr1, $\gamma = 2$, K=2	91.87, A2	91.78, A3	73.91, A2	73.18, A3	77.02, A2	76.02, A3	80.58, A2	80.11, A3
DC-thr1, $\gamma = 1$, K=2	91.80, A2	92.13, A3	78.09, A2	76.09, A3	80.82, A2	78.76, A3	83.80, A2	82.24, A3
DC-thr1, $\gamma = 0.75$, K=2	91.73, A2	92.18, A3	79.13, A2	76.98, A3	81.69, A2	79.76, A3	84.51, A2	83.05, A3
DC-thr1, $\gamma = 0.5$, K=2	91.49, A2	92.02, A3	79.47, A2	77.18, A3	82.00, A2	79.84, A3	84.76, A2	83.44, A3
DC-thr1, $\gamma = 0.25$, K=2	91.49, A2	91.98, A3	79.44, A2	77.09, A3	82.00, A2	79.91, A3	84.73, A2	83.38, A3
DC-thr1, $\gamma < 0$, K=2	91.49, A2	91.98, A3	79.44, A2	77.09, A3	82.00, A2	79.91, A3	84.73, A2	83.38, A3
DC-wtd, K=2	91.38, A2	92.02, A3	79.35, A2	77.11, A3	82.05, A2	79.96, A3	84.78, A2	83.38, A3
DC-wtd, K=3	92.73, A2	93.13, A3	88.07, A2	86.73, A3	89.20, A2	87.73, A3	90.07, A2	89.20, A3

Table 2. Comparison of ASR percentage accuracy (ASRA) for clean and noisy speech (10% burst noise) for FA, A1M, A1P, A2, and A3. FA - Forward Algorithm, Experiment A1M, K = 2 - best (max of likelihoods) of two patterns using FA, Experiment A1P, K = 2 - product of the likelihoods (using FA) of two patterns, Experiment A2 - MPDTW algorithm + CMPFA-2, Experiment A3 – MPDTW algorithm + CMPFA-1. K is the number of test patterns used.

The results for clean and noisy speech for FA and CMPFA is given in Table 2. We have not shown the results of CMPBA as it is similar to CMPFA. In the table, ASRA (Clean) stands for ASR accuracy for clean speech. In the tables, for experiment A2, in the ASRA column, the ASR percentage accuracy is written. Note that DC-thr1 is equivalent to DC-avg when $\gamma = \infty$ (product operation) and it is equivalent to DC-max when γ < 0 (max operation). Also, DC-thr1 is same as DC-thr2 when K = 2. When K = 2, two patterns are recognized at a time, while K = 3 stands for 3 patterns being recognized at a time. In the table, -5 dB ASRA stands for ASR accuracy for noisy speech which has 10% burst noise at SNR -5 dB. It can be seen that the baseline performance of FA for clean speech is close to 90%. For example, for noisy case at -5 dB SNR, for speech with 10% burst noise, it decreases to ≈ 57%. Interestingly, the experiment A1M (for K = 2 patterns) provides a mild improvement of 0.2% and 3.2% for clean and noisy speech (at -5 dB SNR burst noise) respectively, over the FA benchmark. This shows that use of multiple patterns is indeed beneficial, but just maximization of likelihoods is weak. Experiment A1P (for K = 2 patterns) works better than A1M clearly indicating that taking the product of the two likelihoods is better than taking their max. The proposed new algorithms (experiment A2 and A3) for joint recognition provides dramatic improvement for the noisy case, w.r.t. the FA performance. For example at -5 dB SNR 10% burst noise, for K = 2 patterns, the proposed algorithms (experiments A2 and A3) using DC-thr1 at

threshold γ = 0.5, gave an improvement of 22.60% speech recognition accuracy (using CMPFA-2) and 20.31% speech recognition accuracy (using CMPFA-1) compared to FA performance. For K = 3 patterns, the recognition accuracy increases by 31.20% (using CMPFA-2) and 29.86% (using CMPFA-1). So there was almost a 10% improvement in speech recognition accuracy from K = 2 to K = 3. We also see that as the SNR improves, the gap in the speech recognition accuracy between performance of DC-thr1 at threshold $\gamma = \infty$ and $\gamma <$ 0 reduces. In fact as SNR approaches to that of clean speech, $\gamma = \infty$ is better than $\gamma < 0$.

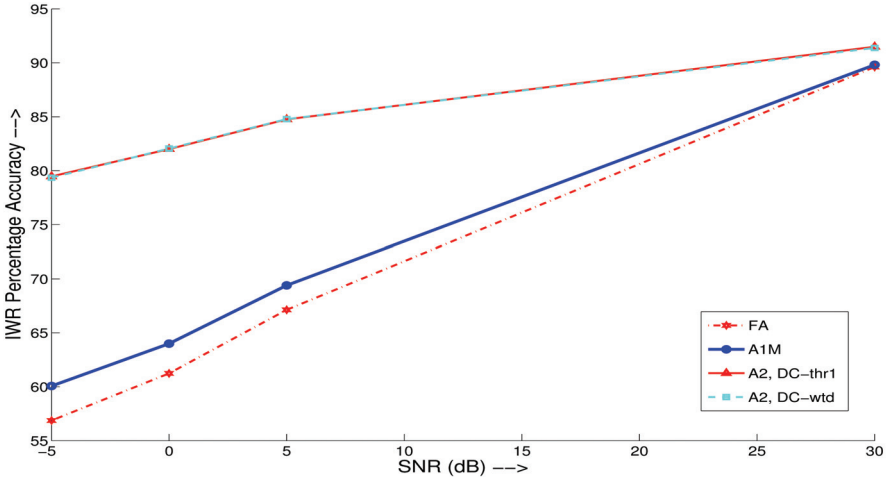


Fig. 16. Percentage accuracies for experiments FA, A1M, A2 for different levels of burst noises. FA - Forward Algorithm, A1M - best of two patterns using FA, A2 - MPDTW algorithm + CMPFA-2 algorithm. Results for A2 using DC-thr1 (at threshold γ = 0.5) and DC-wtd are shown.

For clean speech, the speech recognition accuracy when K = 2, improved by 2.26% using CMPFA-2 and 2.57% using CMPFA-1 (DC-thr1) over that of FA. This improvement could be because some mispronunciation of some words were be taken care of. It is also better than experiment A1M. We also see CMPFA-1 is better than CMPFA-2 for clean speech. However, experiment A1P works the best for clean speech (when K = 2). This could be because in the proposed methods, the K patterns are forced to traverse through the same state sequence. Doing individual recognition on the two patterns and then multiplying the likelihoods has no such restrictions. And in clean speech there is no need for selectively weighting the feature vectors. So experiment A1P works slightly better than experiments A2 and A3 for clean speech.

We also see that as per our analysis in section 3.4 and the results shown in Table 2, using DC-thr1 for CMPFA-1 algorithm (experiment A3) for clean speech at lower thresholds gives better recognition results than using it for CMPFA-2 algorithm (experiment A2). At higher thresholds CMPFA-2 algorithm is better. For noisy speech (speech with burst noise) it is better to use CMPFA-2 than CMPFA-1.

From the table, it is seen that as K increases 2 to 3, there is an improvement in recognition accuracy. We also see that for K = 3, the performance does vary much, whether the burst noise is at -5 dB SNR or +5 dB SNR. This is because the noise corrupted regions of speech is

almost completely neglected during recognition (whether noise is at -5 dB SNR or +5 dB SNR) and the clean portion of the other patterns are given a higher weight.
A graph showing variation of IWR percentage accuracy versus the burst noise at some SNR is shown in Fig. 16. In this figure the experiment A2 using DC-thr1 is plotted at threshold γ = 0.5, where DC-thr1 works very well for speech with burst noise. We see that using DC-wtd gives us optimum or near optimum values. The performance of DC-wtd is equal to to the optimum performance of DC-thr1, as we predicted in section 3.4. And the advantage of DC-wtd is that we don't need any threshold. We also see that as per our analysis in section 3.4 and the results shown in Table 2, using DC-thr1 for CMPFA-1 algorithm (experiment A3) for clean speech at lower thresholds gives better recognition results than using it for CMPFA-2 algorithm (experiment A2). At higher thresholds CMPFA-2 algorithm is better. For noisy speech (speech with burst noise) it is better to use CMPFA-2 than CMPFA-1.
The results for clean and noisy speech for VA and CMPVA is given in Table 3. It is seen that CMPVA-1 performs similarly to CMPFA-1 and CMPVA-2 performs similar to CMPFA-2.

Algorithm	ASRA (Clean)	ASRA (Clean)	-5 dB ASRA	-5 dB ASRA	0 dB ASRA	0 dB ASRA	5 dB ASRA	5 dB ASRA
VA	89.70	89.70	57.13	57.13	61.49	61.49	67.38	67.38
A1M, K=2	89.87	89.87	60.33	60.33	64.29	64.29	69.49	69.49
A1P, K=2	92.38	92.38	65.00	65.00	69.36	69.36	74.84	74.84
DC-thr1, $\gamma = \infty$, K=2	91.91, A2	91.78, A3	61.44, A2	61.73, A3	66.36, A2	66.18, A3	72.62, A2	72.47, A3
DC-thr1, $\gamma = 2$, K=2	91.87, A2	91.76, A3	74.11, A2	73.20, A3	77.18, A2	76.18, A3	80.69, A2	80.09, A3
DC-thr1, $\gamma = 1$, K=2	91.78, A2	92.20, A3	78.11, A2	76.36, A3	80.84, A2	78.73, A3	83.80, A2	82.27, A3
DC-thr1, $\gamma = 0.75$, K=2	91.76, A2	92.13, A3	79.11, A2	77.04, A3	81.76, A2	79.84, A3	84.60, A2	83.07, A3
DC-thr1, $\gamma = 0.5$, K=2	91.51, A2	92.02, A3	79.47, A2	77.24, A3	82.13, A2	79.96, A3	84.80, A2	83.42, A3
DC-thr1, $\gamma = 0.25$, K=2	91.53, A2	92.00, A3	79.47, A2	77.18, A3	82.16, A2	80.02, A3	84.80, A2	83.40, A3
DC-thr1, $\gamma < 0$, K=2	91.53, A2	92.00, A3	79.47, A2	77.18, A3	82.16, A2	80.02, A3	84.80, A2	83.40, A3
DC-wtd, K=2	91.44, A2	91.98, A3	79.42, A2	77.22, A3	82.07, A2	79.98, A3	84.76, A2	83.42, A3
DC-wtd, K=3	92.73, A2	93.07, A3	88.20, A2	86.80, A3	89.13, A2	87.67, A3	90.07, A2	89.27, A3

Table 3. Comparison of ASR percentage accuracy (ASRA) for clean and noisy speech (10% noise) for VA, A1M, A1P, A2, and A3. VA - Viterbi Algorithm, Experiment A1M, *K* = 2 - best of two patterns using VA, Experiment A1P, *K* = 2 - product of the likelihoods (using VA) of two patterns, Experiment A2 - MPDTW algorithm + CMPVA-2, Experiment A3 - MPDTW algorithm + CMPVA-1.

Percentage noise	-5dB FA	-5 dB CMPFA-1	0 dB FA	0 dB CMPFA-1	5 dB FA	5 dB CMPFA-1
10%, burst	56.87	77.11	61.24	79.96	67.13	83.38
20%, burst	44.55	64.93	49.47	69.38	56.71	75.36
30%, burst	33.67	50.98	38.67	56.89	46.49	64.02
50%, burst	11.38	11.91	16.09	16.42	25.20	29.09
AWGN	3.62	2.89	8.89	7.47	22.84	19.44

Table 4. Comparison of ASR percentage accuracy (ASRA) for noisy speech (burst) for FA and CMPFA-1 (for *K* = 2 patterns). DC-wtd is used. Different percentages of burst noise (10%, 20%, 30%, 50%) noises added to the speech pattern form the test cases. Additive White Gaussian Noise (AWGN) is added to 100% of the speech pattern is also a test case.

So far we have shown the results of burst noise added to only 10% of the speech patterns. Now different percentage of burst noise is added randomly to the patterns. The results for FA and CMPFA-1 (for *K* = 2 patterns) is shown in Table 4. DC-wtd is used. We see that speech is affected with 10%, 20%, 30% burst noise, the ASR performance using CMPFA-1 (for *K* = 2 patterns) is much better than using just using FA. However when noise is added to 50% of the frames, there is only a marginal increase in performance. This is because many

regions of both the patterns will be noisy and the CMPFA-1 does not have a clean portion of speech to give a higher weighting. When 100% of the speech are affected by additive white Gaussian noise (AWGN), then just using FA is better than CMPFA-1. Similar results are given in Table 5 for VA and CMPVA-1.

Percentage noise	-5dB VA	-5 dB CMPVA-1	0 dB VA	0 dB CMPVA-1	5 dB VA	5 dB CMPVA-1
10%, burst	57.13	77.22	61.49	79.98	67.38	83.42
20%, burst	44.67	65.02	49.64	69.56	57.27	75.40
30%, burst	34.33	51.20	39.31	56.91	47.00	64.18
50%, burst	11.89	12.33	16.89	16.89	26.29	29.78
AWGN	3.73	3.20	8.96	7.44	22.89	19.40

Table 5. Comparison of ASR percentage accuracy for noisy speech (burst) for VA and CMPVA-1 (for K = 2 patterns). DC-wtd is used. Different percentages of burst noise (10%, 20%, 30%, 50%) noises added to the speech pattern form the test cases. Additive White Gaussian Noise (AWGN) is added to 100% of the speech pattern is also a test case.

Algorithm	5 dB	10 dB
FA (babble noise)	44.18	59.64
CMPFA-1 (babble noise), K=2	47.80	64.36
CMPFA-1 (babble noise), K=3	47.40	65.93
VA (babble noise), K=2	44.27	59.73
CMPVA-1 (babble noise), K=2	47.84	64.33
CMPVA-1 (babble noise), K=3	47.40	65.87
FA (machine gun noise)	66.53	71.36
CMPFA-1 (machine gun noise), K=2	76.76	81.33
CMPFA-1 (machine gun noise), K=3	81.53	84.20
VA (machine gun noise)	66.71	71.47
CMPVA-1 (machine gun noise), K=2	76.89	81.16
CMPVA-1 (machine gun noise), K=3	81.60	84.13

Table 6. Comparison of ASR percentage accuracy for noisy speech (babble noise or machine gun noise) for FA, VA, CMPFA-1 and CMPVA-1 (for K = 2 patterns). SNR of the noisy speech is 5 dB or 10 dB. DC-wtd is used.

Now we compare the results of the proposed algorithms with other kinds of transient noises like machine gun noise and babble noise. Machine gun and babble noise from NOISEX 92 was added to the entire speech pattern at 5 dB or 10 dB SNR. The results are given in Table 6. The HMMs are trained using clean speech in the same way as done before. We see that for speech with babble noise at 10 dB SNR, the percentage accuracy using FA is 59.64%. It increases to 64.36% when CMPFA-1 (when K = 2 patterns) is used, which is rise of nearly 5%. When K = 3, the accuracy is further improved. When machine gun noise at 10 dB SNR is used the FA gives an accuracy of 71.36%, while the CMPFA-1 when K = 2 patterns gives an accuracy of 81.33% and when K = 3 the accuracy is 84.20%. We see an increase of nearly 10% when K = 2 and 13% when K = 3. We see from the results that the more transient or bursty the noise is, the better the proposed algorithms work. Since machine gun noise has a more transient (bursty) nature compared to babble noise it works better. In the case of machine gun noise, if there is some portion of speech affected by noise in $O_{1:T_1}^{1}$, there could be a

corresponding clean speech portion in $O^2_{1:T_2}$. This clean portion will be given a higher weight by the proposed algorithms during recognition. However this is not possible if the entire speech is affected by white Gaussian noise or babble noise. When both the patterns are affected by similar noise at the same portion, then there is no relatively clean portion of speech for the proposed algorithms to choose. Hence they work worse. We see that as K increases from 2 to 3 patterns, the ASR accuracy improves significantly for machine gun noise, while for babble noise, the effect is small (in fact at 5 dB the ASR accuracy slightly reduces from K = 2 to K = 3).

5.3 Selective HMM training experiment

Here again, we carried out speaker independent IWR experiments (based on the formulation in section 4), using the IISc-BPL database. Left to Right HMMs are trained using the Baum-Welch training algorithm. MFCC, Δ MFCC, and Δ² MFCC are used without the energy components (total 36 dimension vector). Cepstral mean subtraction is done. 25 male and 25 female speakers are used for training, with three repetitions of each word by each speaker. So the total number of training patterns (D) for each word is 150. 5 HMM states per word is used with 3 Gaussian mixtures per state. Both clean speech and speech with a burst noise of 10% at -5 dB SNR (local) was used for HMM training. The burst noise can occur randomly anywhere in the spoken word with uniform probability distribution. Note that in the noisy case all the training patterns have burst noise in them. We used CMPVA-2 for our experiments. (In this section 5.3, K stands for the number of training patterns used to create one virtual pattern.)

We first consider an example to gain insight into the working of SHT. Four clean patterns of the word "Hello", O1, O2, O3, O4, by a speaker are considered. Noise at -5 dB SNR is added to the first 10% of the frames in pattern O3. For this example, the initial HMM was found using the Baum-Welch algorithm using only these 4 patterns. One virtual pattern is created (using equation 43) from the 4 patterns (K = 4). Fig. 17 shows the decrease in the distortion measure with each SHT iteration showing clear convergence. However, for all the patterns this decrease is not always monotonic in nature and there may be small fluctuations at higher iterations.

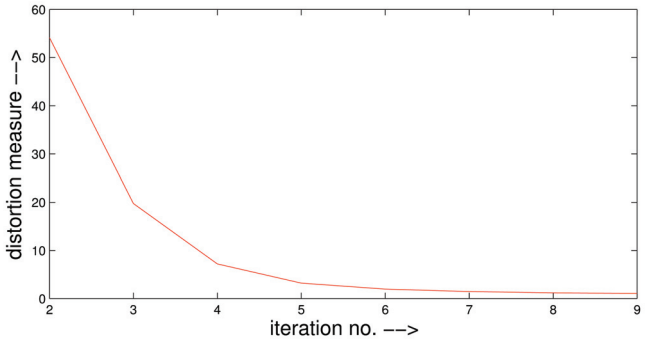


Fig. 17. Distortion with each iteration.

Fig. 18 shows the weights $w_3(\phi(t))$ given to each frame of the virtual pattern for pattern O3 for the first and ninth iteration. We can see the amount of contribution of O3, i.e., to the virtual pattern: i.e., the all the initial frames are given a low weight as O3 is noisy in this

region. And this weighting decreases with every iteration as the virtual pattern converges. Similarly Fig. 19 gives the weights of O2. We see that the initial few frames do not have less weighting, contrasting with O3.

Fig. 20 shows the difference in likelihood of O1, O2, O4 with O3 given the HMMλ are shown. $P(O1/\lambda)-P(O3/\lambda)$, $P(O2/\lambda)- P(O3/\lambda)$, $P(O4/\lambda) - P(O3/\lambda)$ are the three curves shown in that figure. These probabilities are computed using the Forward algorithm. In Fig. 20, at iteration 0, the HMMs is the Baum-Welch algorithm run on the original training data. $P(O2/\lambda)$ and $P(O4/\lambda)$ are greater than $P(O3/\lambda)$. After each SHT iteration the HMM is updated and the differences of $P(O2/\lambda) -P(O3/\lambda)$ and $P(O4/\lambda) - P(O3/\lambda)$ increases. This happens because the HMM is updated by giving less weightage to the noisy portion of O3. We also see that although some portion of O3 is noisy, $P(O1/\lambda)$ is less than $P(O3/\lambda)$. This is because the HMM is trained using only 4 patterns out of which one HMM pattern (O3) is partially noisy. So the initial HMM using the Baum-Welch algorithm is not very good. We see that after the iterations the difference $P(O1/\lambda) - P(O3/\lambda)$ reduces. This indicated that after each iteration the HMM parameters are updated such that the unreliable portions of O3 is getting a lesser weight.

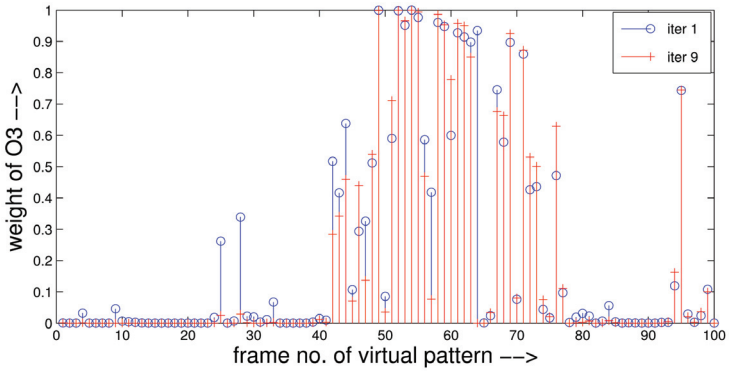


Fig. 18. Weight of O3 for creating of virtual pattern at iterations numbers 1 and 9. Noise at -5 dB SNR is added to the first 10% of the frames in pattern O3.

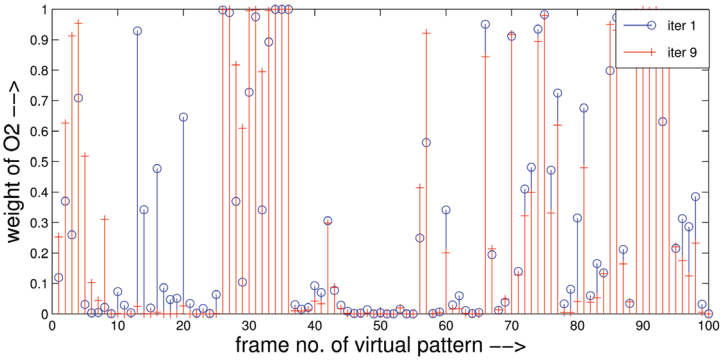


Fig. 19. Weight of O3 for creating of virtual pattern at iterations numbers 1 and 9. Noise at -5 dB SNR is added to the first 10% of the frames in pattern O3.

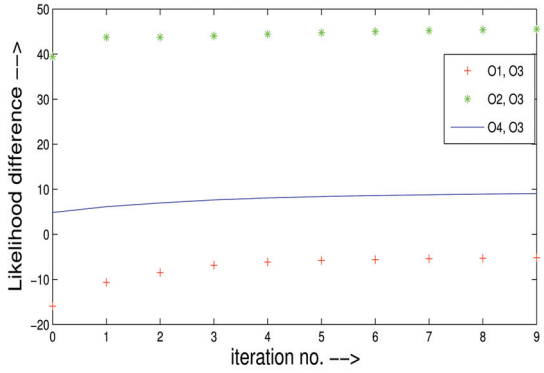


Fig. 20. Difference between Likelihood of patterns O1, O2, O4 with O3 given HMM λ.

While the above results are promising in terms of the functioning of the SHT algorithm, the level of improvement in HMM performance for test data could be limited, depending on the nature and size of the test data. Using the small size pilot experiment, the HMM performance is tested using Forward algorithm (FA) for 20 unseen speakers (11 female and 9 male) using clean speech. There are 3 test patterns per speaker. The experimental setup used for training was done as mentioned in the first paragraph of this sub-section. Each of the 150 training patterns (3 training patterns per speaker, for 50 speakers) are affected by 10% burst noise at -5 dB SNR. The covariance matrix for the Gaussian mixtures for each HMM state was fixed to that of the initial Baum-Welch algorithm run on the original training data. The number of patterns to create one virtual pattern (K) is 3. The 3 patterns spoken by the same speaker are considered for creating one virtual pattern as the MPDTW path alignment may be better for the same speaker. So we have a total of 50 virtual patterns ($E = D/K$) per word since the total number of training patterns (D) per word is 150 (number of speakers is 50). The virtual patterns are created using equation 44. We used CMPVA-2 for the experiments. When Baum-Welch (BW) algorithm is used to train speech with 10% burst noise at -5 dB then for the clean test data the ASR percentage accuracy is 88.36%. It increases to 88.76% using the new SHT (using equation 44 to create virtual patterns) when the covariance matrix of each HMM state is kept constant (*covar − const*). Let this experiment be called $SHT-2$. If the covariance matrix is allowed to adapt (*covar − adapt*), the covariance decreases (determinant value) after each iteration as the virtual patterns are converging to what is more likely and this may reduce the ability of the recognizer to capture the variabilities of the test patterns. (Let such an experiment be called experiment $SHT-1$.) When the covariance matrix is not kept constant the percentage accuracy reduces to 86.04%. So we keep the covariance constant.

We now experiment by keeping averaging the weights over time and iterations (see equations 47, 48). When averaging the weights over time (equation 47) keeping the covariance matrix constant, the percentage accuracy increases to 88.84%. Let this be experiment be called $SHT-3$. In equation 47, we set $P = 1$, $l_i(\phi(t)) = 0.5$, $l_i(\phi(t-1)) = l_i(\phi(t+1)) = 0.25$. This shows that smoothing the weights $w_i(\phi(t))$'s improves ASR accuracy. For averaging the weights over iterations (equation 48), $l_i^m(\phi(t)) = 0.5$, $l_i^{m-1}(\phi(t)) = 0.3$, $l_i^{m-2}(\phi(t)) = 0.2$. However, the ASR accuracy reduces to 73.38%. Let this experiment be called $SHT-4$.

Now we increase the number of virtual patterns used for training. In the database used, the number of patterns of a word spoken by a speaker is 3. Let the patterns be O1, O2, O3. We can create 1 virtual pattern using O1, O2, O3 (K = 3). We can also create 3 other virtual patterns using O1-O2, O2-O3, O3-O1 (K = 2 for each). Thus we have 4 virtual patterns created using training patterns O1, O2, O3. We do this for very speaker and every word. So the total number of virtual patterns per word is 200 (since the number of training patterns per word is 150). HMMs are trained on these virtual patterns. The covariance matrix is kept constant and equation 44 is used for calculating the virtual patterns. Let this be called experiment *SHT* − 5. The ASR accuracy using FA increases to 89.07%. The more number of virtual patterns we use to train the HMMs, the better the test data variability is captured. We see from the experiments that as the number of virtual patterns per word (using equation 44) increases from 50 to 200, the percentage accuracy also increases from 88.76% to 89.07%, clearly indicating that it helps using more virtual patterns as training data. However in this case, by averaging over time (called experiment *SHT* − 6), using equation 47, the accuracy remained at 89.07%. The results are summarized in Table 7. Thus it was shown that the word error rate decreased by about 6.1% using the proposed SHT training method over the baseline Baum-Welch method.

Experiment	Percentage ASR accuracy
BW	88.36
SHT-1-covar-adapt	86.04
SHT-2-covar-const	88.76
SHT-3-covar-const	88.84
SHT-4-covar-const	73.38
SHT-5-covar-const	89.07
SHT-6-covar-const	89.07

Table 7. Comparison of Percentage ASR percentage accuracy for different algorithms. The training patterns have 10% of their frames affected by burst noise at -5 dB SNR. Testing was done on clean speech using FA. BW - Baum-Welch algorithm. *SHT* −1 - SHT using 1 virtual patterns per word per speaker and adapting covariance. *SHT* − 2 - SHT using 1 virtual patterns per word per speaker and constant covariance. *SHT* − 3 - SHT using 1 virtual patterns per word per speaker and constant covariance; averaging of weights across time is done. *SHT* − 4 - SHT using 1 virtual patterns per word per speaker and constant covariance; averaging of weights across iterations is done. *SHT* − 5 - SHT using 4 virtual patterns per word per speaker and constant covariance. *SHT* − 6 – SHT using 4 virtual patterns per word per speaker and constant covariance; averaging of weights across time is done.

HMMs were also trained using clean speech. Testing is also done on clean speech on unseen speakers using the Forward algorithm. Using Baum-Welch algorithm, we get an ASR accuracy of 91.18%. Using the proposed SHT algorithm (experiment *SHT* − 5), we get an accuracy of 91.16%. When averaging over time is done (experiment *SHT* −6), the ASR accuracy remains at 91.16%. Thus we see that using the proposed SHT training method does not reduce the ASR performance for clean speech.

HMMs were also trained using training patterns corrupted with machine gun noise (from NOISEX 92 database) at 10 dB SNR. The experimental setup is same as used before. Totally there are 150 training patterns (3 per speaker for 50 speakers). Testing was done on unseen clean speech using the FA. Using normal Baum-Welch training the ASR accuracy was

85.56%. However it reduced to 85.20% using 200 virtual training virtual patterns (experiment *SHT* − 5). Using averaged weights over time (experiment *SHT* − 6), the percentage accuracy increased to 85.29%. However it is still lower compared to the Baum-Welch training algorithm. The performance could have been better if there were more virtual patterns created from the training data set of 150 patterns.

6. Conclusions

We have formulated new algorithms for joint evaluation of the likelihood of multiple speech patterns, using the standard HMM framework. This was possible through the judicious use of the basic DTW algorithm extended to multiple patterns. We also showed that this joint formulation is useful in selective training of HMMs, in the context of burst noise or mispronunciation among training patterns.

Although these algorithms are evaluated in the context of IWR under burst noise conditions, the formulation and algorithm can be useful in different contexts, such as connected word recognition (CWR) or continuous speech recognition (CSR). In spoken dialog systems, if the confidence level of the test speech is low, the system can ask the user to repeat the pattern. However, in the continuous speech recognition case, a user cannot be expected to repeat a sentence/s exactly. But still the proposed methods can be used. Here is one scenario. For booking a railway ticket, the user says, "I want a ticket from Bangalore to Aluva". The recognition system asks the user, "Could you please repeat from which station would you like to start?". The user repeats the word "Bangalore". So this word "Bangalore" can be jointly recognized with the word "Bangalore" from the first sentence to improve speech recognition performance.

One of the limitations of the new formulation is when the whole pattern is noisy, i.e., when the noise is continuous not bursty; the proposed algorithms don't work well. Also, for the present, we have not addressed the issue of computational complexity, which is high in the present implementations. Efficient variations of these algorithms have to be explained for real-time or large scale CSR applications.

Finally we conclude that jointly evaluating multiple speech patterns is very useful for speech training and recognition and it would greatly aid in solving the automatic speech recognition problem. We hope that our work will show a new direction of research in this area.

Appendix A1 - Proof for the recursive equation in CMPFA-1

We shall derive the recursive equation for CMPFA-1 (equation 21) using the example shown in Fig. 9. Consider two patterns $O^1_{1:T_1}$ and $O^2_{1:T_2}$. The MPDTW algorithm gives the time alignment (MPDTW path) between these two patterns (as shown in Fig. 9). We now fit a layer of HMM states on this MPDTW path.

$$\alpha_{\phi(t)}(j) = P(\mathbf{O}^1_{1:t_1}, \mathbf{O}^2_{1:t_2}, \ldots, \mathbf{O}^K_{1:t_K}, q_{\phi(t)} = j/\lambda) \tag{49}$$

where $q_{\phi(t)}$ is the HMM state at $\phi(t)$ and λ is the HMM model with state $j \in 1 : N$, where N is the total number of states in the HMM. In the given example $\phi(1)$ = (1, 1), $\phi(2)$ = (2, 2), $\phi(3)$ = (3, 2). Each state j can emit a variable number of feature vectors varying from 1 to K.

$$\begin{aligned} \alpha_{\phi(1)}(i) &= P(O_1^1, O_1^2, q_{\phi(1)} = i/\lambda) \\ &= P(O_1^1, O_1^2/q_{\phi(1)} = i, \lambda)\, P(q_{\phi(1)} = i/\lambda) \\ &= \pi_i\, b_i(O_1^1, O_1^2) \end{aligned} \tag{50}$$

where $b_i(O_{t_1}^1, O_{t_2}^2)$ is the probability of feature vectors $O_{t_1}^1$ and $O_{t_2}^2$ emitted given state i, $\pi_i = P(q_{\phi(1)} = i/\lambda)$ which is the state initial probability. It is assumed to be same as the state initial probability given by the HMM. This can be done because the value of $b_j(O_{t_1}^1, O_{t_2}^2)$ is normalized as shown in section 3.3, such that the probability of K (here $K = 2$) vectors being emitted by a state is comparable to the probability of a single vector being emitted by that state. So we are inherently recognizing one virtual pattern from K test patterns.

$$\begin{aligned} \alpha_{\phi(2)}(j) &= P(O_1^1, O_1^2, O_2^1, O_2^2, q_{\phi(2)} = j/\lambda) \\ &= \sum_{q_{\phi(1)}=i} P(O_1^1, O_1^2, O_2^1, O_2^2, q_{\phi(1)} = i, q_{\phi(2)} = j/\lambda) \\ &= \sum_{q_{\phi(1)}=i} P(O_1^1, O_1^2/q_{\phi(1)} = i)\, P(O_2^1, O_2^2/q_{\phi(2)} = j)\, P(q_{\phi(1)} = i)\, P(q_{\phi(2)} = j/q_{\phi(1)} = i) \\ &= \left[\sum_i \alpha_{\phi(1)}(i)\, a_{ij}\right] b_j(O_2^1, O_2^2) \end{aligned} \tag{51}$$

where $a_{ij} = P(\phi(t) = j/\phi(t-1) = i)$. It is the transition probability of moving from state i to state j and is assumed to be same as that given by the HMM.

$$\begin{aligned} \alpha_{\phi(3)}(j) &= P(O_1^1, O_1^2, O_2^1, O_2^2, O_3^1, q_{\phi(3)} = j/\lambda) \\ &= \sum_{q_{\phi(2)}=i} \sum_{q_{\phi(1)}=k} P(O_1^1, O_1^2, O_2^1, O_2^2, O_3^1, q_{\phi(1)} = k, q_{\phi(2)} = i, q_{\phi(3)} = j/\lambda) \\ &= \sum_{q_{\phi(1)}=i} \sum_{q_{\phi(2)}=k} P(O_1^1, O_1^2/q_{\phi(1)} = k) P(O_2^1, O_2^2/q_{\phi(2)} = i) \\ &\quad P(O_3^1/q_{\phi(3)} = j) P(q_{\phi(1)} = k) P(q_{\phi(2)} = i/q_{\phi(1)} = k) P(q_{\phi(3)} = j/q_{\phi(2)} = i) \\ &= \left[\sum_i \alpha_{\phi(2)}(i)\, a_{ij}\right] b_j(O_3^1) \end{aligned} \tag{52}$$

We assume a first order process. Here state j at $\phi(3)$ emits only O_3^1 and not O_2^2, as O_2^2 was already emitted at $\phi(2)$ by the HMM state. So we don't reuse vectors.

What was done in this example can be generalized to K patterns given the MPDTWpath ϕ between them and we get the recursive equation in equation 21. Since CMPVA-1 is similar to CMPFA-1, almost the same derivation (with minor changes) can be used to derive the recursive relation of CMPVA-1.

7. References

[Arslan & Hansen, 1996] Arslan, L.M. & Hansen, J.H.L. (1996). "Improved HMM training and scoring strategies with application to accent classification," *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*.

[Arslan & Hansen, 1999] Arslan, L.M. & Hansen, J.H.L. (1999). "Selective Training for Hidden Markov Models with Applications to Speech Classication," *IEEE Trans. on Speech and Audio Proc.*, vol. 7, no. 1.

[Bahl et al., 1983] Bahl, L.R.; Jelinek, F. & Mercer, R.L. (1983). "A maximum likelihood approach to continuous speech recognition", *IEEE Trans. PAMI*, PAMI-5 (2), pp. 179-190.

[Baum & Petrie, 1966] Baum, L.E. & Petrie, T. (1966). "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Stat.*, 37: pp. 1554-1563.

[Baum & Egon, 1967] Baum, L.E.& Egon, J.A. (1967). "An inequality with applications to statistical estimation for probabilistic functions of aMarkov process and to a model for ecology," *Bull. Amer. Meteorol. Soc.*, 73: pp. 360-363.

[Baum & Sell, 1968] Baum, L.E. & Sell, G.R. (1968). "Growth functions for transformations on manifolds," *Pac. J. Math.*,, 27 (2): pp. 211-227.

[Baum et al., 1970] Baum, L.E., Petrie, T., Soules, G. & Weiss, N. (1970). "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Stat.*, 41 (1): pp. 164-171.

[Baum, 1972] L.E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, 3: pp. 1-8, 1972.

[Cincarek et al., 2005] Cincarek, T.; Toda, T.; Saruwatari, H. & Shikano, K. (2005). "Selective EM Training of Acoustic Models Based on Sufficient Statistics of Single Utterances," *IEEE Workshop Automatic Speech Recognition and Understanding*..

[Cooke et al., 1994] Cooke, M.P.; Green, P.G. & Crawford, M.D. (1994). "Handling missing data in speech recognition," *Proc. Int. Conf. Spoken Lang. Process.*, pp. 1555-1558.

[Cooke et al., 2001] Cooke, M.; Green, P.; Josifovski, L. & Vizinho, A. (2001). "Robust automatic speech recognition with missing and unreliable acoustic data,"*Speech Commun.* 34(3), pp. 267-285.

[Fiscus, 1997] Fiscus, J.G. (1997). "A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)", *Proc. IEEE ASRU Workshop*, Santa Barbara.

[Gersho & Gray, 1992] Gersho, A. & Gray, R.M. (1992). *Vector Quantization and Signal Compression*, Kluwer Academic Publishers.

[Haeb-Umbach et al., 1995] Haeb-Umbach, R.; Beyerlein, P. & Thelen, E. (1995). "Automatic Transcription of Unknown Words in A Speech Recognition System," *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, pp. 840-843.

[Holter et al., 1998] Holter, T. & Svendsen, T. (1998). "Maximum Likelihood Modeling of Pronunciation Variation," *Proc, of ESCA Workshop on Modeling Pronunciation Variation for ASR*, pp. 63-66.

[Huang et. al, 2004] Huang, C.; Chen, T. & Chang, E. (2004). "Transformation and Combination of Hidden Markov Models for Speaker Selection Training," *Proc. Int. Conf. on Spoken Lang. Process.*, pp. 10011004.

[Itakura & Saito, 1968] Itakura, F. & Saito, S. (1968). "An Analysis-Synthesis Telephony Based on Maximum Likelihood Method," *Proc. Int'l Cong. Acoust.*, C-5-5.

[Juang & Rabiner, 1990] Juang, B.-H. & Rabiner, L.R. (1990). "The segmental K-means algorithm for estimating parametersof hidden Markov models," *IEEE Trans. Audio, Speech, and Signal Process.*, vol. 38, issue 9, pp. 1639-1641.

[Lleida & Rose, 2000] Lleida, E. & Rose, R.C. (2000). "Utterance verification in continuous speech recognition: decoding and training procedures", *IEEE Trans. on Speech and Audio Proc.*, vol. 8, issue: 2, pp. 126-139.

[Myers et al., 1980] Myers, C., Rabiner, L.R. & Rosenburg, A.E. (1980). "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-28(6): 623-635.

[Nair & Sreenivas, 2007] Nair, N.U. & Sreenivas, T.V. (2007). "Joint Decoding of Multiple Speech Patterns For Robust Speech Recognition," *IEEE Workshop Automatic Speech Recognition and Understanding*, pp. 93-98, 9-13 Dec. 2007.

[Nair & Sreenivas, 2008 a] Nair, N.U. & Sreenivas, T.V. (2008). "Forward/Backward Algorithms For Joint Multi Pattern Speech Recognition," *Proceeding of* 16*th European Signal Processing Conference (EUSIPCO-2008)*.

[Nair & Sreenivas, 2008 b] Nair, N.U. & Sreenivas, T.V. (2008). "Multi Pattern Dynamic Time Warping for Automatic Speech Recognition," *IEEE TENCON 2008*.

[Nilsson, 1971] Nilsson, N. (1971). *Problem-Solving Methods in Artificial Intelligence*, NY, NY, McGraw Hill.

[Nishimura et al., 2003] Nishimura, R.; Nishihara, Y.; Tsurumi, R.; Lee, A.; Saruwatari, H. & Shikano, K. (2003). "Takemaru-kun: Speech-oriented Information System for RealWorld Research Platform," *International Workshop on Language Understanding and Agents for RealWorld Interaction*, pp. 7078.

[Rabiner et al., 1986] Rabiner, L.R.; Wilpon, J.G. & Juang, B.H. (1986). "A segmental K-means training procedure for connected word recognition," *AT & T Tech. J.*, vol. 64. no. 3. pp. 21-40.

[Rabiner, 1989] Rabiner, L.R. (1989). "A tutorial to Hidden Markov Models and selected applications in speech recognition", *Proceedings of IEEE*, vol. 77, no. 2, pp. 257-285.

[Rabiner & Juang, 1993] Rabiner, L.R. & Juang, B.H. (1993). *Fundamentals of Speech Recognition.*, Pearson Education Inc.

[Raj & Stern, 2005] Raj B. & Stern, R.M. (2005). "Missing-feature approaches in speech recognition," *IEEE Signal Proc. Magazine.*, vol. 2, pp. 101-116.

[Sakoe & Chiba, 1978] Sakoe, H. & Chiba, S. (1978). "Dynamic programming optimization for spoken word recognition," *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-26 (1): 43-49.

[Schwartz & Chow, 1990] Schwartz, R. & Chow, Y.-L. (1990). "The N-best algorithms: an efficient and exact procedure for finding the N most likely sentence hypotheses", *Proc. IEEE ICASSP*, vol.1, pp. 81-84.

[Shannon, 1948] Shannon, C.E. (1948). "A Mathematical Theory of Communication", *Bell System Technical Journal*, vol. 27, pp. 379-423, 623-656.

[Singh et al., 2002] Singh, R.; Raj, B. & Stern, R.M. (2002). "Automatic Generation of Subword Units for Speech Recognition Systems," *IEEE Transactions on Speech and Audio Processing*, vol. 10(2), 89-99.

[Svendson, 2004] Svendsen, T. (2004). "Pronunciation modeling for speech technology," *Proc. Intl. Conf. on Signal Processing and Communication (SPCOM04)*.

[Soong & Hung, 1991] Soong, F.K.& Hung, E.-F. (1991). "A Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition," *Proc. IEEE ICASSP 91*, vol 1, pp. 705-708.

[Viterbi, 1967] Viterbi, A. (1967). "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Trans. Inf.Theory*, vol. IT-13, no.2, pp. 260-269.

[Wu & Gupta, 1999] Wu, J. & Gupta, V. (1999). "Application of simultaneous decoding algorithms to automatic transcription of known and unknown words", *Proc. IEEE ICASSP*, vol. 2, pp. 589-592.

[Yoshizawa et al., 2001] Yoshizawa, S.; Baba, A.; Matsunami, K.; Mera, Y.; Yamada, M.; Lee, A. & Shikano, K. (2001). "Evaluation on unsupervised speaker adaptation based on sufficient HMM statistics of selected speakers," *Proc. European Conference on Speech Communication and Technology*, pp. 1219-1222.

8

Overcoming HMM Time and Parameter Independence Assumptions for ASR

Marta Casar and José A. R. Fonollosa
Dept. of Signal Theory and Communications, Universitat Politècnica de Catalunya (UPC)
Spain

1. Introduction

Understanding continuous speech uttered by a random speaker in a random language and in a variable environment is a difficult problem for a machine. To take into account the context implies a broad knowledge of the world, and this has been the main source of difficulty in speech related research. Only by simplifying the problem – restricting the vocabulary, the speech domain, the way sentences are constructed, the number of speakers, and the language to be used, and controlling the environmental noise – has automatic speech recognition been possible.

For modeling temporal dependencies or multi-modal distributions of "real-world" tasks, Hidden Markov Models (HMMs) are one of the most commonly used statistical models. Because of this, HMMs have become the standard solution for modeling acoustic information in the speech signal and thus for most current speech recognition systems.

When putting HMMs into practice, however, there are some assumptions that make evaluation, learning and decoding feasible. Even if effective, these assumptions are known to be poor. Therefore, the development of new acoustic models that overcome traditional HMM restrictions is an active field of research in Automatic Speech Recognition (ASR).

For instance, the independence and conditional-independence assumptions encoded in the acoustic models are not correct, potentially degrading classification performance. Adding dependencies through expert knowledge and hand tuning can improve models, but it is often not clear which dependencies should be included.

Different approaches for overcoming HMM restrictions and for modeling time-domain dependencies will be presented in this chapter. For instance, an algorithm to find the beststate sequence of HSMMs (Hidden Semi-Markov Models) allows a more explicit modeling of context. Durations and trajectory modeling have also been on stage, leading to more recent work on the temporal evolution of the acoustic models. Augmented statistical models have been proposed by several authors as a systematic technique for modeling HMM additional dependencies, allowing the representation of highly complex distributions. These dependencies are thus incorporated in a systematic fashion, even if the price for this flexibility is high.

Focusing on time and parameter independence assumptions, we will explain a method for introducing N-gram based augmented statistical models in detail. Two approaches are presented: the first one consists of overcoming the parameter independence assumption by modeling the dependence between the different acoustic parameters and mapping the input

signal to the new probability space. The second proposal attempts to overcome the time independence assumption by modeling the temporal dependencies of each acoustic parameter.
The main conclusions obtained from analyzing the proposals presented will be summarized at the end, together with a brief dissertation about general guidelines for further work in this field.

2. ASR using HMM

2.1 Standard systems

Standard ASR systems rely on a set of so-called acoustic models that link the observed features of the voice signal with the expected phonetic of the hypothesis sentence. The most common implementation of this process is probabilistic, that is, Hidden Markov Models, or HMMs (Rabiner, 1989; Huang et al., 2001).
A Markov Model is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. This characteristic is defined as the Markov property. An HMM is a collection of states that fulfills the Markov property, with an output distribution for each state defined in terms of a mixture of Gaussian densities (Rabiner, 1993). These output distributions are generally determined by the direct acoustic vector plus its dynamic features (namely, its first and second derivatives), plus the energy of the spectrum. The dynamic features are the way of representing the context in HMMs, but generally they are only limited to a few subsequent feature vectors and do not represent long-term variations. Frequency filtering parameterization (Nadeu et al., 2001) has become a successful alternative to cepstral coefficients.
Conventional HMM training is based on Maximum Likelihood Estimation (MLE) criteria (Furui & Sandhi, 1992), via powerful training algorithms, such as the Baum-Welch algorithm or the Viterbi algorithm. In recent years, the discriminant training method and the criteria of Minimum Classification Error (MCE), based on the Generalized Probabilistic Descent (GPD) framework, has been successful in training HMMs for speech recognition (Juang et al., 1997). For decoding, both Viterbi and Baum-Welch algorithms have been implemented with similar results, but a better computational behavior is observed with the former.
The first implementations of HMM for ASR were based on discrete HMMs (or DHMM). DHMMs imply the need of a quantization procedure to map observation vectors from a continuous space to the discrete space of the statistical models. There is, of course, a quantization error inherent in this process that can be eliminated if continuous HMMs (or CHMMs) are used. In CHMMs, a different form of output probability functions is needed. Multivariate Gaussian mixture density functions are a clear first choice, as they can approximate any continuous density function (Huan et al., 2001). However, computational complexity can become a major drawback in the maximization of the likelihood by way of re-estimation, as it will need to accommodate the M-mixture observation densities used.
In many implementations, the gap between discrete and continuous mixture density HMMs has been bridged with some minor assumptions. For instance, in tied-mixture HMMs, the mixture density functions are tied together across all the models to form a set of shared kernels.
Another solution is the use of semi-continuous HMMs (or SCHMM), where a VQ codebook is used to map the continuous input feature vector $\mathbf{x}$ onto o, as in discrete HMMs. However,

in this case, the output probabilities are no longer directly used (as in DHMM), but are rather combined with the VQ density functions. That is, the discrete model-dependent weighting coefficients are combined with the continuous codebook probability density functions. Semi-continuous models can also be seen as equivalent to M-mixture continuous HMMs with all of the continuous output probability density functions shared among all Markov states. Therefore, SCHMMs do maintain the modeling ability of large-mixture density functions. In addition, the number of free parameters and the computational complexity can be reduced, because all of the probability density functions are tied together, thus providing a good compromise between detailed acoustic modeling and trainability.
However, standard ASR systems still don't provide convincing results when environmental conditions are changeable. Most of the actual commercial speech recognition technologies still work using either a restricted lexicon (i.e. digits, or a definite number of commands) or a semantically restricted task (i.e., database information retrieval, tourist information, flight information, hotel services, etc.). Extensions to more complex tasks and/or vocabulary still have a bad reputation in terms of quality, which entails the mistrust of both potential users and customers.
Due to the limitations found in HMM-based speech recognition systems, research has progressed in numerous directions. Among all the active fields of research in speech recognition, we will point out only those similar to the approach presented in this chapter.

2.2 Semi-continuous HMM

Semi-continuous hidden Markov models can be considered as a special form of continuous mixture HMMs, with the continuous output of probability density functions sharing a mixture Gaussian density codebook (see (Huang & Jack, 1989)). The semi-continuous output probability density function is represented by a combination of the discrete output probabilities of the model and the continuous Gaussian density functions of the codebook. Thus, the amount of training data required, as well as the computational complexity of the SCHMM, can be largely reduced in comparison to continuous mixture HMM. Thus, SCHMMs become the perfect choice for training small vocabulary and/or for low resource applications.
Moreover, the ease of combining and mutually optimizing the parameters of the codebook and HMM leads to a unified modeling approach. In addition, the recognition accuracy of semi-continuous HMMs is comparable to that of both discrete HMMs and continuous HMMs under some conditions, which include considering the same number of Gaussian mixtures for all techniques and keeping this number low. It is not a coincidence that these conditions apply to the applications in which we are interested: real-time and/or low-resource scenarios.

2.3 Time independence and parameter independence assumptions

In HMMs, there are some assumptions that make evaluation, learning and decoding feasible. One of them is the Markov assumption for the Markov chain (Huang et al., 2001), which states that the probability of a state s_t depends only on the previous state s_{t-1}. Also, when working with different parameters to represent the speech signal, we rely on the parameter independence assumption. This assumption states that the acoustical parameters modeled by HMMs are independent, and so are the output-symbol probabilities emitted.

However, in many cases, the independence and conditional-independence assumptions encoded in these latent-variable models are not correct, potentially degrading classification performance. Adding dependencies through expert-knowledge and hand-tuning improved models can be done, but it is often not clear which dependencies should be included.

For modeling dependencies between features, Gaussian mixture distribution-based techniques are very common. The parametric modeling of cepstral features with full covariance Gaussians using the ML principle is well-known and has led to good performance. However, although standard cepstral features augmented with dynamics information performs well in practice, some authors have questioned its theoretical basis from a discriminant analysis point of view (Saon et al., 2000). Thus, work has been done to extend LDA methods to HDA (Heteroscedastic Discriminant Analysis) (Saon et al., 2000) or maximum likelihood rotations such as LDA+MLLT. However, these techniques are expensive with real-time and/or low resource applications.

For modeling time-domain dependencies, several approaches have focused on studying the temporal evolution of the speech signal to optimally change the duration and temporal structure of words, known as duration modeling (Pylkkönen & Kurimo, 2003). However, incorporating explicit duration models into the HMM structure also breaks some of conventional Markov assumptions: when the HMM geometric distribution is replaced with an explicitly defined one, Baum-Welch and Viterbi algorithms are no longer directly applicable.

Thus, in Bonafonte et al. (1993), Hidden Semi-Markov Models (or HSMMs) were proposed as a framework for a more explicit modeling of duration. In these models, the first order Markov hypothesis is broken in the loop transitions. Then, an algorithm to find the best state sequence in the HSMM was defined, aiming for a more explicit modeling of context.

In another approach to overcome the temporal limitations of the standard HMM framework, alternative trajectory modeling (Takahashi, 1993) has been proposed, taking advantage of frame correlation. The models obtained can improve speech recognition performance, but they generally require a demoralizing increase in model parameters and computational complexity.

A smooth speech trajectory is also generated by HMMs through maximization of the models' output probability under the constraints between static and dynamic features, leading to more recent work on the temporal evolution of the acoustic models (Casar & Fonollosa, 2006b).

Therefore, a natural next step, given this previous research, is to work on a framework for dealing with temporal and parameter dependencies while still working with regular HMMs, which can be done by using augmented HMMs.

Augmented statistical models have been proposed previously as a systematic technique for modeling additional dependencies in HMMs, allowing the representation of highly complex distributions. Additional dependencies are thus incorporated in a systematic fashion. However, the price for flexibility is high, even when working with more computationally-friendly purposes (Layton & Gales, 2006).

In an effort to model the temporal properties of the speech signal, class labels modeling (Stemmer et al., 2003) has been studied in a double layer speech recognition framework (Casar & Fonollosa, 2006a). The main idea was to deal with acoustic and temporal information in two different steps. However, the complexity of a double decoding procedure was not offset by the results obtained. But temporal dependence modeling is still a challenge, and a less complex scheme needed to be developed.

The approach presented in this chapter consists of creating an augmented set of models. However, instead of modeling utterance likelihoods or the posterior probabilities of class labels, we focus on temporal and inter-parameter dependence.

3. Using N-grams for modeling dependencies

To better analyze the influence of temporal and parameter dependencies in recognition performance, both dependencies can be modeled in an independent fashion. Thus, a new set of acoustic models will be built for each case without losing the scope of regular HMMs.
For both cases, the most frequent combinations of features from the MFCC-based parameterized signal will be selected following either temporal or parameter dependence criteria. Language modeling techniques (i.e. by means of the CMU Statistical Language Modeling (SLM) Toolkit[1]) should be used for performing this selection. In this way, a new probability space can be defined, to which the input signal will be mapped, defining a new set of features.
Going into the mathematical formalism, we start by recalling how, in standard semi-continuous HMMs (SCHMMs), the density function $b_i(x_t)$ for the output of a feature vector x_t by state i at time t is computed as a sum over all codebook classes $m \in M$ (Huang et al., 2001):

$$b_i(x_t) = \sum_m c_{i,m} \cdot p(x_t \mid m,i) \approx \sum_m c_{i,m} \cdot p(x_t \mid m) \tag{1}$$

In our case, new weights should be estimated, as there are more features (inter-parameter dependencies or temporal dependencies) to cover the new probability space. Also, the posterior probabilities $p(x_t \mid m)$ will be modified, as some independence restrictions will no longer apply.
From this new set of features, a regular SCHMM-based training will be performed, leading to a new set of augmented statistical models.

3.1 Modeling inter-parameter dependence

In most HMM-based ASR systems, acoustic parameters are supposed to be independent from each other. However, this is no more than a practical assumption, as successive derivatives are by definition related to the parameters from which they are derived. Therefore, we can model the dependence between the different acoustic parameters, and thus overcome the parameter independence assumption.
Let us assume that we work with four MFCC features: cepstrum (f_0), its first and second derivatives (f_1,f_2) and the first derivative of the energy (f_3). We can express the joint output probability of these four features by applying Bayes' rule:

$$P(f_0, f_1, f_2, f_3) = P(f_0)P(f_1 \mid f_0)P(f_2 \mid f_1, f_0)P(f_3 \mid f_2, f_1, f_0) \tag{2}$$

where f_i corresponds to each of the acoustic features used to characterize the speech signal. Assuming parameter independence, HMM theory expresses equation (2) as:

$$P(f_0, f_1, f_2, f_3) = P(f_0)P(f_1)P(f_2)P(f_3) \tag{3}$$

[1] See http://www.speech.cs.cmu.edu

If parameter independence is to be overcome, some middle ground has to be found between equations (2) and (3). Thus, instead of using all dependencies to express the joint output probability, only the most relevant dependence relations between features are kept. For the spectral features, we take into account the implicit temporal relations between the features. For the energy, experimental results show in a more relevant dependence on the first spectral derivative than to the rest.
Thus, equation (2) is finally expressed as:

$$P(f_0, f_1, f_2, f_3) = P(f_0)P(f_1 \mid f_0)P(f_2 \mid f_1, f_0)P(f_3 \mid f_1) \tag{4}$$

In practice, not all of the combinations of parameters will be used for modeling each parameter dependency for each $P(f_i)$, but only the most frequent ones. Taking into account the parameter dependence restrictions proposed, a basic N-gram analysis of the dependencies in the training corpus is performed, defining those most frequent combinations of acoustic parameterization labels for each spectral feature. That is, we will consider dependence between the most frequent parameter combinations for each feature (considering trigrams and bigrams), and assume independence for the rest.
The input signal will be mapped to the new probability space. Recalling equation (1), we can redefine the output probability of state i at time t for each of the features used as $P_i(f_k)$, where f_k corresponds to each of the acoustic features used to characterize the speech signal.
Starting with the first acoustic feature, the cepstrum, the new output probability is defined as a sum over all codebook classes $m \in M$ of the new posterior probability function weighted by the original weights $c_{i,m}$ for the acoustic feature f_0. That is:

$$P_i(f_0) = \sum_m c_{i,m}^0 \cdot p(f_0 \mid m) \tag{5}$$

For the second acoustic feature, the first derivative of the cepstrum (f_1), the new output probability is defined as:

$$P_i(f_1) = \sum_m c_{i,m,\hat{m}_0}^1 \cdot p(f_1 \mid m) \tag{6}$$

The new weights in this output probability are defined according to N-gram-based feature combinations, taking advantage of the bi-gram $"\hat{m}_0, m"$, where $\hat{m}_0$ is the likeliest class for feature f_0 at each state i and time t considered in the sum of probabilities. It is defined as:

$$\hat{m}_0 = \arg\max_m p(f_0 \mid m) \tag{7}$$

When the bi-gram $"\hat{m}_0, m"$ is not defined, and $c_{i,m,\hat{m}_0}^1 = c_{i,m}^1$, which are the original weights for feature f_1.
For the third acoustic feature, the second derivative of the cepstrum (f_2), the new output probability is defined as:

$$P_i(f_2) = \sum_m c_{i,m,\hat{m}_0,\hat{m}_1}^2 \cdot p(f_2 \mid m) \tag{8}$$

Now the new weights are defined according to N-gram-based feature combinations as $c_{i,m,\hat{m}_0,\hat{m}_1}^2$. Extrapolating equation (7):

$$\hat{m}_k = \arg\max_m p\left(f_k \mid m\right) \tag{9}$$

Now, if the tri-gram "$\hat{m}_1, \hat{m}_0, m$" is not defined, $c^2_{i,m,\hat{m}_0,\hat{m}_1} = c^2_{i,m,\hat{m}_0}$ if the bi-gram "$\hat{m}_0, m$" applies, or $c^2_{i,m,\hat{m}_0,\hat{m}_1} = c^2_{i,m}$ otherwise.

Finally, for the energy:

$$P_i\left(f_3\right) = \sum_m c^3_{i,m,\hat{m}_1} \cdot p\left(f_3 \mid m\right) \tag{10}$$

where the new weights are defined according to the bi-grams "$\hat{m}_1, m$". If this bi-gram is not defined, again the original weights $c^3_{i,m}$ apply.

From these new output probabilities, a new set of SCHMMs can be obtained using a Baum-Welch training and used for decoding, following the traditional scheme.

3.2 Modeling temporal dependencies

Generally, HMM-based ASR systems model temporal dependencies between different frames by means of the successive derivatives of the acoustic features. However, a more explicit modeling of the time domain information seems relevant for improving recognition accuracy.

The observation probability distributions used in HMMs assume that successive information s_1 ... s_t within a state i can be considered independent. This is what is generally known as the Markov assumption for the Markov chain, and it is expressed as:

$$P\left(s_t \mid s_1^{t-1}\right) = P\left(s_t \mid s_{t-1}\right) \tag{11}$$

where s_1^{t-1} represents the state sequence $s_1, s_2, \ldots, s_{t-1}$.

Taking into account a state sequence of length N, equation (11) can be reformulated to:

$$P\left(s_t \mid s_1^{t-1}\right) = P\left(s_t \mid s_{t-N} \ldots s_{t-1}\right) \tag{12}$$

For simplicity, not all of the sequence of observations is taken into account, but only the two previous ones for each observation s_t, working with the 3-gram s_{t-2}, s_{t-1}, s_t. Then, equation (12) can be expressed as:

$$P\left(s_t \mid s_1^{t-1}\right) = P\left(s_t \mid s_{t-2}, s_{t-1}\right) \tag{13}$$

Applying independence among features (recall equation (3)), the output probability of each HMM feature will be expressed as:

$$P\left(f_i\right) = P\left(f_i \mid f_{i_{t-2}}, f_{i_{t-1}}\right) \tag{14}$$

Again, the most frequent combinations of acoustic parameterization labels can be defined, and a set of augmented acoustic models can be trained. The output probability (from equation (1)) of state i at time t for each feature k will be rewritten following the same line of argument as in previous sections (see section 3.1, and equations (5)-(9)).

Now:

$$P_i(f_k) = \sum_m c^k_{i,m,\hat{m}_{k,t-1},\hat{m}_{k,t-2}} \cdot p(f_k \mid m) \tag{15}$$

with

$$\hat{m}_{k,t-\mathrm{i}} = \arg\max_m p(f_k \mid m, t-i) \tag{16}$$

Notice that if the trigram $\hat{m}_{k,t-2}, \hat{m}_{k,t-1}, m$ does not exist, the bigram or unigram case will be used.

4. Some experiments and results

4.1 Methods and tools

For the experiments performed to test these approaches, the semi-continuous HMM -based speech recognition system RAMSES (Bonafonte et al., 1998) was used as reference ASR scheme, and it is also used in this chapter as baseline for comparison purposes.

When working with connected digit recognition, 40 semidigit models were trained for the first set of acoustic models, with the addition of one noisy model for each digit, each modeled with 10 states. Silence and filler models were also used, each modeled with 8 states. When working with continuous speech recognition, demiphones models were used. For the first set of acoustic models, each phonetic unit was modeled by several 4-state left-to-right models, each of them modeling different contexts. In the second (augmented) set of HMMs, each phonetic unit was modeled by several models that modeled different temporal dependencies, also using 4-state left-to-right models.

Connected digits recognition was used as the first working task for testing speech recognition performance, as it is still a useful practical application. Next, a restricted large vocabulary task was tested in order to evaluate the utility of the approach for today's commercial systems.

Different databases were used: the Spanish corpus of the SpeechDat and SpeechDatII projects[2], and an independent database obtained from a real telephone voice recognition application, known as DigitVox, were used for the experiments related to connected digits recognition. The Spanish Parliament dataset (PARL) of the TC-STAR project[3] was used for testing the performance of the models for continuous speech recognition.

4.2 Experiments modeling parameter dependencies

In the first set of experiments, we modeled parameter dependencies. The different configurations used are defined by the number of N-grams used for modeling the dependencies between parameters for each new feature. In the present case, no dependencies are considered for the cepstral feature, 2-grams are considered for the first cepstral feature and for the energy, and 2 and 3-grams for the second cepstral derivative.

---

[2] A.Moreno, R.Winksky, 'Spanish fixed network speech corpus' *SpeechDat Project*. LRE-63314.

[3] TC-STAR: Technology and corpora for speech to speech translation, www.tc-star.org

As explained in section 3, as we cannot estimate all the theoretical acoustic parameter combinations, we define those **N** most frequent combinations of parameterization labels for each spectral feature. A low **N** means that only some combinations were modeled, maintaining a low dimension signal space for quantization. On the other hand, increasing **N** more dependencies are modeled at the risk of working with an excessive number of centroids to map the speech signal.
Different configurations were tested. Each configuration is represented by a 4-digit string with the different values of **N** used for each feature. The total number of code words to represent each feature is the original acoustic codebook dimension corresponding to this feature plus the number of N-grams used. The different combinations that result in the configurations chosen were selected after several series of experiments, defined to either optimize recognition results or to simplify the number of N-grams used.
In Table 1, we present the best results obtained for connected digit recognition experiments. Results are expressed according to SRR (Sentence Recognition Rate) and WER (Word Error Rate) to measure the performance.

Database	Configuration	SRR	WER
SpeechDat	Baseline	90.51	2.65
	-/2000/2000,2000/20000	91.04	2.52
DigitVox	Baseline	93.30	1.27
	-/2000/2000,2000/2000	03.71	1.17

Table 1. Connected digit recognition rates modeling inter-parameter dependencies

We can see an important improvement in speech recognition for this task using the SpeechDat dataset, with a relative WER decrease of nearly a 5%. When using the DigitVox dataset, this improvement is slightly higher, with a relative WER decrease of 7.788%. Because both datasets are independent from the training datasets, we didn't expect adaptation of the solution to the training corpus.

4.3 Experiments modeling temporal dependencies

When modeling temporal dependencies, each new HMM feature models the temporal dependencies of the original acoustic features. Again, the different configurations are represented by a 4-digit string (henceforth **N**), with the number of N-grams used in equation (15) for modeling each acoustic parameter. In contrast to inter-parameter dependence modeling, a wider range of **N** leads to an increase in recognition accuracy. Thus, this is a more flexible solution, where we can chose between optimizing the accuracy and working with reasonable codebook size (close to the state-of-the art codebooks when working with standard implementations) while still improving the recognition performance.
First, we want to focus attention on the evolution of recognition performance regarding **N** and also analyze the differences in performance when testing the system with the SpeechDat database (a different set of recordings from the training dataset) or an independent database (DigitVox).
Table 2 presents the results for connected digit recognition, according to SRR and WER, working with both databases.

Database	Configuration	SRR	WER
SpeechDat	Baseline	90.51	2.65
	14113/13440/69706/6113	92.30	1.96
DigitVox	Baseline	93.30	1.27
	14113/13440/69706/6113	93.79	1.14

Table 2. Connected digit recognition rates modeling time dependencies

Results obtained with the SpeechDat dataset show that by modeling time dependencies, we can achieve a great improvement in recognition, outperforming the inter-parameter dependencies modeling approach with a relative WER reduction of around 26% compared to baseline results. However, the improvement when using the DigitVox dataset was slightly lower, with a relative WER reduction of 10.2%. Thus, this solution seems more likely to be adapted to the training corpus for connected digit recognition.

To test whether this time dependencies modeling based solution works better using a bigger (and wider) training corpus, continuous speech recognition was used, with new sets of acoustic models based on demiphones, using the PARL dataset.

The results presented in Table 3 show a WER reduction between 14.2% and 24.3%. We observe some saturation in WER improvement when **N** is increased over certain values: after reaching optimum values, WER improvement becomes slower, and we should evaluate if the extra improvements really do justify the computational cost of working with such large values of **N** (which means working with high codebook sizes). Afterwards, additional WER improvement tends to zero, so no extra benefit is obtained by working with a very high number of N-grams. Thus a compromise between the increase in codebook size and the improvement in recognition accuracy is made when deciding upon the best configuration.

Database	Configuration	WER	WER_{var}
TC-STAR	Baseline	28.62	-
	3240/2939/2132/ 6015	24.56	14.19%
	7395/6089/4341/ 8784	21.73	24.07%
	20967/18495/17055/15074	21.66	24.32%

Table 3. Continuous speech recognition rates modeling time dependencies

The performance of the time dependencies modeling based system compared to the reference ASR system has also been analyzed in terms of the computational cost of recognition. Despite of the computational cost increase associated with the complexity of the system's training scheme, the system clearly outperforms the reference system in general terms. This good performance is due to a reduction in the computational cost of recognition of about 40% for those solutions which are a good compromise between codebook size

increase and recognition accuracy improvement (i.e. N-gram configuration "7395/6089/4341/8784" in Table 3).

6. Discussion

The future of speech-related technologies is clearly connected to the improvement of speech recognition quality. Commercial speech recognition technologies and applications still have some limitations regarding vocabulary length, speaker independence and environmental noise or acoustic events. Moreover, real-time applications still miss some improvement with the system delays.

Although the evolution of ASR needs to deal with these restrictions, they should not be addressed directly. Basic work on the core of the statistical models is still needed, which will contribute to higher level improvements.

HMM-based statistical modeling, the standard state-of-the-art for ASR, is based on some assumptions that are known to affect recognition performance. Throughout this chapter, we have addressed two of these assumptions by modeling inter-parameter dependencies and time dependencies. We noted different approaches for improving standard HMM-based ASR systems introducing some actual solutions.

Two proposals for using N-gram-based augmented HMMs were also presented. The first solution consists of modeling the dependence between the different acoustic parameters, thus overcoming the parameter independence assumption. The second approach relies on modeling the temporal evolution of the regular frequency-based features in an attempt to break the time independence assumption.

Experiments on connected digit recognition and continuous speech recognition have also been explained. The results presented here show an improvement in recognition accuracy, especially for the time dependencies modeling based proposal. Therefore, it seems that time-independence is a restriction for an accurate ASR system. Also, temporal evolution seems to need to be modeled in a more detailed way than the mere use of the spectral parameter's derivatives.

It is important to note than a more relevant improvement is achieved for continuous speech recognition than for connected digit recognition. For both tasks, independent testing datasets were used in last instance. Hence, this improvement does not seem to be related to an adaptation of the solution to the training corpus, but to better modeling of the dependencies for demiphone-based models. Thus, more general augmented models were obtained when using demiphones as HMM acoustic models.

Moreover, although the present research solutions should not be especially concerned with computational cost (due to the constant increase in processing capacity of computers), it is important to keep in mind implementation for commercial applications and devices. Taking computational cost into consideration, we find that the training computational cost increase of this modeling scheme clearly pays off by reducing the computational cost of recognition by about 40%.

Further work will be needed to extend this method to more complex units and tasks, i.e. using other state-of-the-art acoustic units and addressing very large vocabulary ASRs or even unrestricted vocabulary tasks.

8. References

Bonafonte, A.; Ros, X. & Mariño, J.B. (1993). An efficient algorithm to find the best state sequence in HSMM, *Proceedings of European Conf. On Speech Technology (EUROSPEECH)*

Bonafonte, A.; Mariño, J.B.; Nogueiras, A. & Fonollosa, J.A.R. (1998). Ramses: el sistema de reconocimiento del habla continua y gran vocabulario desarrollado por la UPC, *VIII Jornadas de Telecom I+D*

Casar, M.; *Fonollosa,* J.A.R. & Nogueiras,A. (2006a). A path based layered architecture using HMM for automatic speech recognition, *Proceedings of ISCA European Signal Processing Conference (EUSIPCO)*

Casar, M. & Fonollosa, J.A.R. (2006b). Analysis of HMM temporal evolution for automatic speech recognition and utterance verification, *Proceedings of IEEE Int. Conf. On Spoken Language Processing (ICSLP)*

Furui, S. & Sandhi, M. (1992). *Advances in Speech Signal Processing,* Marcel Dekker, Inc., ISBN:0-8247-8540, New Jork, USA

Huang, X.D. & Jack, M.A. (1998). Unified techniques for vector quantisation and Hidden Markov modeling using semi-continuous models, *Proceedings of IEEE Int. Conf. On Acoustics, Speech and Signal Processing (ICASSP)*

Huang, X.; Acero, A. & Hon, H.W. (2001). *Spoken Language Processing,* Prentice Hall PTR, ISBN:0-13-022616-5, New Jersey, USA

Layton, M.I. & Gales, M.J.F. (2006). Augmented Statistical Models for Speech Recognition, *Proceedings of IEEE Int. Conf. On Acoustics, Speech and Signal Processing (ICASSP)*

Mariño, J.B.; Nogueiras, A.; Paches-Leal, P. & Bonafonte, A. (2000). The demiphone: An efficient contextual subword unit for continuous speech recognition. *Speech Communication,* Vol.32, pp:187-187, ISSN:0167-6393

Nadeu, C; Macho, D. & Hernando, J. (2001). Time and frequency filtering of filter-bank energies for robust HMM speech recognition. *Speech Communication,* Vol.34, Issues 1-2 (April 2001) pp:93-114, ISSN:0167-6393

Pylkkönen, J. & Kurimo, M. (2003). Duration modeling techniques for continuous speech recognition, *Proceedings of European Conf. On Speech Technology (EUROSPEECH)*

Rabiner, L.R. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition, *Proceedings of the IEEE, No.2, Vol.77, pp:257-289,* ISSN:0018-9219

Rabiner, L. (1993). *Fundamentals of Speech Recognition,* Prentice Hall PTR, ISBN:0-13-015157-2, New Jersey, USA

Saon, G.; Padmanabhan, M.; Goinath, R. & Chen, S. (2000). Maximum likelihood discriminant feature spaces, *Proceedings of IEEE Int. Conf. On Acoustics, Speech and Signal Processing (ICASSP)*

Stemmer, G.; Zeissler, V.; Hacker, C.; Nöth, E. & Niemann, H. (2003). Context-dependent output densities for Hidden Markov Models in Speech recognition, *Proceedings of European Conf. On Speech Technology (EUROSPEECH)*

Takahashi, S. (1993). Phoneme HMMs constrained by frame correlations, *Proceedings of IEEE Int. Conf. On Acoustics, Speech and Signal Processing (ICASP)*

9

Practical Issues of Building Robust HMM Models Using HTK and SPHINX Systems

Juraj Kacur and Gregor Rozinaj
Slovak University of Technology, Faculty of Electrical Engineering and Information Technology, Bratislava
Slovakia

1. Introduction

For a couple of decades there has been a great effort spent to build and employ ASR systems in areas like information retrieval systems, dialog systems, etc., but only as the technology has evolved further other applications like dictation systems or even automatic transcription of natural speech (Nouza et al., 2005) are emerging. These advanced systems should be capable to operate on a real time base, must be speaker independent, reaching high accuracy and support dictionaries containing several hundreds of thousands of words.

These strict requirements can be currently met by HMM models of tied context dependent (CD) phonemes with multiple Gaussian mixtures, which is a technique known from the 60ties (Baum & Eagon, 1967). As this statistical concept is mathematically tractable it, unfortunately, doesn't completely reflect the physical underlying process. Therefore soon after its creation there have been lot of attempts to alleviate that. Nowadays the classical concept of HMM has evolved into areas like hybrid solutions with neural networks, utilisation of different than ML or MAP training strategies that minimize recognition errors by the means of corrective training, maximizing mutual information (Huang et. al., 1990) or by constructing large margin HMMs (Jiang & Li, 2007). Furthermore, a few methods have been designed and tested aiming to suppress the first order Markovian restriction by e.g. explicitly modelling the time duration (Levinson, 1986), splitting states into more complex structures (Bonafonte et al., 1996), using double (Casar & Fonollosa, 2007) or multilayer structures of HMM. Another vital issue is the robust and accurate feature extraction method. Again this matter is not fully solved and various popular features and techniques exist like: MFCC and CLPC coefficients, PLP features, TIFFING (Nadeu & Macho, 2001), RASTA filter (Hermasky & Morgan, 1994), etc.

Even despite the huge variety of advanced solutions many of them are either not general enough or are rather impractical for the real-life employment. Thus most of the currently employed systems are based on continuous context independent (CI) or tied CD HMM models of phonemes with multiple Gaussian mixtures trained by ML or MAP criteria. As there is no analytical solution of this task, the training process must be an iterative one (Huang et al., 1990). Unfortunately, there is no guarantee of reaching local maxima, thus lot of effort is paid to the training phase in which many stages are involved. Thus there are some complex systems that allow convenient and flexible training of HMM models, where the most famous are HTK and SPHINX.

This chapter provides you with the description of some basic facilities and methods implemented by HTK and SPHINX systems and guides you through a thorough process of building speaker independent CDHMM models using the professional database MOBILDAT-SK (Darjaa et al., 2006). First, basic tolls for building practical HMM models are described using HTK and SPHINX facilities. Then several experiments revealing the optimal tuning of the training phase are discussed and evaluated ranging from: selecting feature extraction methods and their derivatives, controlling and testing the overtraining phenomenon, selecting modelled units: CI and CD phonemes vs. models of functional words, setting proper tying options for CD phonemes, etc. Further, the popular training procedures for HTK and SPHNIX systems will be briefly outlined, namely: REFREC (Linderberg et al., 2000) / MASPER (Zgank & Kacic, 2003) and SphinxTrain (Scriptman, 2000). After the presentation of both training schemes the newly suggested modifications are discussed, tested and successfully evaluated. Finally, the achieved results on both systems are compared in terms of the accuracy, memory usage and the training times.
Thus the following paragraphs should give you the guideline how to adjust and build both robust and accurate HMM models using standard methods and systems on the professional database. Further, if it doesn't provide you with the exact settings, because they may be language specific, at least it should suggest what may be and what probably is not so relevant in building HMM models for practical applications.

2. Systems for building robust HMMs

2.1 Hidden Markov Toolkit- HTK

The HTK system is probably the most widely employed platform for training HMM models. Its outputs (HMMs) adhering to the suggested training steps are regarded as a sort of standard and are believed to be eligible for real life, large vocabulary ASR systems. The latest version is 3.4, however, the results are related to 3.2.1 version (Young et al., 2002).
HTK is a complex tool that provides advanced and flexible means for any stage of the HMM training: speech and data processing, definition of HMM models (discrete, continuous and semi-continuous), dictionary related tasks, initializations and training methods, model enhancement and adaptation tools, it can use both finite state grammar (BNF) or statistical language models via N grams, has tools for online as well as offline recognition and implements various evaluation methods, etc. Furthermore, each release is accompanied with a very precise documentation.
As some of the facilities would be directly involved in our experiments let us mention them. HTK supports many speech extraction methods like: various filter banks, LPC and CLPC coefficients, PLP, and MFCC parameters. Except that several auxiliary features are available like: normalized energy, differential and acceleration coefficients, cepstral mean subtraction and vocal tract length normalization. It supports and process description files with or without time labels. When the time information is available the Viterbi training can be used for the initialization phase and the classical Baum-Welche algorithm for the training, otherwise the flat start method and the embedded training are the only options. Moreover, to speed up the training process and to eliminate possible error recordings, both forward and incremental backward pruning methods can be applied. HTK supports discrete (DHMM), continuous (CDHMM) and semi-continuous (SCHMM) models. Any structure of the transition matrix is allowed and can differ from model to model. Moreover there are two non- emitting states at both ends of each model which allow the construction of the so called

T model. Furthermore, each element of the model (means, variances, mixtures, etc.) can be tied to the corresponding elements of other models. In HTK there are implemented 2 methods for parameter's tying, namely: the data driven one and the decision trees. The decoder supports forced alignment for multiple pronunciations, and the time alignment that can be performed on different levels and assess multiple hypotheses as well. To ease the implementation for online systems a separate recognition tool called ATK has been released. Of course an evaluation tool supporting multiple scoring methods is available.

2.2 SPHINX

The SPHINX system is eligible for building large vocabulary ASR systems since late 80ties (Lee et al., 1990). Currently there are SPHINX 2, 3, 3.5 and 4 decoder versions and a common training tool called SphinxTrain. The latest updates for SphinxTrain are from 2008, however, here mentioned options and results will refer to the version dated back to 2004. Unfortunately, the on-line documentation is not extensive, so the features mentioned here onwards are only those listed in manuals dated back to 2000 (Scriptman, 2000).
SphinxTrain can be used to train CDHMM or SCHMM for SPHINX 3 and 4 decoders (conversion for version 2 is needed). SphinxTrain supports MFCC and PLP speech features with delta or delta-delta parameters. Transcription file contains words from a dictionary, but neither multilevel description nor time labels are supported. There are 2 dictionaries, the main for words (alternative pronunciations are allowed but in the training process are ignored), and the second one is the so called filler dictionary where non-speech models are listed. The main drawback is the unified structure of HMM models that is common to all models both for speech and non-speech events. At the end of each model there is one non-emitting state, thus no "T" model is supported. Further, it is possible to use only the embedded training and the flat start initialization processes. Observation probabilities are modelled by multi mixture Gaussians and the process of gradual model enhancement is allowed. SphinxTrain performs tying of CD phonemes by constructing decision trees; however no phoneme classification file is required as the questions are automatically formed. Instead of setting some stoppage conditions for the state tying, the number of tied states must be provided by the designer prior to the process which expects deep knowledge and experience. Unlike HTK, only whole states can be tied. Apart of the SphinxTrainer there is a statistical modelling tool (CMU) for training unigrams, bigrams and trigrams.

3. Training database MOBILDAT–SK

A crucial aspect to succeed in building an accurate and robust speaker independent recognition system is the selection of the proper training database. So far there has been designed and compiled many databases following different employment assumptions and designing goals, like: AURORA, TIMIT, SPEECHDAT, SPEECON, etc. Further, the task of recognition is more challenging in adverse environments and requires more steps, additional pre-processing and more sophisticated handling. Since we want to demonstrate to the full extend the capabilities, options, modification and pitfalls of the HMM training process, we decided to use the Slovak MOBILDAT database (Darjaa et al., 2006) which was recorded over GSM networks and generally provides more adverse environments (wider range of noises, lower SNRs, distortions by compression techniques and short lapses of connections). The concept of MOBILDAT database is based on the widely used structure of

the SPEECHDAT database, whose many versions have been built for several languages using fix telephone lines and are regarded as professional databases.
The Slovak MOBILDAT-SK database consists of 1100 speakers that are divided into the training set (880) and the testing set (220). Each speaker produced 50 recordings (separate items) in a session with the total duration ranging between 4 to 8 minutes. These items were categorized into the following groups: isolated digit items (I), digit/number strings (B,C), natural numbers (N), money amounts (M), yes/no questions (Q), dates (D), times (T), application keywords (A), word spotting phrase (E), directory names (O), spellings (L), phonetically rich words (W), and phonetically rich sentences (S, Z). Description files were provided for each utterance with an orthographical transcription but no time marks were supplied. Beside the speech, following non- speech events were labeled too: truncated recordings (~), mispronunciation (*), unintelligible speech (**), filed pauses (fil), speaker noise (spk), stationary noise (sta), intermitted noise (int), and GSM specific distortion (%). In total there are 15942 different Slovak words, 260287 physical instances of words, and for 1825 words there are more than one pronunciation listed (up to 5 different spellings are supplied). Finally, there are 41739 useable speech recordings in the training portion, containing 51 Slovak phonemes, 10567 different CD phonemes (word internal) and in total there are slightly more than 88 hours of speech.

4. Robust and accurate training process

Our final goal is to choose a proper training scheme and adjust its relevant parameters in order to get robust and accurate HMM models that can be used in practical large vocabulary applications like: dialog systems, information retrieval systems and possibly dictation or automatic transcription systems. However, issues regarding the construction of stochastic language models represented by unigrams, bigrams or generally N- grams are out of the scope of this document. Thus in this section, aspects like: eligible speech features, HMM model structures and modelled units, overtraining phenomenon, and the tying of states will be discussed.

4.1 Feature extraction for speech recognition

One of the first steps in the design of an ASR system is to decide which feature extraction technique to use. At the beginning it should be noted that this task is not yet completely solved and a lot of effort is still going on in this area. The aim is to simulate the auditory system of humans, mathematically describe it, simplify for practical handling and optionally adapt it for a correct and simple use with the selected types of classification methods.
A good feature should be sensitive to differences in sounds that are perceived as different in humans and should be "deaf" to those which are unheeded by our auditory system. It was found (Rabiner & Juan, 1993) that the following differences are audible: different location of formants in the spectra, different widths of formants and that the intensity of signals is perceived non-linearly. On the other hand, following aspects do not play a role in perceiving differences: overall tilt of the spectra like: $X(\omega)\omega^{\alpha}$, where α is the tilt factor and $X(\omega)$ is the original spectra, filtering out frequencies laying under the first formant frequency, removing frequencies above the 3rd format frequency, and a narrow band stop filtering.
Furthermore, features should be insensitive to additive and convolutional noises or at least they should represent them in such a way that these distortions are easy to locate and

suppress in the feature space. Finally, when using CDHMM models it is required for the feasibility purposes that the elements of feature vectors should be linearly independent so that a single diagonal covariance matrix can be used. Unfortunately, yet there is no feature that would ideally incorporate all the requirements mentioned before.

Many basic speech features have been designed so far, but currently MFCC and PLP (Hönig et al., 2005) are the most widely used in CDHMM ASR systems. They both represent some kind of cepstra and thus are better in dealing with convolutional noises. However, it was reported that some times in lower SNRs they are outperformed by other methods, e.g. TIFFING (Nadeu & Macho, 2001). Furthermore, the DCT transform applied in the last step of the computation process minimize the correlation between elements and thus justifies the usage of diagonal covariance matrices. Besides those static features it was soon discovered that the changes in the time (Lee at al., 1990) represented by delta and acceleration parameters play an important role in modelling the evolution of speech. This is important when using HMMs as they lack the natural time duration modelling capability. Overall energy or zero cepstral coefficients with their derivations also carry valuable discriminative information thus most of the systems use them. Furthermore, to take the full advantage of cepstral coefficients, usually a cepstral mean subtraction is applied in order to suppress possible distortions inflicted by various transmission channels or recording devices. At the end we shall not forget about the liftering of cepstra in order to emphasise its middle part so that the most relevant shapes of spectra for recognition purposes would be amplified. Well, this appealing option has no real meaning when using CDHMM and Gaussian mixtures with diagonal covariance matrices. In this case it is simply to show that the liftering operation would be completely cancelled out when computing Gaussian pdf.

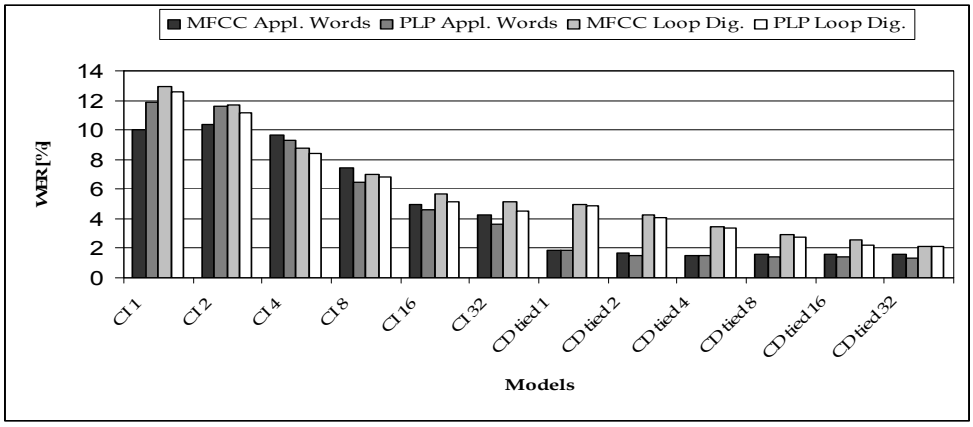


Fig. 1. Word error rates of PLP and MFCC features for application words and looped digits tests as a function of HMM models (CI and tied CD with multiple mixtures).

All the above-mentioned features and auxiliary settings were tested and evaluated on our database in terms of the recognition accuracy. Three tests were done on the test set portion of the database: single digits, digits in the loop, and application words. The training was based on the MASPER training procedure (will be presented later in the text) using the HTK system. In fig. 1 there are shown results for both MFCC and PLP features with delta, acceleration, and C0 coefficients, modified by the mean subtraction (this setting showed the

best results for both features). These were calculated over different HMM models (CI and tied CD phoneme models with multiple Gaussian mixtures) and both the application words and looped digit tests. From these 2 tests one can induce that slightly better results are obtained by PLP method, but in order to get a numeric evaluation of the average WER for all the models, both tests for MFCC and PLP were computed separately. These averaged errors over models and tests revealed that PLP is slightly better, scoring 20.34% while MFCC showed 20.96% of WER that amounts to a 3% drop in an average word error rate. Further, we investigated the significance of auxiliary features and modification techniques. For both methods the cepstral mean subtraction brought essentially improved results on average by 33.83% for MFCC and 21.28%for PLP. That reveals the PLP is less sensitive to the cepstral mean subtraction, probably, because it uses non linear operations (equal loudness curve, 0.33 root of the power, calculation of the all pole spectra) applied prior to the signal is transformed by the logarithm and cepstral features are calculated. Next the role of C0 (zero cepstral coefficient) was tested and compared to the solely static PLP and MFCC vectors, where it brought relative improvement by 19.7% for MFCC and 9.7% for PLP, again PLP showed to be less sensitive to the addition of a static feature or modification. Next the inclusion of delta coefficients disclosed that their incorporation brought down the averaged error by 61.15% for MFCC and 61.41% for PLP. If this absolute drop is further transformed to the relative drop calculated over a single difference coefficient (if all are equally important), it shows that one delta coefficient on average causes a 4.7% WER drop for MFCC and 4.72% for PLP. Finally, the acceleration coefficients were tested, and their inclusion resulted in a 41.16% drop of WER for MFCC and 52.47% drop for PLP. Again, if these absolute drops are calculated for a single acceleration coefficient it was found that one such a coefficient causes on average a 3.16% drop of WER for MFCC and a 4.03% for PLP. Interestingly enough, both dynamic features caused to be more significant for PLP than for MFCC in relative numbers, however, for the additional C0 (static feature) this was just the opposite. That may suggest that PLP itself (in this task) is better in extracting static features for speech recognition.

4.2 Discrete, continuous or semi-continuous HMM

The issue of the observation probability modelling will not be experimentally tested here. There have been many discussions and experiments on which type of HMM models is better in certain environments, etc. (Huang et al., 1990), but let us just in brief mention several fact and findings. The usage of discrete HMM (non-parametric modelling of the distribution) has clear advantage of being able to model any distribution, however, it requires huge amount of data to do so and moreover it uses the VQ procedure. This powerful technique introduces an irreversible distortion and is based on a vague notion of the acoustic distance (Rabiner & Juan, 1993). On the other hand, Gaussian mixture model (CDHMM) does not introduce anything like that, but requires lot of computing and for some complex models (lot of free parameters) it may not produce robust estimations. To eliminate both of those problems, semi-continuous HMM were introduced and showed better result in some applications (Huang et al., 1990). However, the fact that all Gaussain mixtures are trained on and share the same data which for some phonemes are from linguistic and physical pint of view completely different, poses an accuracy problem.

Even though the most successful systems are based on CDHMM, in some applications with higher degree of noise presence DHMM or discrete mixture HMM (DMHMM) were

reported to provide more accurate results (Kosaka et al., 2007). Usually this is explained by the inability of Gausian mixture pdf to model the occurrence of noisy speech. However, this is not the case as for example, the theoretical results from the artificial neural networks domain, namely the radial bases function (RBF), roughly say that a RBF network can approximate any continuous function defined on a compact set with the infinitely small error (Poggio & Girosi, 1990). Thus it poses as a universal approximator. If we compare the structure of a RFB network with N inputs (size of a feature vector), M centres (Gaussian mixtures) and one output (probability of a feature vector) we find out that these are actually the same. Generally, Gaussian mixtures can be viewed as an approximation problem how to express any continuous function of the type f: R^N-> R by the means of sum of Gaussian distributions, which is just what the RBF networks do. Thus the derived theoretical results for RBF must also apply to this CDHMM case regarding the modelling ability of Gaussians mixtures. Unfortunately, the proof says nothing about the number of mixtures (centres). Therefore, based on these theoretical derivations we decided to use CDHMM without additional experiments.

4.3 Modeling of speech units

Another issue to be tackled before building up an ASR system is to decide which speech and non speech units to model. In the early times where only small vocabulary systems were needed the whole word approach was the natural choice and exhibited good results. This method is rather unpractical for open systems where new words should be easily added and totally infeasible for large vocabulary, speaker independent applications where several dozens of realizations for every word are required. The opposite extreme is to construct models only for single phonemes which would solve the flexibility and feasibility problems. However, pronunciations of phonemes are strongly dependent on the context they are uttered in, the so called coarticulation problem, which caused a sudden drop in the accuracy. Next natural step was the inclusion of the context for each phoneme, both the left and right equally, which resulted in the context dependent phonemes (Lee et al., 1990). This obviously increased the accuracy but because of their huge number the problem of robustness re-emerged. Theoretically there are (in Slovak) 51^3 CD phonemes but practically there are about 11000 of them in 80 hours of recorded speech (specially constructed database in order to contain phonetically rich sentences like MOBILDAT). Thus the latest step led to the building of tied CD phonemes where phonetically and /or feature similar logical states are mapped to the same physical one. To allow more degrees of freedom usually states of models are tied instead of the whole models, however, if all states of a model are tied to another one then, such a couple creates one physical model. Still, the option of modelling the whole words may be appealing especially for application words that are quite frequent and usually exhibit huge variation in their sound forms (Lee et al., 1990). Furthermore, frequent and critical words or even phrases that are vital from the accuracy point of view, like yes and no, digits etc. can be modelled separately as unique models (Huang et al., 1990). Except speech units also other non-speech events have to be modelled so that a whole conversation could be correctly expressed by the concatenated HMM models. Then those usual events in a real conversation must be identified and classified according to their location, origin of creation and physical character. It is common to use a general model of a background which should span different time intervals and thus must allow backward connections. Events that are produced by the speaker himself (exhaling, sneezing, coughing,

laughing etc.) can be modelled by a unique model, but to increase the modelling ability these events are further divided into groups e.g. sound produced unintentionally like coughing, sneezing, etc. and intentional sounds like laughing, hesitating- various filling sounds, etc.

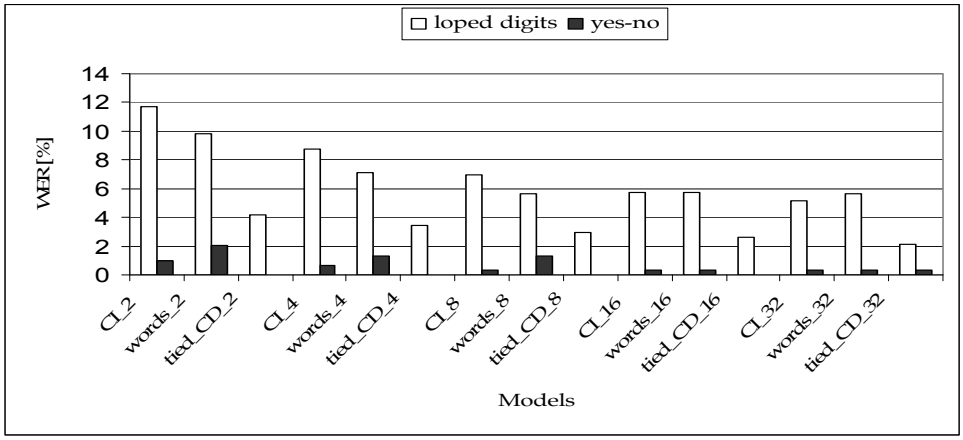


Fig. 2. Word error rates for CI, whole word, and tied CD phoneme models with different number of mixtures and both looped digit and yes-no tests. There are 3 states per a phoneme and a strictly left right structure of models.

To determine which models perform best and at what cost the following experiments were executed. CI and tied CD phoneme models were train with different number of Gaussian mixtures as was suggested by the REFREC or MASPER training schemes. To verify the effectiveness of the whole word models, models for digits and yes / no words were constructed as well. The whole word models consisted of the same number of states as their phoneme-concatenated counterparts and followed the same transition matrix structure (strictly left right, no skip). However, to utilize more efficiently the whole word models in mimicking the co-articulation effect, the HMM structure was enhanced so as to allow a one state skip. This structure was tested for whole word models as well as for CI and CD models created by the MASPER training scheme. In fig. 2 there are shown results for whole word models, CI and tied CD phoneme models with different number of mixtures, 3 states per a phoneme, and with a strictly left-right structure, for looped digits and yes / no tests. The same results for the left - right structure with one allowed state to be skipped and 3 states per a phoneme are depicted in fig. 3.

For the strict left right structure of HMM models there is surprisingly very little difference in terms of averaged WER between CI phonemes 4.09% and whole word models 3.94%. The tied CD phoneme models outperformed even the whole word models scoring on average only 1.57% of WER. Similar tests with the one state skip structure however, brought different results as seen in fig. 3. The averaged WER for CI models is 5.46%, tied CD models scored 3.85% and the whole word models 3.12%. These results deserve few comments. First an obvious degradation of WER for CI by 33.4% and tied CD phoneme models by 145% when moving from strictly left - right structure to the one state skip structure that potentially allows more modelling flexibility. By introducing additional skips the minimal

occupancy in a phoneme model has reduced to only one time slot (25ms) comparing to the original 3 (45ms) which is more realistic for a phoneme. By doing so some phonemes were in the recognition process passed unnaturally fast, that ended up in a higher number of recognized words. This is known behaviour and is tackled by introducing the so called word insertion penalty factor that reduces the number of words the best path travels through. In the case of short intervals like phonemes there is probably only a small benefit in increasing the duration flexibility of a model that is more obvious for CD models as they are even more specialized. On the other hand, when modelling longer intervals like words which have strong and specific co-articulation effects inside, the increased time duration flexibility led to the overall improved results by 4.5%. However, when comparing the best tied CD phoneme models with the best word models, the tied CD models still provided better results. This can be explained by their relatively good accuracy as they take in account the eminent context and their robustness because they were trained from the whole training section of the database. On the contrary, the word models were trained only from certain items, like digits and yes - no answers, so there might not have been enough realizations. Therefore the appealing option for increasing the accuracy by modelling whole functional words should not be taken for granted and must be tested. If there is enough data to train functional word models then the more complex structures are beneficial.

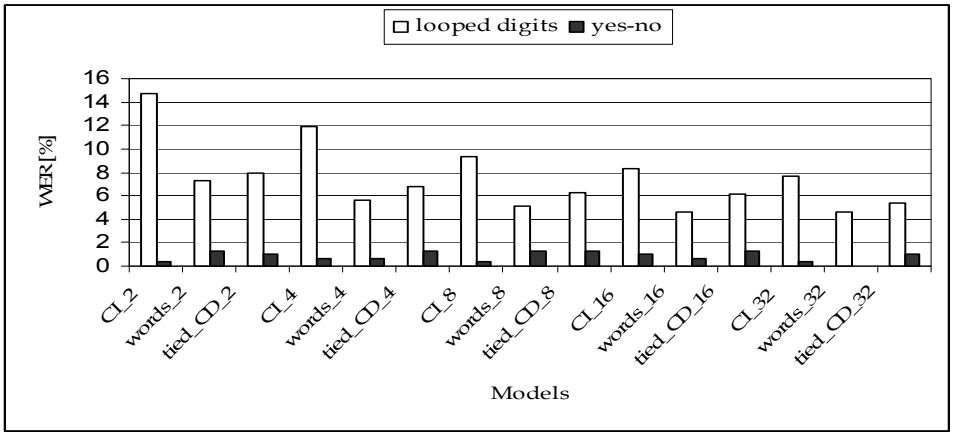


Fig. 3. Word error rates for CI, whole word, and tied CD phoneme models with different number of mixtures and both looped digit and yes-no tests. There are 3 states per a phoneme and a left right structure of HMM models with one allowed state to skip.

4.4 Overtraining phenomenon for HMM

Using the limited data which do not cover the whole feature space proportionally and being restricted to the models that only roughly approximate the physical underlying process, it was soon realized that during the course of training and despite to the assured convergence of the training algorithms, results started to deteriorate on the unseen data at the certain point. Although this phenomenon is ubiquities in all real classification and decision taking algorithms, some methods like artificial neural networks, decision trees, etc. are well known to be vulnerable. In brief, the overtraining is simply explained by the fact that as the training

converges on the training data, the models or classifiers are getting too specific about the training data and inevitably start to loose a broader view over a particular task (loosing the generalization ability). There are more methods to detect and eliminate the overtraining but let's mention some of them: the usage of test sets, restricting the complexity of models, gathering more general training data, setting floors for parameters, etc.

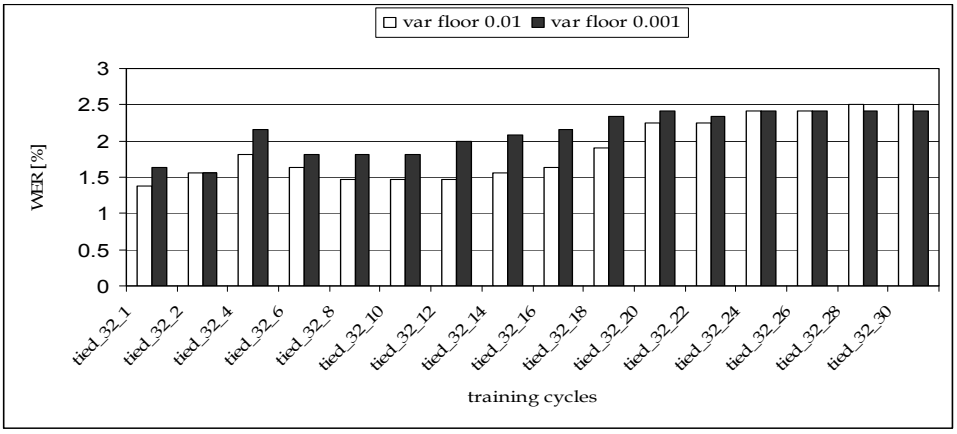


Fig. 4. WER as a function of the training cycles for tied CD phonemes with 32 mixtures, two variance floors (1% and 1‰), and evaluated for the application words test.

Although this phenomenon usually doesn't pose too serious problems regarding the HMM concept, in practical settings it must be dealt with. As the available data are limited, the most effective ways to prevent this phenomenon to happen are: restrict the number of training cycles, set up the variance floors for Gaussian pdf, tie similar means, variances, mixtures, states, and even models, do not construct models with too many Gaussian mixtures, and check the performance of HMM models on the test set. To examine and enumerate the above mentioned methods on the professional database, we decided to accomplish following test: we exposed models that are most prone to the overtraining (with more than 4 mixtures both CI and tied CD phoneme models), to 30 additional training cycles using 2 different variance floors (1% and 1‰ of the overall variance). Again, the training followed the MASPER training scheme for CI and tied CD phoneme models with 1 up to 32 Gaussian mixtures. In fig. 4 there is depicted the WER measure as a function of training cycles for tied CD phonemes with 32 mixtures and both variance floors (1% and 1‰) for the application words test. The same results but for CI phonemes are in fig. 5.

As it can be seen from fig. 4 additional trainings caused the rise of WER for both variance floors, however, WER got stabilized. Bit different situation was observed for CI phonemes were the extra training cycles caused the WER do drop further, but this decrease after 6 or 8 iterations stopped and remained at the same level, for both variance floors. This can be due to a large amount of samples for CI HMM models of phonemes. For the tied CD phonemes the higher sensitivity to the overtraining was observed, which is not a surprise as these models are much more specialized. In both cases the selected values for variance floors provided similar final results. This can be viewed that both floors are still rather low to completely prevent the overtraining given the amount of training samples and the

complexity of models. However, the experiments proved that the original training scheme and the settings on the given database are in eligible ranges and are reasonably insensitive to the overtraining. On the other hand, it was documented that the extensive training may not bring much gain, and it can even deteriorates the accuracy.

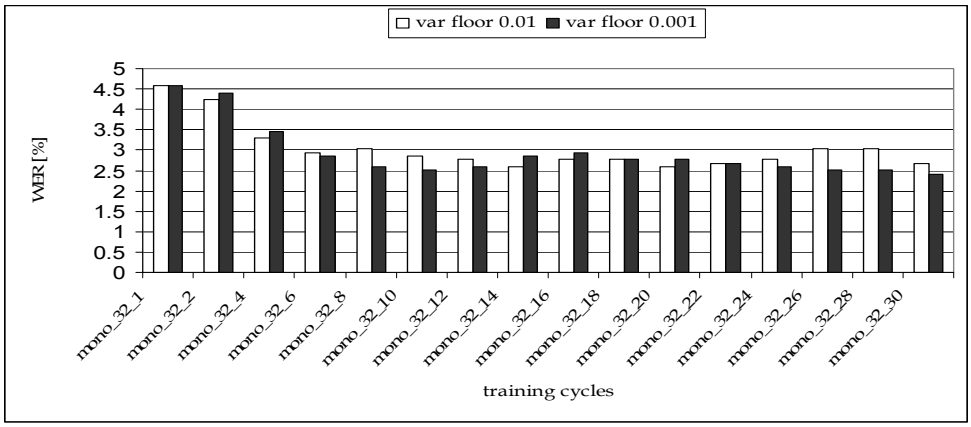


Fig 5. WER as a function of the training cycles for CI phonemes with 32 mixtures, two variance floors (1% and 1‰), and evaluated for the application words test.

4.5 Tying process for context dependent phoneme models

CD phonemes are the most frequently modeled sub-word units in practical large vocabulary systems. However, this would not be possible without tying similar mixtures, states or even whole models of phonetically and/or feature similar units. There are more method to do the tying but probably the most popular are the data based clustering and the decision tree method (Young et al., 2002). The data based clustering tries to merge predefined states by the designer so that the largest cluster reaches the maximum size (it is the largest distance between any two states in the cluster that are to be tied). Process begins with simple states and progress further by merging closest groups. Except the maximal distance, it is advantageous to set up the minimal number of frames per cluster (occupation count) that it will be trained from. On the other hand the decision tree clustering follows different strategy. It uses language questions based on predefined classification of phonemes which are language specific. These questions are asked about the left and right context of a CD phoneme. Based on each question the original cluster is divided according to the answer, positive or negative. Resulting two clusters are better in modeling the data than the original one and thus this separation causes the increase of the modeling probability of the data. Only those clusters are left which caused the highest increase. The process is stopped if the highest increase is less than the predefined constant. To prevent forming clusters with insufficient data the minimal occupation count is set. The greatest advantage of this method lies in the existence of the decision trees that can be preserved and later used to synthesized CD phonemes that were not present in the training phase.

Following the above mentioned discussion we decided to use the decision tree based tying method. However, to get the "optimal" result one must experiment with the proper settings. We were able to classify 40 different groups of phonemes in Slovak, in order to produce the

splitting questions. There is no harm in providing as many as possible questions because all are tested and only those which are really relevant take place. Thus there were two options to set: minimal log likelihood increase and the minimal number of the training frames per a cluster. In the MASPER procedure as well as in the HTK book example these were set to 350 for the minimal log likelihood increase and 100 for the minimal occupation count. As these options depend on each other we tested both, ranging from 50% less to 50% more than the suggested settings. These settings are language specific, and moreover, they depend on the size and type of the database (how many different CD phonemes are there, how many realizations, what are their acoustic dissimilarities, etc.). Increasing both values leads to more robust, but less precise models as well as to lower number of physical states. Of course, their decrease would have the opposite effect. Thus this is the place for experiments and the final tuning for most systems. First, in tab. 1 there are averaged WER and relative improvements for tied CD phoneme models over application words and looped digits tests. Originally suggested values were shifted in their values by: ±50%, ±25%, and 0%.

	-50%	-25%	original	25%	50%
settings	175 log prob. 50 occup.	280 log prob. 80 occup.	350 log prob. 100 occup.	420 log prob 120 occup.	525 log prob. 150 occup.
average WER	2.52	2.53	2.55	2.56	2.52
improvement %	1.12	0.97	0	-0.31	1.07

Table 1. Average WER for tied CD phoneme models for application words and looped digits tests as a function of the minimal log likelihood increase and the minimal occupation count.

Min. occupation count =100	Minimal log likelihood increase		
	100	200	350
average WER	2.49	2.51	2.55
relative improvement %	2.44	1.81	0

Table 2. Averaged WER and relative improvements for tied CD phoneme models over application words and looped digit tests, with the minimal occupation count set to 100.

These results don't clearly show the preferred direction for finding the optimal parameters, i.e. whether to increase the robustness or accuracy. Therefore, as the minimal number for the occupation count the 100 threshold value was accepted. This assures that even for the most complex models with 32 mixtures, there will be on average at lest 3 observations for a single mixture. This seems to be the lowest reasonable number from the robustness point of view. In table 2 there are listed results for several minimal log likelihood increases keeping the occupancy count fixed.

As it can be seen the best result is approximately 100 (minimal log likelihood increase) for the 100 threshold value of the minimal occupancy count. This suggests that making the HMM models more specialized (there are more splits) brought additional 2.4% decrease in the averaged WER comparing to the original settings.

5. Training of HMMs using HTK system

There are many successful ASR systems that were trained by the HTK tools. In our experiments with HTK we decided to use the MASPER (Zgank et. al., 2004) training scheme which is a cross-lingual counterpart of the reference recognition system REFREC 0.96 (Lindberg et al., 2000). Furthermore, both procedures closely cooperate with SPEECHDAT or MOBILDAT databases and handle all relevant aspects of building robust HMM.

5.1 MASPER / REFREC training procedures

The general concept of REFREC 0.96 is based on that one presented in the HTK documentation, however enhanced to serve for multilingual purposes. During the course of the run it produces following models: flat start CI phoneme models with 1 up to 32 Gaussians mixtures, CI models generated in the 2nd run that are initialized on the time aligned data, CD models (with only one Gaussian) and tied CD models with 1 to 32 Gaussian mixtures. On the evaluating part of the training there are 3 small vocabulary tests provided for all models involving: application words, single digits and digits in the loop. REFREC 0.96 uses MFCC speech features with C0, delta and acceleration coefficients that make up a vector with 39 elements. Utterances that are in a way damaged by the existence of GSM noise (%), unintelligible speech (**), mispronunciation (*) and truncation (~) are removed. For speaker produced noises (spk) and hesitations (fil) separate models are used while the other markers are ignored. The training starts with the flat start initialization and a variance floor is also calculated. This initialization and first cycles of embedded training are executed over the phonetically reach utterances with the creation of SIL (general background) and SP (short pause T model) models. Then the Viterbi forced alignment utilizing multiple pronunciations is done as well as the acoustically "suspicious" recordings are removed. Next, the process goes on in cycles of two training passes followed by a mixture incrementing stage by the factor of 2 as far as 32 mixtures are reached. These final models are used to do the time alignment over all utterances so that the Viterbi initialization and the single model training of CI phonemes can be done in the second stage of the training. CI phonemes derived in the second stage with 1 Gaussian mixture are used for cloning CD phonemes. In this way more accurate models of CI phonemes with 1 mixture are obtained than those trained in the first run. These single-mixture models are further used by the cloning and tying procedures in the construction of tied CD models. After the cloning, the CD models are trained in 2 cycles of the embedded training and then tied which is done by the decision tree algorithm. After the tying, the gradual process of two re-estimations passes interleaved by mixtures incrementing stage is repeated up to the 32 mixtures are reached. Finally, CI phoneme models from the second run are enhanced and trained in cycles using the embedded training up to 32 mixtures.

To enable an effective design of the multilingual and cross-lingual ASR systems some further modification must have been done to the REFREC 0.96, which resulted in the MASPER procedure. These changes are as follows: cepstral mean subtraction, modifications to the parameters of tree based clustering, and the production of the training statistics.

5.2 Proposed modification to the MASPER training scheme

As we can see, REFREC 0.96 or MASPER are advanced procedures for building mono, multi or cross-lingual HMM models for large vocabulary ASR systems. However, we discovered

some deficiency of these schemes in handling the training data, i.e. the removal of all utterances partially contaminated with truncated, mispronounced and unintelligible speech even though the rest of the recording may be usable. Thus in the following the modification to the MASPER procedure aiming to model the damaged parts of the speech while preserving useful information will be presented and evaluated.

Let's start with some statistic regarding the portion of damaged and removed speech. After the rejection of corrupted speech files there were in total 955611 instances of all phonemes. The same analysis applied just to the rejected speech files has discovered further 89018 realizations of usable phonemes, which amounts to 9.32% of all appropriate phoneme instances. More detailed statistic regarding the recordings, CI and CD phonemes used by MASPER and modified MASPER procedures on MOBILDAT –SK is summarized in table 3.

Statistics of the database	MASPER	modified MASPER	Absolute increase	Relative increase
recordings	40861	43957	3096	7,58%
CI phonemes	51	51	0	0%
CD phonemes	10567	10630	63	0,60%
instances of CI phonemes	955611	1044629	89018	9,32%
average number of instances per a CD phoneme	~90.4	~98.27	~7.84	~8.7%

Table 3. Statistics of CI and CD phonemes contained in MOBILDAT SK that are utilized by MASPER and modified MASPER procedures.

To be more specific, in fig. 6 there are depicted realizations of Slovak phonemes used by MASPER and modified MASPER procedures. The modified MASPER procedure preserves eligible data from the damaged recordings by using a unified model of garbled speech that acts as a patch over corrupted words. These words are not expanded to the sequence of phonemes, but instead, they are mapped to a new unified model of garbled speech, the so called BH model (black hole- attract everything). Then the rest of a sentence can be processed in the same way as in the MASPER procedure. The new model is added to the phoneme list (context independent and serves as a word break) and is trained together with other models. However, its structure must be more complex as it should map words of variable lengths spoken by various speakers in different environments.

Following this discussion about the need for a complex model of garbled words while having limited data, there are two related problems to solve: the complexity of such a model and its enhancement stages. From the modelling point of view the ergodic model with as many states as there are speech units would be the best, however, it would extremely increase the amount of parameters to estimate. As it was expected there must have been tested more options ranging from the simplest structures like a single state model to models with 5 states (emitting). At the end of the training it was assumed that this model should be ergodic, just to get the full modelling capacity, which is not strictly related to the time evolution of the speech.

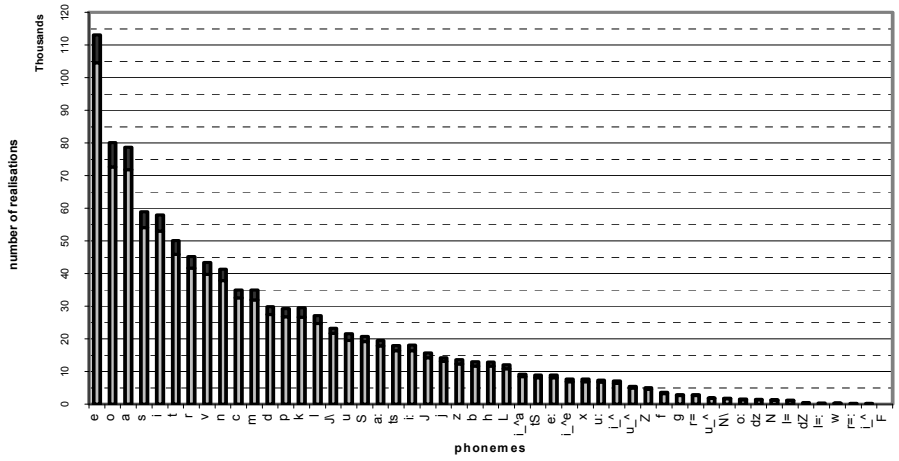


Fig. 6. Number of instances for each phoneme in MOBILDAT-SK processed by MASPER and modified MASPER (black tops)

Initial structure	Method of enhancement	Final structure
Stricly left-right, start and end in the first and last emitting states	Addition of all backward connection, no T model	ergodic
ergodic	no	ergodic
Left-right, with all forward connections	Addition of backward connections	ergodic
Left right, with all forward connections	Addition of backward connections, single model training of BH in the second stage	ergodic

Table 4. Tested initial structures, methods of enhancement and final structures for BH model.

Furthermore, there were more strategies how to enhance the original BH model so that in a final stage it would be ergodic. In table 4 there are summed up all the tested possibilities of original and final structures of the BH model as well as the applied modifications during the training. All BH models and updating methods were evaluated by standard tests used in MASPER (single digits, looped digits and application words). From all these experiments the 5 state BH model with initial structure allowing all forward connections showed to be slightly better then remaining ones. In fig. 7, there are shown results for CI and CD phoneme models and the looped digits test (perplexity= 10).

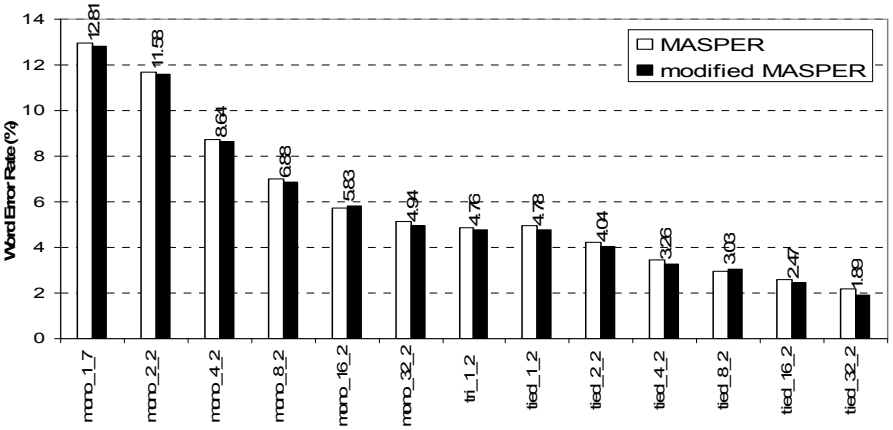


Fig. 7. Word error rates for the looped digits test and different CI and CD phoneme models, using MASPER and modified MASPER training methods.

Tests and models	SVIP AP		SVIP DIG		SVWL	
	Orig.	Mod.	Orig.	Mod.	Orig.	Mod
mono_1_7	10.03	10.21	4.13	5.05	12.96	12.81
mono_2_2	10.38	10.81	4.13	5.05	11.67	11.58
mono_4_2	9.69	9.17	5.05	5.05	8.73	8.64
mono_8_2	7.44	7.18	2.75	2.75	6.98	6.88
mono_16_2	4.93	5.02	1.83	1.83	5.71	5.83
mono_32_2	4.24	4.15	2.29	2.29	5.13	4.94
tri_1_2	1.99	1.99	1.38	1.38	4.86	4.76
tied_1_2	1.82	1.82	1.38	1.38	4.95	4.78
tied_2_2	1.64	1.56	1.38	1.38	4.21	4.04
tied_4_2	1.47	1.3	0.92	0.92	3.45	3.26
tied_8_2	1.56	1.21	1.38	1.38	2.96	3.03
tied_16_2	1.56	1.47	0.92	0.92	2.59	2.47
tied_32_2	1.56	1.3	0.92	0.92	2.16	1.89

Table 5. Word error rates for different CI and CD phoneme models using MASPER and modified MASPER and 3 tests: application words, single digits and looped digits.

Furthermore, in table 5 there are listed results for all tests and models. As it can be seen this modification brought on average improved results both for CI as well as CD phonemes; for tied CD models almost a 5% improvement was achieved. The main drawback of this approach is however, the longer training process which added 11% of extra computation.

6. Training of HMMs using SPHINX system

The HMM training tool whose output can be directly used with the SPHNIX decoders is called SphinxTrain. Thus let us in the following list some basic information about it.

6.1 SphinxTrain

It is an independent application which trains HMM models containing SphinxTrain tools and control scripts which actually govern the training (Scriptman, 2000).
The training process gradually undergoes following stages: verification of the input data, feature extraction (MFCC and PLP static coefficients plus standard auxiliary features are supported), vector quantization (used only for discrete models), initialization of HMM models (flat start), training of CI HMM models (no time alignment is supported, thus only the embedded training is available) with gradually incremented mixtures, automatic generations of language questions and building decision trees that are needed in the tying process for CD phonemes, pruning of the decision trees and tying similar states, training of tied CD phoneme models with gradually augmented number of mixtures.
To sum it up, as it can be seen, besides others there are several differences to the MASPER scheme: there are only two stages of the training, no single model training in the second stage, no alignment for multiple pronunciations, no sp model and no modification to the SIL structure (backward connection), number of training cycles is not fixed but is controlled by the convergence factor, non existence of predefined classes of phonemes, all models have the same structure, etc.

6.2 Proposed modifications to SphinxTrain procedure

Modifications that were done and tested can be divided into two categories: formal which involved conversions of data types and structures and the functional which affected the training process or the structure of HMM models. Functional changes and settings include following issues: selection of modelled non-speech events (so called fillers), set proper recordings for initialization and training phases, number of states per model, number of Gaussians mixtures, number of tied states for CD models, how to handle the problem of missing short pause model especially for the spelled items, etc. Thus in the following those issues will be addressed and tested on MOBILDAT- SK.
Unlike HTK a filer dictionary has to be constructed containing all non-speech events that should be modelled. This part of the design process is important as the non-speech models would share the same structure as the speech units (limitation of SphinxTrain). As a consequence, these models won't be so flexible (no backward connections etc.) thus they should rather be specialized to particular events. However, only few of the events were marked in the MOBILDAT database. Therefore we decided to use a general background model that includes all silences either within sentences or at the beginning and end of each recording. Two other models were added to the filer dictionary, one marking the noises produced by the speaker (spk) and a hesitation model (fil), as they were separately marked in the database.
Regarding the problem of usable recordings for the training process we decided to remove all items that contained damaged speech as no appropriate BH (garbled speech) model could be constructed following the tests with the modified MASPER procedure.
As most of the systems use multiple Gaussian mixtures ranging from 1 to 32 we decided to construct all of them for CI and tied CD phonemes and test their performance in all other experiments.
In order to find "optimal" setting for our system we performed tests regarding the number of states per model (either 3 or 5), number of tied CD phonemes that ranged from 2000 to

18000 and different training scenarios for spelled items to eliminate the missing short pause model. There were altogether 4 training scenarios for the spelled items. The original one ignored the problem and did not consider any background model between phonemes, even despite that there is a high probability of silence when the phonemes are spelled. The second one removed these recordings, just to avoid any incorrect transcription to be involved in the training. The 3rd scenario blindly assumed that there must be high a priory probability of pauses and thus inserted the silence model between all spelled items, and the last scenario uses the forced alignment tool from the SPHINX decoder (this was not included in the earlier versions of SphinxTrain scripts). This alignment does not produce any time marks, does not perform selection between multiple realizations of words, it just decides to place models from the filer dictionary between words. We applied this tool to all recordings (unmarked silences may occur also in other recordings) in the early stage of the training using CI models with 16 Gaussian mixtures. Tests were performed on the SPHINX 4 decoder (Walker et al., 2004) as it supports finite state grammar (JFSG) and the evaluation was done on the HTK system so that very similar conditions were maintained.

3 states	CD models, number of Gaussian mixtures			Average accuracy over different CD models for fix number of tied states
Number of tied states	8	16	32	
2000	97.67	97.87	98.14	97.89
5000	97.58	98.05	98.19	97.94
9000	97.67	97.88	97.96	97.83
12000	97.45	97.79	97.91	97.72
15000	97.5	97.88	98.16	97.84
180000	97.51	97.8	98.11	97.80

Table 6. The accuracy of CD HMM models with 3 states per a model and various numbers of mixtures for different number of tied states.

In table 6, results for 3 state models and different number of tied states for CD models are listed, the same results but for 5 state models are shown in table 7. As it can be seen the best results on average were obtained for 5000 tied states in the case of 3 state models, however, the differences in the widely tested ranges were interestingly small. For the test with 5 state models the best number was 18000 which is relatively high, but again the mutual differences were negligible. For 5 state models there were 5/3 times more different states thus it is natural these should be modelled with higher number of physical states. Comparing tables 6 and 7 it can be seen that on average the models with higher number of states (5) provided slightly better results, the average relative improvement is 0.21%. This is of no wander as they may have better modelling capabilities. On the other hand, there are more free parameters to be estimated which may produce more accurate but not robust enough models which in this test was apparently not the case. Finally, the 4 training scenarios were compared and the results are listed in table 8.

5 states	Number of Gaussian mixtures for CD models			Average accuracy over different CD models for fix number of tied states
Number of tied states	8	16	32	
2000	97.8	98.13	98.33	98.08
5000	97.85	98.28	98.11	98.08
9000	97.98	97.85	98.07	97.96
12000	97.84	97.98	98.07	97.96
15000	97.93	98.12	98.21	98.08
180000	97.97	98.17	98.3	98.14

Table 7. The accuracy of CD HMM models with 5 states per a model and various numbers of mixtures for different number of tied states.

Models	Scenarios			
	Original	Spelled items removed	SIL inserted into spelled items	Forced alignment
CI -4 Gaussians	94.49	94.85	95.02	95.26
CI -8 Gaussians	95.19	95.23	95.49	95.58
CI -16 Gaussians	95.96	96.08	96.34	95.96
CI -32 Gaussians	96.24	96.48	96.57	96.43
CD -4 Gaussians	97.31	97.62	97.63	97.46
CD -8 Gaussians	97.67	97.63	97.69	97.7
CD-16 Gaussians	97.88	98.15	97.82	97.7
CD-32 Gaussians	97.96	98.25	98.12	98.25
Average over models	96.58	96.78	96.83	96.82

Table 8. The accuracy of different tied CD and CI HMM models for 4 training scenarios.

As it can be seen the worst case is the original training scheme. On the other hand, the best results on average are provided by the "blind" insertion of SIL models between spelled phonemes. This suggests that there was really high incidence of pauses and the forced alignment was not 100% successful in detecting them.

7. Conclusion

Even though there are many new and improved techniques for HMM modelling of speech units and different feature extraction methods, still they are usually restricted to laboratories or specific conditions. Thus most of the systems designed for large vocabulary and speaker independent tasks use the "classical" HMM modelling by CDHMM with multiple Gaussian mixtures and tied CD models of phonemes.

In this chapter the construction of robust and accurate HMM models for Slovak was presented using 2 of the most popular systems and the training schemes. These were tested on the professional MOBILDAT –SK database that poses more adverse environment. In practical examples issues like: feature extraction methods, structures of models, modelled units, overtraining, and the number of tied states were discussed and tested. Some of the here suggested adjustments were successfully used while building Slovak ASR (Juhar, et al., 2006). Then the advanced training scheme for building mono, cross and multilingual ASR systems (MASPER based on HTK) that incorporates all the relevant training aspects was presented. Next, its practical modification aiming to increase the amount of usable training data by the BH model was suggested and successfully tested. Further, the training method utilizing the SPHINX system (SphinxTrain) was discussed and in the real conditions its "optimal" settings were found for the case of MOBILDAT –SK database. Finally, useful modifications for eliminating the problem of the missing short pause model in the case of spelled items were suggested and successfully tested. To compare both systems the best settings (modifications) for MASPER and SphinxTrain were used. Averaged word error rates were calculated over all models using application words and looped digits tests. Achieved results in table 9 are also listed separately for CI and tied CD models with 4, 8, 16, and 32 mixtures, the memory consumption and the training times are also included.

	Average WER		Memory consumption [MB]		Training time [hours]	
	Mod. Masper	Mod. SphinxTrain	Mod. Masper	Mod. SphinxTrain	Mod. Masper	Mod. SphinxTrain
All models	4.28	6.92	95	177	25h 8min	20h 58min
CD models	2.38	3.66	91.7	174	8h 48min	8h 53min
CI models	6.18	10.17	3.29	3.14	16h 20min	12h 5min

Table 9. Overall comparison of modified MASPER and modified SphinxTrain training procedures in terms of the accuracy, memory consumption and the training times. Word error rates were calculated for models with 4, 8, 16 and 32 mixtures.

As it can be seen, SphinxTrain scored on average worse by 38% in terms of WER evaluated over all models and executed tests. Its models occupy 86% more memory and are stored in 4 different files; however the training time for SphinxTrain is 20% shorter. If the averaged results are looked at separately, i.e. looped digits and application words tests, more precise image is obtained, see table 10. Models trained the by SphinxTrain procedure showed on average better results for the looped digits test than those on MASPER, on the other hand, SphinxTrain models were much less successful in the task of application words (perplexity 30), which contain richer set of CI and CD phonemes. That may suggest the tying and the training processes were not so effective. In the case of the MASPER procedure CI models were taken from the second run of the training so they were initialized and trained (only 7 initial cycles) on the time aligned recordings from the first run and thus they converged faster at the beginning. This fact is also used in the tying process of CD phonemes where models with only 1 mixture are taken in account. Finally, it should be noted that different decoders had to be used (HVite and SPHINX 4) during the evaluation. Despite the fact the same grammars and test sets were used, these decoders still have their specific settings

which may not have been optimized for particular tests, e. g. the insertion probability of fillers (SPHINX 4), pruning options, etc. Thus the results except the training phase partially reflect the decoding process as well, which was not the primary aim.

	looped digits		aplication words	
	Mod. Masper	Mod. SphinxTrain	Mod. Masper	Mod. SphinxTrain
All models	3.93	3.17	4.63	10.66
CD models	2.61	2.15	2.15	5.17
CI models	5.26	4.19	7.10	16.16

Table 10. Comparison of modified MASPER and modified SphinxTrain training procedures in terms of the accuracy, evaluated separately for looped digits and application words tests. Word error rates were calculated for models with 4, 8, 16 and 32 mixtures.

8. References

Baum, L. & Eagon, J. (1967). An inequality with applications to statistical estimation for probabilities functions of a Markov process and to models for ecology. *Bull AMS*, Vol. 73, pp. 360-363

Bonafonte, A.; Vidal, J. & Nogueiras, A. (1996). Duration modeling with expanded HMM applied to speech recognition, *Proceedings of ICSLP 96,* Vol. 2, pp. 1097-1100, ISBN: 0-7803-3555-4. Philadelphia, USA, October, 1996

Casar, M. & Fonllosa, J. (2007). Double layer architectures for automatic speech recognition using HMM, in book *Robust Speech recognition and understanding*, I-Tech education and publishing, ISBN 978-3-902613-08-0, Croatia, Jun, 2007

Darjaa, S.; Rusko, M. & Trnka, M. (2006). MobilDat-SK - a Mobile Telephone Extension to the SpeechDat-E SK Telephone Speech Database in Slovak, *Proceedings of the 11-th International Conference Speech and Computer (SPECOM'2006)*, pp. 449-454, St. Petersburg 2006, Russia

Hermasky, H. & Morgan, N. (1994). RASTA Processing of Speech, *IEEE Transactions on Speech and Audio Processing*, Vol. 2, No. 4, Oct. 1994

Hönig, F; Stemmer, G.; Hacker, Ch. & Brugnara, F. (2005). Revising Perceptual linear Prediction (PLP), *Proceedings of INTERSPEECH 2005,* pp. 2997-3000, Lisbon, Portugal, Sept., 2005

Huang, X.; Ariki, Y. & Jack, M. (1990). *Hidden Markov Models for Speech Recognition*, Edinburg university press, 1990

Jiang, H. & Li X. (2007) A general approximation-optimization approach to large margin estimation of HMMs, in book *Robust Speech recognition and understanding*, I-Tech education and publishing, ISBN 978-3-902613-08-0, Croatia, Jun, 2007

Juhar, J.; Ondas, S.; Cizmar, A; Rusko, M.; Rozinaj, G. & Jarina, R. (2006). Galaxy/VoiceXML Based Spoken Slovak Dialogue System to Access the Internet. *Proceedings of ECAI 2006 Workshop on Language-Enabled Educational Technology and Development and Evaluation of Robust Spoken Dialogue Systems*, pp.34-37, Riva del Garda, Italy, August, 2006

Kosaka, T.; Katoh, M & Kohda, M. (2007). Discrete-mixture HMMs- based approach for noisy speech recognition, in book *Robust Speech recognition and understanding*, I-Tech education and publishing, ISBN 978-3-902613-08-0, Croatia, Jun, 2007

Lee, K.; Hon, H. & Reddy, R. (1990). An overview of the SPHINX speech recognition system, *IEEE transactions on acoustics speech and signal processing*, Vol. 38, No. 1, Jan., 1990

Levinson, E. (1986). Continuously variable duration hidden Markov models for automatic speech recognition. *Computer Speech and Language*, Vol. 1, pp. 29-45, March, 1986

Lindberg, B.; Johansen, F.; Warakagoda, N.; Lehtinen, G; Kacic, Z; Zgang, A; Elenius, K. & Salvi G. (2000). A Noise Robust Multilingual Reference Recognizer Based on SpeechDat(II), *Proceedings of ICSLP 2000*, Beijing, China, October 2000

Nadeu, C. & Macho, D. (2001). Time and Frequency Filtering of Filter-Bank energies for robust HMM speech recognition, *Speech Communication.* Vol. 34, Elsevier, 2001

Nouza, J.; Zdansky, J.; David, P.; Cerva, P.; Kolorenc, J. & Nejedlova, D. (2005). Fully Automated System for Czech Spoken Broadcast Transcription with Very Large (300K+) Lexicon. *Proceedings of Interspeech 2005*, pp. 1681-1684, ISSN 1018-4074, Lisboa, Portugal, September, 2005,

Poggio, T. & Girosi, F. (1990). Networks for approximation and learning, *Proceedings of the IEEE* 78, pp. 1481-1497

Rabiner, L. & Juan, B. (1993). *Fundamentals of speech recognition*, ISBN 0-13-015157-2, Prentice Hall PTR, New Yersey.

Scriptman (2000). Online documentation of the SphinxTrain training scripts, location: http://www.speech.cs.cmu.edu/sphinxman/scriptman1.html, last modification Nov. 2000

W. Walker, P. Lamere, P. Kwok (2004). Sphinx-4: A Flexible Open Source Framework for Speech Recognition, Report, location: http://research.sun.com/techrep /2004/smli_tr-2004-139.pdf

Young, S.; Evermann, G.; Hain, T.; Kershaw, D.; Moore, G.; Odell, J.; Ollason, D.; Povey, Dam.; Valtchev, V. & Woodland, P. (2002). The HTK Book V.3.2.1, Cambridge University Engineering Department, Dec. 2002

Zgank, A.; Kacic, Z.; Diehel, F.; Vicsi, K.; Szaszak, G.; Juhar, J.; Lihan, S. (2004). The Cost 278 MASPER initiative- Crosslingual Speech Recognition with Large Telephone Databases, *Proceedings of Language Resources and Evaluation (LREC)*, pp. 2107-2110, Lisbon, 2004

Language modelling

10

Statistical Language Modeling for Automatic Speech Recognition of Agglutinative Languages

Ebru Arısoy[1], Mikko Kurimo[2], Murat Saraçlar[1], Teemu Hirsimäki[2], Janne Pylkkönen[2], Tanel Alumäe[3] and Haşim Sak[1]
[1]*Boğaziçi University,*
[2]*Helsinki University of Technology,*
[3]*Tallinn University of Technology,*
[1]*Turkey*
[2]*Finland*
[3]*Estonia*

1. Introduction

Automatic Speech Recognition (ASR) systems utilize statistical acoustic and language models to find the most probable word sequence when the speech signal is given. Hidden Markov Models (HMMs) are used as acoustic models and language model probabilities are approximated using *n*-grams where the probability of a word is conditioned on *n-1* previous words. The *n*-gram probabilities are estimated by Maximum Likelihood Estimation. One of the problems in *n*-gram language modeling is the data sparseness that results in non-robust probability estimates especially for rare and unseen *n*-grams. Therefore, smoothing is applied to produce better estimates for these *n*-grams.

The traditional *n*-gram word language models are commonly used in state-of-the-art Large Vocabulary Continuous Speech Recognition (LVSCR) systems. These systems result in reasonable recognition performances for languages such as English and French. For instance, broadcast news (BN) in English can now be recognized with about ten percent word error rate (WER) (NIST, 2000) which results in mostly quite understandable text. Some rare and new words may be missing in the vocabulary but the result has proven to be sufficient for many important applications, such as browsing and retrieval of recorded speech and information retrieval from the speech (Garofolo et al., 2000). However, LVCSR attempts with similar systems in agglutinative languages, such as Finnish, Estonian, Hungarian and Turkish so far have not resulted in comparable performance to the English systems. The main reason of this performance deterioration in those languages is their rich morphological structure. In agglutinative languages, words are formed mainly by concatenation of several suffixes to the roots and together with compounding and inflections this leads to millions of different, but still frequent word forms. Therefore, it is practically impossible to build a word-based vocabulary for speech recognition in agglutinative languages that would cover all the relevant words. If words are used as language modeling units, there will be many out-of-vocabulary (OOV) words due to using limited vocabulary sizes in ASR systems. It was shown that with an optimized 60K lexicon

the OOV rate is less than 1% for North American Business news (Rosenfeld, 1995). Highly inflectional and agglutinative languages suffer from high number of OOV words with similar size vocabularies. In our Turkish BN transcription system, the OOV rate is 9.3% for a 50K lexicon. For other agglutinative languages like Finnish and Estonian, OOV rates are around 15% for a 69K lexicon (Hirsimäki et al., 2006) and 10% for a 60K lexicon respectively and 8.27% for Czech, a highly inflectional language, with a 60K lexicon (Podvesky & Machek, 2005). As a rule of thumb an OOV word brings up on average 1.5 recognition errors (Hetherington, 1995). Therefore solving the OOV problem is crucial for obtaining better accuracies in the ASR of agglutinative languages. OOV rate can be decreased to an extent by increasing the vocabulary size. However, even doubling the vocabulary is not a sufficient solution, because a vocabulary twice as large (120K) would only reduce the OOV rate to 6% in Estonian and 4.6% in Turkish. In Finnish even a 500K vocabulary of the most common words still gives 5.4% OOV in the language model training material. In addition, huge lexicon sizes may result in confusion of acoustically similar words and require a huge amount of text data for robust language model estimates. Therefore, sub-words are proposed as language modeling units to alleviate the OOV and data sparseness problems that plague systems based on word-based recognition units in agglutinative languages.

In sub-word-based ASR; (i) words are decomposed into meaningful units in terms of speech recognition, (ii) these units are used as vocabulary items in *n*-gram language models, (iii) decoding is performed with these *n*-gram models and sub-word sequences are obtained, (iv) word-like units are generated from sub-word sequences as the final ASR output.

In this chapter, we mainly focus on the decomposition of words into sub-words for LVCSR of agglutinative languages. Due to inflections, ambiguity and other phenomena, it is not trivial to automatically split the words into meaningful parts. Therefore, this splitting can be performed by using rule-based morphological analyzers or by some statistical techniques. The sub-words learned with morphological analyzers and statistical techniques are called grammatical and statistical sub-words respectively. Morphemes and stem-endings can be used as the grammatical sub-words. The statistical sub-word approach presented in this chapter relies on a data-driven algorithm called Morfessor Baseline (Creutz & Lagus, 2002; Creutz & Lagus, 2005) which is a language independent unsupervised machine learning method to find morpheme-like units (called *statistical morphs*) from a large text corpus.

After generating the sub-word units, *n*-gram models are trained with sub-words similarly as if the language modeling units were words. In order to facilitate converting sub-word sequences into word sequences after decoding, word break symbols can be added as additional units or special markers can be attached to non-initial sub-words in language modeling. ASR systems that successfully utilize the n-gram language models trained for sub-word units are used in the decoding task. Finally, word-like ASR output is obtained from sub-word sequences by concatenating the sub-words between consecutive word breaks or by gluing marked non-initial sub-words to initial ones. The performance of words and sub-words are evaluated for three agglutinative languages, Finnish, Estonian and Turkish.

This chapter is organized as follow: In Section 2, our statistical language modeling approaches are explained in detail. Section 3 contains the experimental setup for each language. Experimental results are given in Section 4. Finally, this chapter is concluded with a detailed comparison of the proposed approaches for agglutinative languages.

2. Statistical language modeling approaches

The morphological productivity of agglutinative languages makes it difficult to construct robust and effective word-based language models. With a dictionary size of a few hundred thousand words, we can still have OOV words, which are constructed through legal morphological rules. Therefore, in addition to words, sub-word units are utilized in LVCSR tasks for Finnish, Estonian and Turkish. Fig. 1 shows a phrase in each language segmented into proposed grammatical and statistical sub-word units. The details of these units will be explained thoroughly in this section.

Finnish example: Words: pekingissä vieraileville suomalaisille kansanedustajille
Grammatical sub-words:
Morphemes: pekingi ssä # vieraile v i lle # suomalais i lle # kansa n edusta j i lle
Statistical sub-words:
Morphs: peking issä # vieraile ville # suomalaisille # kansanedustaj ille

Estonian example: Words: teede ja sideministeerium on teinud ettepaneku
Grammatical sub-words:
Morphemes: tee de # ja # side ministeerium # on # tei nud # ette paneku
Statistical sub-words:
Morphs: teede # ja # sideministeerium # on # te i nud # ettepaneku

Turkish example: Words: tüketici derneklerinin öncülüğünde
Grammatical sub-words:
Morphemes: tüketici # dernek leri nin # öncü lüğ ü nde
Stem-endings: tüketici # dernek lerinin # öncü lüğünde
Statistical sub-words:
Morphs: tüketici # dernek lerinin # öncü lüğü nde

Fig. 1. Finnish, Estonian and Turkish phrases segmented into statistical and grammatical sub-words

2.1 Word-based model

Using words as recognition units is a classical approach employed in most state-of-the-art recognition systems. The word model has the advantage of having longer recognition units which results in better acoustic discrimination among vocabulary items. However the vocabulary growth for words is almost unlimited for agglutinative languages and this leads to high number of OOV words with moderate size vocabularies in ASR systems. It has been reported that the same size text corpora (40M words) result in less than 200K word types for English and 1.8M and 1.5M word types for Finnish and Estonian respectively (Creutz et al., 2007a). The number of word types is 735K for the same size Turkish corpus.

2.2 Sub-word-based models

Large number of OOV words and data sparseness are the main drawbacks of the word-based language modeling units in ASR of agglutinative and highly inflectional languages. Therefore, several sub-word units were explored for those languages to handle these drawbacks. Naturally, there are many ways to split the words into smaller units to reduce a lexicon to a tractable size. However, for a sub-word lexicon suitable for language modeling applications such as speech recognition, several properties are desirable:

i. The size of the lexicon should be small enough that the n-gram modeling becomes more feasible than the conventional word based modeling.
ii. The coverage of the target language by words that can be built by concatenating the units should be high enough to avoid the OOV problem.
iii. The units should be somehow meaningful, so that the previously observed units can help in predicting the next one.
iv. For speech recognition one should be able to determine the pronunciation for each unit.

A common approach to find the sub-word units is to program the language-dependent grammatical rules into a morphological analyzer and utilize it to split the text corpus into morphemes. As an alternative approach, sub-word units that meet the above desirable properties can be learned with unsupervised machine learning algorithms. In this section, we investigated both of the approaches.

2.2.1 Grammatical sub-words; morphemes and stem-endings

Using morphemes and stem-endings as recognition units is becoming a common approach in statistical language modeling of morphologically rich languages. Morphemes were utilized as language modeling units in agglutinative languages such as Finnish (Hirsimäki et al., 2006), Estonian (Alumäe, 2005) and Turkish (Hacioglu et al., 2003) as well as in Czech (Byrne et al., 2001) which is a highly inflectional language. Merged morphemes were proposed instead of word phrases for Korean (Kwon and Park, 2003). In Kanevsky and Roukos (1998) stem-ending based modeling was proposed for agglutinative languages and it is used in ASR of Turkish with both surface form (Mengüşoğlu & Deroo, 2001; Bayer et al., 2006) and lexical form (Arısoy et al., 2007) representations of endings. In addition, a unified model using both words, stem-endings and morphemes was proposed for Turkish (Arısoy et al., 2006).

A morphological analyzer is required to obtain morphemes, stems and endings. However, due to the handcrafted rules, morphological analyzers may suffer from an OOV problem, since in addition to morphotactic and morphophonemic rules, a limited root vocabulary is also compiled in the morphological analyzer. For instance, a Turkish morphological parser (Sak et al., 2008) with 54,267 roots can analyze 96.7% of the word tokens and 52.2% of the word types in a text corpus of 212M words with 2.2M unique words. An example output from this parser for Turkish word *alın* is given in Fig. 2. The English glosses are given in parenthesis for convenience. The inflectional morphemes start with a + sign and the derivational morphemes start with a - sign. Part-of-speech tags are attached to roots in brackets and lexical morphemes are followed by nominal and verbal morphological features in brackets. As was shown in Fig. 2, the morphological parsing of a word may result in multiple interpretations of that word due to complex morphology. This ambiguity can be resolved using morphological disambiguation tools for Turkish (Sak et al., 2007).

alın[Noun]+[A3sg]+[Pnon]+[Nom] *(forehead)*
al[Noun]+[A3sg]+Hn[P2sg]+[Nom] *(your red)*
al[Adj]-[Noun]+[A3sg]+Hn[P2sg]+[Nom] *(your red)*
al[Noun]+[A3sg]+[Pnon]+NHn[Gen] *(of red)*
al[Adj]-[Noun]+[A3sg]+[Pnon]+NHn[Gen] *(of red)*
alın[Verb]+[Pos]+[Imp]+[A2sg] *((you) be offended)*
al[Verb]+[Pos]+[Imp]+YHn[A2pl] *((you) take)*
al[Verb]-Hn[Verb+Pass]+[Pos]+[Imp]+[A2sg] *((you) be taken)*

Fig. 2. Output of the Turkish morphological parser (Sak et al., 2008) with English glosses.

To obtain a morpheme-based language model, all the words in the training text corpus are decomposed into their morphemes using a morphological analyzer. Then a morphological disambiguation tool is required to choose the correct analysis among all the possible candidates using the given context. In Arısoy et al. (2007) the parse with the minimum number of morphemes is chosen as the correct parse since the output of the morphological parser used in the experiments was not compatible with the available disambiguation tools. Also, a morphophonemic transducer is required to obtain the surface form representations of the morphemes if the morphological parser output is in the lexical form as in Fig. 2.
In statistical language modeling, there is a trade-off between using short and long units. When grammatical morphemes are used for language modeling, there can be some problems related to the pronunciations of very short inflection-type units. Stem-endings are a compromise between words and morphemes. They provide better OOV rate than words, and they lead to more robust language models than morphemes which require longer *n*-grams. The stems and endings are also obtained from the morphological analyzer. Endings are generated by concatenating the consecutive morphemes.
Even though morphemes and stem-endings are logical sub-word choices in ASR, they require some language dependent tools such as morphological analyzers and disambiguators. The lack of successful morphological disambiguation tools may result in ambiguous splits and the limited root vocabulary compiled in the morphological parsers may result in poor coverage, especially for many names and foreign words which mostly occur in news texts.
One way to extend the rule-based grammatical morpheme analysis to new words that inevitably occur in large corpora, is to split the words using a similar maximum likelihood word segmentation by Viterbi search as in the unsupervised word segmentation (statistical morphs in section 2.2.2), but here using the lexicon of grammatical morphs. This drops the OOV rate significantly and helps to choose the segmentation using the most common units where alternative morphological segmentations are available.

2.2.2 Statistical sub-words; morphs

Statistical morphs are morpheme-like units obtained by a data driven approach based on the Minimum Description Length (MDL) principle which learns a sub-word lexicon in an unsupervised manner from a training lexicon of words (Creutz & Lagus, 2005). The main idea is to find an optimal encoding of the data with a concise lexicon and a concise representation of the corpus.
In this chapter, we have adopted a similar approach as Hirsimäki et al. (2006). The Morfessor Baseline algorithm (Creutz & Lagus, 2005) is used to automatically segment the word types seen in the training text corpus. In the Morfessor Baseline algorithm the minimized cost is the coding length of the lexicon and the words in the corpus represented by the units of the lexicon. This MDL based cost function is especially appealing, because it tends to give units that are both as frequent and as long as possible to suit well for both training the language models and also decoding of the speech. Full coverage of the language is also guaranteed by splitting the rare words into very short units, even to single phonemes if necessary. For language models utilized in speech recognition, the lexicon of the statistical morphs can be further reduced by omitting the rare words from the input of the Morfessor Baseline algorithm. This operation does not reduce the coverage of the lexicon, because it just splits the rare words then into smaller units, but the smaller lexicon may offer a remarkable speed up of the recognition. The pronunciation of, especially, the short units may be ambiguous and may cause severe problems in languages like English, in which the

pronunciations can not be adequately determined from the orthography. In most agglutinative languages, such as Finnish, Estonian and Turkish, rather simple letter-to-phoneme rules are, however, sufficient for most cases.

The steps in the process of estimating a language model based on statistical morphs from a text corpus is shown in Fig. 3. First word types are extracted from a text corpus. Rare words are removed from the word types by setting a frequency cut-off. Elimination of the rare words is required to reduce the morph lexicon size. Then the remaining word types are passed through a word splitting transformation. Based on the learned morph lexicon, the best split for each word is determined by performing a Viterbi search using within-word n-gram probabilities of the units. At this point the word break symbols, # (See Fig. 1), are added between each word in order to incorporate that information in the statistical language models, as well. We prefer to use additional word break symbols in morph-based language modeling since unlike stems, a statistical morph can occur at any position in a word and marking the non-initial morphs increases the vocabulary size.

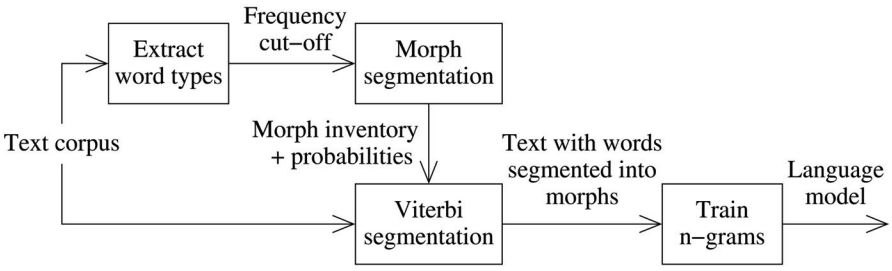


Fig. 3. The steps in the process of estimating a language model based on statistical morphs from a text corpus (Hirsimäki et al., 2006).

The statistical morph model has several advantages over the rule-based grammatical morphemes, e.g. that no hand-crafted rules are needed and all words can be processed, even the foreign ones. Even if good grammatical morphemes are available for Finnish, it has been shown that the language modeling results by the statistical morphs seem to be at least as good, if not better (Hirsimäki et al., 2006; Creutz et al., 2007b).

3. Experimental setups

Statistical and grammatical units are used as the sub-word approaches in the Finnish, Estonian and Turkish LVCSR experiments. For language model training in Finnish and Estonian experiments we used the growing n-gram training algorithm (Siivola & Pellom, 2005). In this algorithm, the n-grams that increase the training set likelihood enough with respect to the corresponding increase in the model size are accepted into the model (as in the MDL principle). After the growing process the model is further pruned with entropy based pruning. The method allows us to train compact and properly smoothed models using high order n-grams, since only the necessary high-order statistics are collected and stored (Siivola et al., 2007). Using the variable order n-grams we can also effectively control the size of the models to make all compared language models equally large. In this way the n-grams using shorter units do not suffer from a restricted span length which is the case when only 3-grams or 4-grams are available. For language model training in Turkish, n-gram language models were built with SRILM toolkit (Stolcke, 2002). To be able to handle computational

limitations, entropy-based pruning (Stolcke, 1998) is applied. In this pruning, the *n*-grams that change the model entropy less than a given threshold are discarded from the model.
The recognition tasks are speaker independent fluent dictation of sentences taken from newspapers and books for Finnish and Estonian. BN transcription system is used for Turkish experiments.

3.1 Finnish

Finnish is a highly inflected language, in which words are formed mainly by agglutination and compounding. Finnish is also the language for which the algorithm for the unsupervised morpheme discovery (Creutz & Lagus, 2002) was originally developed. The units of the morph lexicon for the experiments in this paper were learned from a joint corpus containing newspapers, books and newswire stories of totally about 150 million words (CSC, 2001). We obtained a lexicon of 50K statistical morphs by feeding the learning algorithm with the word list containing the 390K most common words. The average length of a morph was 3.4 letters including a word break symbol whereas the average word length was 7.9 letters. For comparison we also created a lexicon of 69K grammatical morphs based on rule-based morphological analysis of the words. For language model training we used the same text corpus and the growing n-gram training algorithm (Siivola & Pellom, 2005) and limited the language model size to approximately 40M n-grams for both statistical and grammatical morphs and words.
The speech recognition task was speaker independent reading of full sentences recorded over fixed telephone line. Cross-word triphone models were trained using 39 hours from 3838 speakers. The development set was 46 minutes from 79 new speakers and the evaluation set was another corresponding set. The models included tied state hidden HMMs of totally 1918 different states and 76046 Gaussian mixture (GMM) components, short-time mel-cepstral features (MFCCs), maximum likelihood linear transformation (MLLT) and explicit phone duration models (Pylkkönen & Kurimo, 2004). No speaker or telephone call specific adaptation was performed. Real-time factor of recognition speed was about 10 xRT.

3.2 Estonian

Estonian is closely related to Finnish and a similar language modeling approach was directly applied to the Estonian recognition task. The text corpus used to learn the morph units and train the statistical language model consisted of newspapers and books, altogether about 127 million words (Segakorpus, 2005). As in the Finnish experiments, a lexicon of 50K statistical morphs was created using the Morfessor Baseline algorithm as well as a word lexicon with a vocabulary of 500K most common words in the corpus. The average length of a morph was 2.9 letters including a word break symbol whereas the average word length was 6.6 letters. The available grammatical morphs in Estonian were, in fact, closer to the stem-ending models, for which a vocabulary of 500K most common units was chosen. Corresponding growing n-gram language models (approximately 40M n-grams) as in Finnish were trained from the Estonian corpus.
The speech recognition task in Estonian consisted of long sentences read by 50 randomly picked held-out test speakers, 8 sentences each (a part of (Meister et al., 2002)). The training data consisted of 110 hours from 1266 speakers recorded over fixed telephone line as well as cellular network. This task was more difficult than the Finnish one, one reason being the more diverse noise and recording conditions. The acoustic models were rather similar cross-

word triphone GMM-HMMs with MFCC features, MLLT transformation and the explicit phone duration modeling than the Finnish one, except larger: 3101 different states and 49648 GMMs (fixed 16 Gaussians per state). Thus, the recognition speed is also slower than in Finnish, about 30 xRT. No speaker or telephone call specific adaptation was performed.

3.3 Turkish

Turkish is another agglutinative language with relatively free word order. The same Morfessor Baseline algorithm (Creutz & Lagus, 2005) as in Finnish and Estonian was applied to Turkish texts as well. Using the 394K most common words from the training corpus, 34.7K morph units were obtained. The training corpus consists of 96.4M words taken from various sources: online books, newspapers, journals, magazines, etc. In average, there were 2.38 morphs per word including the word break symbol. Therefore, n-gram orders higher than words are required to track the n-gram word statistics and this results in more complicated language models. The average length of a morph was 3.1 letters including a word break symbol whereas the average word length was 6.4 letters. As a reference model for grammatical sub-words, we also performed experiments with stem-endings. The reason for not using grammatical morphemes is that they introduced several very short recognition units. In the stem-ending model, we selected the most frequent 50K units from the corpus. This corresponds to the most frequent 40.4K roots and 9.6K endings. The word OOV rate with this lexicon was 2.5% for the test data. The advantage of these units compared to the other sub-words is that we have longer recognition units with an acceptable OOV rate. In the stem-ending model, the root of each word was marked instead of using word break symbols to locate the word boundaries easily after recognition. In addition, a simple restriction was applied to enforce the decoder not to generate consecutive ending sequences.
For the acoustic data, we used the Turkish Broadcast News database collected at Boğaziçi University (Arısoy et et al., 2007). This data was partitioned into training (68.6 hours) and test (2.5 hours) sets. The training and test data were disjoint in terms of the selected dates.
N-gram language models for different orders with interpolated Kneser-Ney smoothing were built for the sub-word lexicons using the SRILM toolkit (Stolcke, 2002) with entropy-based pruning. In order to eliminate the effect of language model pruning in sub-words, lattice output of the recognizer was re-scored with the same order *n*-gram language model pruned with a smaller pruning constant. The transcriptions of acoustic training data were used in addition to the text corpus in order to reduce the effect of out-of-domain data in language modeling. A simple linear interpolation approach was applied for domain adaptation.
The recognition tasks were performed using the AT&T Decoder (Mohri & Riley, 2002). We used decision-tree state clustered cross-word triphone models with approximately 7500 HMM states. Instead of using letter to phoneme rules, the acoustic models were based directly on letters. Each state of the speaker independent HMMs had a GMM with 11 mixture components. The HTK front-end (Young et al., 2002) was used to get the MFCC based acoustic features. The baseline acoustic models were adapted to each TV/Radio channel using supervised MAP adaptation on the training data, giving us the channel adapted acoustic models.

4. Experimental results

The recognition results for the three different tasks: Finnish, Estonian and Turkish, are provided in Tables 1-3. In addition to sub-word language models, large vocabulary word-

based language models were built as the reference systems with similar OOV rates for each language. The word-based reference language models were trained as much as possible in the same way as the corresponding morph language models. For Finnish and Estonian the growing *n*-grams (Siivola & Pellom, 2005) were used. For Turkish a conventional n-gram with entropy-based pruning was built by using SRILM toolkit similarly as for the morphs.
For Finnish, Estonian and Turkish experiments, the LVCSR systems described in Section 3 are utilized. In each task the word error rate (WER) and letter error rate (LER) statistics for the morph-based system is compared to corresponding grammatical sub-word-based and word-based systems. The resulting sub-word strings are glued to form the word-like units according to the word break symbols included in the language model (see Fig. 1) and the markers attached to the units. The WER is computed as the sum of substituted, inserted and deleted words divided by the correct number of words. In agglutinative languages the words are long and contain a variable amount of morphemes. Thus, any incorrect prefix or suffix would make the whole word incorrect. Therefore, in addition to WER, LER is included here as well.

Finnish	Lexicon	OOV (%)	WER (%)	LER (%)
Words	500 K	5.4	26.8	7.7
Statistical morphs	50 K	0	21.7	6.8
Grammatical morphemes	69 K	0*	21.6	6.9

Table 1. The LVCSR performance for the Finnish telephone speech task (see Section 3.1). The words in (*) were segmented into grammatical morphs using a maximum likelihood segmentation by Viterbi search.

Estonian	Lexicon	OOV (%)	WER (%)	LER (%)
Words	500 K	5.6	34.0	12.3
Statistical morphs	50 K	0	33.9	12.2
Grammatical morphemes	500 K	0.5*	33.5	12.4

Table 2. The LVCSR performance for the Estonian telephone speech (see Section 3.2). The words in (*) were segmented into grammatical morphs using a maximum likelihood segmentation by Viterbi search.

Turkish	Lexicon	OOV (%)	WER (%)	LER (%)
Words	100K	5.3	37.0	19.3
Statistical morphs	37.4K	0	35.4	18.5
Grammatical stem-endings	50K	2.5	36.5	18.3

Table 3. Turkish BN transcription performance with channel adapted acoustic models (see Section 3.3). Best results are obtained with 3-gram word, 5-gram morph and 4-gram stem-ending language models. Note that roots are marked in stem-endings instead of using word break symbols.

In all three languages statistical morphs perform almost the same or better than the large vocabulary word reference models with smaller vocabulary sizes. The performance of the morph model is more pronounced in the Finnish system where the Morfessor algorithm was

originally proposed. In addition, grammatical morphemes achieve similar performances with their statistical counterparts. Even though grammatical stem-endings in the Turkish system attain almost the same LER with the statistical morphs, statistical morphs perform better than stem-endings in terms of the WER.

5. Conclusion

This work presents statistical language models trained on different agglutinative languages utilizing a lexicon based on the recently proposed unsupervised statistical morphs. The significance of this work is that similarly generated sub-word unit lexica are developed and successfully evaluated in three different LVCSR systems in different languages. In each case the morph-based approach is at least as good or better than a very large vocabulary word-based LVCSR language model. Even though using sub-words alleviates the OOV problem and performs better than word language models, concatenation of sub-words may result in over-generated items. It has been shown that with sub-words recognition accuracy can be further improved with post processing of the decoder output (Erdoğan et al., 2005; Arısoy & Saraçlar, 2006).

The key result of this chapter is that we can successfully apply the unsupervised statistical morphs in large vocabulary language models in all the three experimented agglutinative languages. Furthermore, the results show that in all the different LVCSR tasks, the morph-based language models perform very well compared to the reference language model based on very large vocabulary of words. The way that the lexicon is built from the word fragments allows the construction of statistical language models, in practice, for almost an unlimited vocabulary by a lexicon that still has a convenient size. The recognition was here restricted to agglutinative languages and tasks in which the language used is both rather general and matches fairly well with the available training texts. Significant performance variation in different languages can be observed here, because of the different tasks and the fact that comparable recognition conditions and training resources have not been possible to arrange. However, we believe that the tasks are still both difficult and realistic enough to illustrate the difference of performance when using language models based on a lexicon of morphs vs. words in each task. There are no directly comparable previous LVCSR results on the same tasks and data, but the closest ones which can be found are around 15% WER for a Finnish microphone speech task (Siivola et al., 2007), around 40% WER for the same Estonian task (Alumäe, 2005; Puurula & Kurimo, 2007) and slightly over 30% WER for a Turkish task (Erdoğan et al., 2005).

Future work will be the mixing of the grammatical and statistical sub-word-based language models, as well as extending this evaluation work to new languages.

6. Acknowledgments

The authors would like to thank Sabancı and ODTÜ universities for the Turkish text data and AT&T Labs – Research for the software. This research is partially supported by TÜBİTAK (The Scientific and Technological Research Council of Turkey) BDP (Unified Doctorate Program), TÜBİTAK Project No: 105E102, Boğaziçi University Research Fund Project No: 05HA202 and the Academy of Finland in the projects *Adaptive Informatics* and *New adaptive and learning methods in speech recognition*.

7. References

Alumäe, T. (2005). Phonological and morphological modeling in large vocabulary continuous Estonian speech recognition system, *Proceedings of Second Baltic Conference on Human Language Technologies*, pages 89–94.

Arısoy, E.; Dutağacı, H. & Arslan, L. M. (2006). A unified language model for large vocabulary continuous speech recognition of Turkish. Signal Processing, vol. 86, pp. 2844–2862.

Arısoy, E. & Saraçlar, M. (2006). Lattice extension and rescoring based approaches for LVCSR of Turkish, *Proceedings of Interspeech*, Pittsburgh, PA, USA.

Arısoy, E.; Sak, H. & Saraçlar, M. (2007). Language modeling for automatic Turkish broadcast news transcription, Proceedings of Interspeech, Antwerp, Belgium.

Bayer, A. O.; Çiloğlu, T & Yöndem, M. T. (2006). Investigation of different language models for Turkish speech recognition, *Proceedings of 14th IEEE Signal Processing and Communications Applications,* pp. 1–4, Antalya, Turkey.

Byrne, W.; Hajic, J.; Ircing, P.; Jelinek, F.; Khudanpur, S.; Krbec, P. & Psutka, J. (2001). On large vocabulary continuous speech recognition of highly inflectional language - Czech, *Proceedings of Eurospeech 2001*, pp. 487–490, Aalborg, Denmark.

Creutz, M. & Lagus, K. (2002). Unsupervised discovery of morphemes, *Proceedings of the Workshop on Morphological and Phonological Learning of ACL-02*, pages 21–30.

Creutz, M. & Lagus, K. (2005). Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor. Technical Report A81, Publications in Computer and Information Science, Helsinki University of Technology. URL: http://www.cis.hut.fi/projects/morpho/.

Creutz, M.; Hirsimaki, T.; Kurimo, M.; Puurula, A.; Pylkkönen, J.; Siivola, V.; Varjokallio, M.; Arısoy, E.; Saraçlar, M. & Stolcke, A. (2007a). Analysis of morph-based speech recognition and the modeling of out-of-vocabulary words across languages. *Proceedings of HLT-NAACL 2007*, pp. 380–387, Rochester, NY, USA.

Creutz, M.; Hirsimäki, T.; Kurimo, M.; Puurula, A.; Pylkkönen, J.; Siivola, V.; Varjokallio, M.; Arısoy, E.; Saraçlar, M. & Stolcke, A. (2007b). Morph-Based Speech Recognition and Modeling of Out-of-Vocabulary Words Across Languages. *ACM Transactions on Speech and Language Processing*, Vol. 5, No. 1, Article 3.

Erdoğan, H.; Büyük, O. & Oflazer, K. (2005). Incorporating language constraints in sub-word based speech recognition. *Proceedings of IEEE ASRU*, San Juan, Puerto Rico

Hacıoğlu, K.; Pellom, B.; Çiloğlu, T.; Öztürk, Ö; Kurimo, M. & Creutz, M. (2003). On lexicon creation for Turkish LVCSR, *Proceedings of Eurospeech,* Geneva, Switzerland.

Hetherington, I. L. (1995). A characterization of the problem of new, out-of-vocabulary words in continuous-speech recognition and understanding. *Ph.D. dissertation*, Massachusetts Institute of Technology.

Hirsimäki T.; Creutz, M.; Siivola, V.; Kurimo, M.; Virpioja, S. & J. Pylkkönen. (2006). Unlimited vocabulary speech recognition with morph language models applied to Finnish. *Computer, Speech and Language*, vol. 20, no. 4, pp. 515–541.

Garofolo, J.; Auzanne, G. & Voorhees, E. (2000). The TREC spoken document retrieval track: A success story, *Proceedings of Content Based Multimedia Information Access Conference*, April 12-14.

Kanevsky, D.; Roukos, S.; & Sedivy, J. (1998). Statistical language model for inflected languages. US patent No: 5,835,888.

Kwon, O.-W. & Park, J. (2003). Korean large vocabulary continuous speech recognition with morpheme-based recognition units. *Speech Communication*, vol. 39, pp. 287–300.

Meister, E.; Lasn, J. & Meister, L. (2002). Estonian SpeechDat: a project in progress, *Proceedings of the Fonetiikan Päivät–Phonetics Symposium 2002 in Finland*, pages 21–26.

Mengüşoğlu, E. & Deroo, O. (2001). Turkish LVCSR: Database preparation and language modeling for an agglutinative language. *Proceedings of ICASSP 2001, Student Forum*, Salt-Lake City.

Mohri, M & Riley, M. D. DCD Library – Speech Recognition Decoder Library. AT&T Labs – Research. http://www.research.att.com/sw/tools/dcd/.

NIST. (2000). *Proceedings of DARPA workshop on Automatic Transcription of Broadcast News*, NIST, Washington DC, May.

Podvesky, P. & Machek, P. (2005). Speech recognition of Czech - inclusion of rare words helps, *Proceedings of the ACL SRW*, pp. 121–126, Ann Arbor, Michigan, USA.

Puurula, A. & Kurimo M. (2007). Vocabulary Decomposition for Estonian Open Vocabulary Speech Recognition. *Proceedings of the* ACL 2007.

Pylkkönen, J. & Kurimo, M. (2004). Duration modeling techniques for continuous speech recognition, *Proceedings of the International Conference on Spoken Language Processing*.

Pylkkönen, J. (2005). New pruning criteria for efficient decoding, *Proceedings of 9th European Conference on Speech Communication and Technology*.

Rosenfeld, R. (1995). Optimizing lexical and n-gram coverage via judicious use of linguistic data, *Proceedings of Eurospeech*, pp. 1763–1766.

Sak, H.; Güngör, T. & Saraçlar, M. (2008). Turkish language resources: Morphological parser, morphological disambiguator and web corpus, *Proceedings of 6th International Conference on Natural Language Processing, GoTAL* 2008*, LNAI 5221*, pp. 417–427..

Sak, H.; Güngör, T. & Saraçlar, M. (2007). Morphological disambiguation of Turkish text with perceptron algorithm, Proceedings of *CICLing 2007, LNCS 4394*, pp. 107–118.

Segakorpus–Mixed Corpus of Estonian. Tartu University. http://test.cl.ut.ee/korpused/segakorpus/.

Siivola, V. & Pellom, B. (2005). Growing an n-gram language model, *Proceedings of 9th European Conference on Speech Communication and Technology*.

Siivola, V.; Hirsimäki, T. & Virpioja, S. (2007). On Growing and Pruning Kneser-Ney Smoothed N-Gram Models. *IEEE Transactions on Audio, Speech and Language Processing*, Volume 15, Number 5, pp. 1617-1624.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit, *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904.

Stolcke, A. (1998). Entropy-based pruning of back-off language models, *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, pages 270–274.

ASR SYSTEMS

11

Discovery of Words: Towards a Computational Model of Language Acquisition

Louis ten Bosch[1], Hugo Van hamme[2] and Lou Boves[1]
[1]*Radboud University Nijmegen,*
[2]*Katholieke Universiteit Leuven,*
[1]*the Netherlands*
[2]*Belgium*

1. Introduction

Human speech recognition seems effortless, but so far it has been impossible to approach human performance by machines. Compared with human speech recognition (HSR), the error rates of state-of-the-art automatic speech recognition (ASR) systems are an order of magnitude larger (Lee, 2004; Moore, 2003; see also Scharenborg et al., 2005). This is true for many different speech recognition tasks in noise-free environments, but also (and especially) in noisy environments (Lippmann, 1997; Sroka & Braida, 2005; Wesker et al., 2005). The advantage for humans remains even in experiments that deprive humans from exploiting 'semantic knowledge' or 'knowledge of the world' that is not readily accessible for machines.

It is well known that there are several recognition tasks in which machines outperform humans, such as the recognition of license plates or barcodes. Speech differs from license plates and bar codes in many respects, all of which help to make speech recognition by humans a fundamentally different skill. Probably the most important difference is that bar codes have been designed on purpose with machine recognition in mind, while speech as a medium for human-human communication has evolved over many millennia. Linguists have designed powerful tools for analyzing and describing speech, but we hardly begin to understand how humans process speech. Recent research suggests that conventional linguistic frameworks, which represent speech as a sequence of sounds, which in their turn can be represented by discrete symbols, fail to capture essential aspects of speech signals and, perhaps more importantly, of the neural processes involved in human speech understanding. All existing ASR systems are tributable to the beads-on-a-string representation (Ostendorf, 1999) invented by linguistics. But is quite possible –and some would say quite likely- that human speech understanding in not based on neural processes that map dynamically changing signals onto sequences of discrete symbols. Rather, it may well be that infants develop very different representations of speech during their language acquisition process. Language acquisition is a side effect of purposeful interaction between infants and their environment: infants learn to understand and respond to speech because it helps to fulfil a set of basic goals (Maslow, 1954; Wang, 2003). An extremely important need is being able to adapt to new situations (speakers, acoustic environments, words, etc.) Pattern recognisers, on the other hand, do not aim at the optimisation of 'purposeful

interaction'. They are trained to recognize pre-defined patterns, and decode an input signal in terms of a sequence of these patterns. As a consequence, automatic speech recognisers have serious problems with generalisations. Although modern ASR systems can adapt to new situations, this capability is limited to a predefined set of transformations (Moore & Cunningham, 2005).

Can the gap in speech recognition performance between humans and machines be closed? Many ASR scientists believe that today's statistical pattern recognisers are not capable of doing this (see e.g. Moore, 2003). Most probably ASR can only be improved fundamentally if entirely new approaches are developed (Bourlard et al, 1996). We are trying to do just this, by investigating the way how infants acquire language and learn words and to see to what extent this learning process can be simulated by a computational model. Many branches of Cognitive Science, such as Psycho-linguistics, and Communication Science have contributed to a large mass of data about the speech processing skills of adults and the ways in which these skills develop during infancy and childhood (MacWhinney, 1998; Gerken & Aslin, 2005; Gopnik et al., 2001; Jusczyk, 1999; Kuhl, 2004; Kuhl et al., 2003; Swingley, 2005; Smith & Yu, 2007). Despite the large number of studies, it is not yet clear how exactly infants acquire speech and language (Werker & Yeung, 2005), and how an adult's speech processing can be as fast and robust against novel and adverse conditions as it apparently is. The design and use of a computational model is instrumental in pinpointing the weak and strong parts in a theory. In the domain of cognition, this is evidenced by the emergence of new research areas such as Computational Cognition and Cognitive Informatics (e.g. Wang et al, 2007).

In this chapter, we describe research into the process of language acquisition and speech recognition by using a computational model. The input for this model is similar to what infants experience: auditory and visual stimuli from a carer grounded in a scene. The input of the model therefore comprises multimodal stimuli, each stimulus consisting of a speech fragment in combination with visual information. Unlike in a conventional setting for training an ASR system, the words and their phonetic representation are not known in advance: they must be discovered and adapted during the training.

In section 2, we will present the model in more detail. The communication between the learner model and the environment is discussed in section 3. In section 4, the mathematical details of one specific instantiation of the learning algorithm are explained, while section 5 describes three experiments with this particular algorithm. Discussion and conclusion are presented in sections 6 and 7.

2. The model

2.1 Background

In order to be able to effectively communicate, infants must learn to understand speech spoken in their environment. They must learn that auditory stimuli such as stretches of speech are not arbitrary sounds, but instead are reoccurring patterns associated with objects and events in the environment. Normally this development process results in neural representations of what linguists call 'words'. This word discovery process is particularly interesting since infants start without any lexical knowledge and the speech signal does not contain clear acoustic cues for boundaries between words. The conventional interpretation is that infants must 'crack' the speech code (Snow & Ferguson, 1977; Kuhl, 2004) and that the discovery of word-like entities is the first step towards more complex linguistic analyses

(Saffran and Wilson, 2003). However, it seems equally valid to say that infants must construct their individual speech code, a complex task in which attention, cognitive constraints, social and pragmatic factors (and probably many more) all play a pivotal role.

Psycholinguistic research shows that infants start with learning prosodic patterns, which are mainly characterised by their pitch contours and rhythm. A few months later, infants can discriminate finer details, such as differences between vowels and consonants (e.g. Jusczyk, 1999; Gopnik et al., 2001). At an age of about 7 months infants can perform tasks that are similar to word segmentation (e.g. Werker et al., 2005 and references therein; Newport, 2006; Saffran et al., 1996; Aslin et al., 1998; Johnson & Jusczyk, 2001; Thiessen & Saffran, 2003). These skills can be accounted for by computational strategies that use statistical co-occurrence of sound sequences as a cue for word boundaries. Other experiments suggest that the discovery of meaningful 'words' is facilitated when the input is multimodal (e.g. speech plus vision), experiments (Prince & Hollich, 2005) and computational models (such as the CELL model, Roy & Pentland, 2002).

As observed above, the design and test of a computational model of word discovery may be pivotal for our understanding of language acquisition in detail. Simultaneously, such a model will inform possible ways to fundamentally alter (and hopefully improve) the conventional training-test paradigm in current ASR research. The classical limitations for defining and modelling words and phonemes in ASR might be radically reduced by exploring alternatives for data-driven word learning (e.g. by the use of episodic models – see Goldinger, 1998; Moore, 2003).

The computational model that we are developing differs from most existing psycho-liguistic models. Psycho-linguistic models of human speech processing (e.g. TRACE, McLelland & Elman, 1986; Shortlist, Norris, 1994; Luce & Lyons, 1998; Goldinger, 1998; Scharenborg et al., 2005; Pisoni & Levi, 2007; Gaskell, 2007) use a predefined lexicon and take symbolic representations of the speech as their input. The fact that a lexicon must be specified means that these models are not directly applicable for explaining word discovery (nor other aspects of language acquisition). The success of these models, however, suggests that concepts such as activation, competition and dynamic search for pattern sequences are essential ingredients for any model aiming at the simulation of human speech processing (cf. Pitt et al, 2002, for a discussion about these topics).

The computational framework that we propose in this paper builds on Boves et al. (2007) and combines the concepts of competition and dynamic sequence decoding. Simultaneously, it builds the lexicon in a dynamic way, starting empty at the beginning of a training run. During training, the model receives new utterances, and depending on the internal need to do so, new representations are hypothesized if existing representations fail to explain the input in sufficient detail.

The model hypothesizes that patterns are stored in memory mainly on the basis of bottom-up processing. Bottom-up models performing pattern discovery are also described in Park & Glass (2006) and ten Bosch & Cranen (2007). These models are based on a multi-stage approach in which first a segmentation of the speech signal is carried out, after which a clustering step assigns labels to each of the segments. In the final stage, then, these symbolic representations are used to search for words. The important difference beween ten Bosch & Cranen (2007) on the one hand and Park & Glass (2006) and Roy & Pentland (2002) on the other is that former does not rely on the availability of a phonetic recogniser to transcribe speech fragments in terms of phone sequences. Models that do bottom-up segmentation

have already been designed in the nineties by Michiel Bacchiani, Mari Ostendorf and others. But the aim of these models was entirely different from ours: the automatic improvement of the transcription of words in the lexicon (Bacchiani et al., 1999).

2.2 Architecture

Our novel computational model of language acquisition and speech processing consists of two interacting sub-models: (1) the carer and (2) the learner. In this paper we focus on the architecture of the learner model. The computational model of the learner must be able to perform three major subtasks.

Feature extraction

The learner model has multimodal stimuli as input. Of course, the speech signal lives in the auditiry modality. To process the audio input, the model has an auditory front-end processor, i.e., a module that converts acoustic signals into an internal representation that can be used for learning new patterns and for decoding in terms of known patterns. The front-end generates a redundant representation that comprises all features that have been shown to affect speech recognition (and production) in phonetic and psycholinguistic experiments. However, for the experiments described in this chapter we only used conventional Mel Frequency Cepstral Coefficients (with c_0) and log energy.

In the second modality (vision), we sidestep issues in visual processing by simulating the perception of objects and events in the scene by means of symbols (in the simplest version) or possibly ambiguous feature vectors (in more complex versions of the model).

Pattern discovery

The learning paradigm of the computational model is different from conventional automatic speech recognition approaches. In conventional speech recognition systems the patterns to be recognised are almost invariably lexical entries (words), representeds in the form of sequences of phonemes. In the current model, we avoid the a priori use of subword units and other segmental models to hypothesize larger units such as words, and explicitly leave open the possibility that the model store patterns in the form similar to episodes (see also McQueen, 2007).

Apart from the question how meaningful (word-like) units can be *represented*, the discovery of words from the speech signal is not straightforward. In our model, we use two strategies: (1) exploit the repetitive character of infant-directed speech (Thiessen et al., 2005) (2) make use of the cross-modal associations in the speech and the vision modality. This is based on the fact that infants learn to associate auditory forms and visual input by the fact that the same or similar patterns reappear in the acoustic input whenever the corresponding visual scene is similar (Smith & Yu, 2007; see also Shi et al, 2008).

The chosen architecture is such that representations of word-like units develop over time, and become more detailed and specialised as more representations must be discriminated.

Memory access

Theorists on the organisation of human memory disagree on the functioning of human memory and how exactly the cognitive processes should be described. However, there is consensus about three processes that each plays a different role in cognition (MacWhinney, 1998). Broadly speaking, a sensory store holds sensory data for a very short time (few seconds), a short-term memory (holding data for about one minute) acts as 'scratch pad' and is also used for executive tasks, while a long-term memory is used to store patterns (facts e.g. names and birthdays, but also skills such as biking) for a very long time.

The short-term memory allows to store a representation of the incoming signal (from the sensory store) and to compare this representation to the learned representations retrieved from long-term memory. If the newly received and previously stored patterns differ mildly, stored representations can be adapted. If the discrepancy is large, novel patterns are hypothesized and their activation is increased if they appear to be useful in following interactions. If they are not useful, their activation will decay and eventually they will not be longer accessible. Short-term memory evaluates and contains activations, while long-term memory stores representations.
The input and architecture of the computational model are as much as possible motivated by cognitive plausibility. The words, their position in the utterance, and its acoustic/phonetic representation are unspecified, and it is up to the model to (statistically) determine the association between the word-like speech fragment and the referent.

3. Interaction and communication

Language acquisition takes place in communication loops between the infant and the environment. In the beginning of language acquisition, the number of persons that the infant interacts with is usually limited, which leads to patterns that are biased towards the personal voice characteristics of these few caretakers. As soon as the infant is addressed by more persons, the stored representations will be adapted in some way to accommodate the differences between speakers (and other differences, such as speaking styles).
The communicative framework involves two active participants and simulates a 'learner' involved in interaction with a 'carer'. The learner discovers words and word-like entities on the basis of the grounded stimuli presented by the carer during the interaction.
The learner starts with an almost empty memory, and during the interaction between learner and carer, the learner gradually detects more and different meaningful sound patterns. This is done by first hypothesising an internal representation of a word-like entity, followed by strengthening or weakening of this representation on the basis of new stimuli. This means that the concept of word is not built-in a priori, but that meaningful acoustic patterns come about as an emergent property during learning. 'Words' in the linguistic sense of the term are meta-level concepts that children acquire when they start talking *about* language.
The multimodal stimuli from which our model must learn consist of two parts (a) the audio parting the form of real speech signals (short utterances) and (b) the visual (semantic) input corresponding to the meaning of the utterances. This visual representation in the experiments described here is an abstract tag, which uniquely refers to the object that is referred to by the utterance. In the experiments described in section 5, we use 13 of these tags (representing 13 target words). The tags represent an abstraction of the information that would otherwise be available along the visual modality. The tag itself does not give any clue about the word, or the phonetic representation of any target word.
The speech used for training the learner is highly repetitive in terms of verbal content and produced by four speakers. In a later phase of the learning process the model will be exposed to speech produced by other speakers. The communication starts when the carer presents a multimodal stimulus to the learner. Once the carer has provided a stimulus, the learner's response consists of the concept the learner thinks is referred to by the audio part of the stimulus. This reply is combined with a confidence measure. In the learner's memory this results in an update of the internal representations of the concepts.

The emphasis is on learning a small vocabulary, starting with an empty lexicon. A basic vocabulary must be formed by listening to simple speech utterances that will be presented in the context of the corresponding concepts.

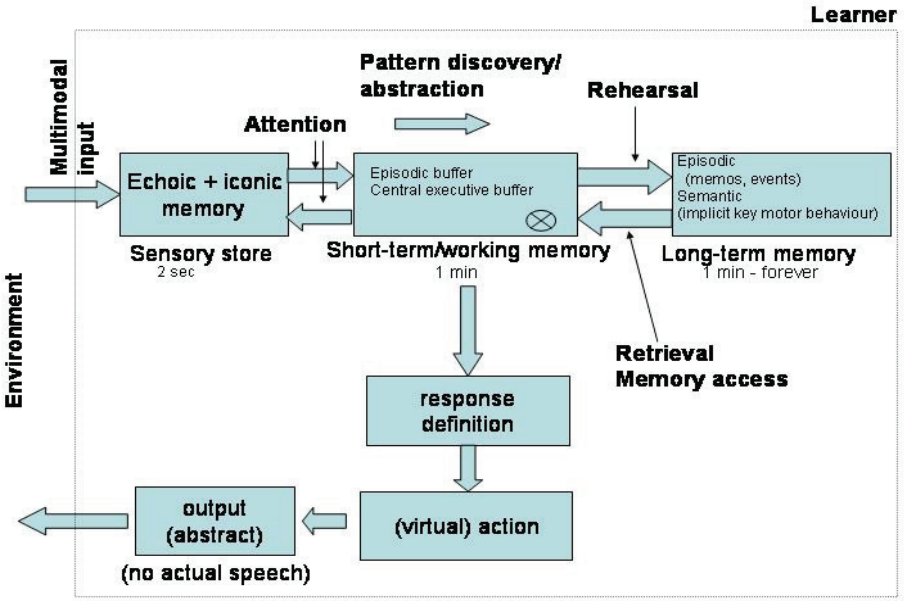


Fig. 1. This picture shows an overview of the overall interaction between learner model (within grey-line box) and the environment (i.e. carer, outside the box). Multimodal stimuli are input of the model (top-left corner). For an explanation see the text.

A schematic representation of the interaction between the learner and the carer is shown in figure 1. The learner is depicted within the grey box, while the carer is indicated as the environment outside the grey box. A training session consists of a number of interaction cycles, each cycle consisting of several turns. Per cycle, the learner receives multimodal input from the carer after which a reply is returned to the carer. In the next turn, the carer provides the learner with a feedback about the correctness of the response, after which it is up to the learner to use this feedback information.

When the learner perceives input, the speech input is processed by the feature extraction module. The outcome is stored in the sensory store, from where it is transferred to short-term memory (STM) if the acoustic input is sufficiently speech-like (to be determined by the attention mechanism in Figure 1). In STM, a comparison takes place between the sensory input on the one hand and the stored representations on the other. The best matching representation (if any) is then replied to the carer.

The role of the carer

The carer provides multimodal utterances to the learner. The moment at which the carer speaks to the learner is determined by a messaging protocol that effectively controls the interaction during a training session. The utterances used during training and their ordering are determined by this protocol. After a reply from the learner, the carer provides feedback

about the correctness of the reply. In the current implementation, the feedback is just a binary yes/no (approval/disproval).

Learning drive

The communication between carer and learner is not enough for learning. Learning is a result of a learning drive. Exactly which drive makes the learner learn? When looking at real life situations, a baby's drive to learn words is ultimately rooted in the desire to have the basic needs for survival fulfilled: get food, care and attention from the carers. In the current model, this 'need' is implemented in the form of an 'internal' drive to build an efficient representation of the multimodal sensory input, in combination with an 'external' drive to optimise the perceived appreciation by the carer.

The internal drive basically boils down to the quality of the parse of the input. Given a certain set of internal representations, the leaner is able to parse the input to a certain extent. If the input cannot be parsed, this means that representations must be updated or even that a new representation must be hypothesised and stored.

The external drive (related to the optimisation of the appreciation by the carer) is directly reflected in the optimisation of the accuracy of the learner's responses (i.e. minimisation of the error rates). The optimisation of the accuracy can mathematically be expressed in terms of constraints on the minimisation between predicted reply (predicted by the learner model) and the observed ground truth as provided in the stimulus tag.

4. Learning and decoding algorithm

In the current implementation of the learner's model, training and decoding is done by first combining acoustic and visual/semantic information from the incoming new stimulus into one single vector. Thus, each stimulus is represented as a vector with a fixed dimension. As a result, a sequence of stimuli is represented as a matrix (in this chapter, this data matrix will be referred to by X). The actual search for patterns is performed by a decomposition technique called Non-Negative Matrix Factorisation (NMF) on X (Hoyer, 2004; Stouten et al, 2007, 2008). NMF is a technique to find structure in (large) data sets. The usefulness of NMF for our purpose derives from the fact that it is able to decompose the very large matrix X into two (much smaller) matrices W and H such that

a. $X \approx WH$: The distance between X and the product WH is minimised according to some distance function (see below)
b. All components of X, W and H are positive or zero

In our experiments, X represents previously learnd (but constantly updatable) patterns residing in the long term memory of the learner. Prior to the training X is initialised to the empty matrix. Each new utterance is first encoded in a vector which is then appended to the current matrix X. If there is no forgetting, the number of columns of X equals the number of utterances observed so far in the training run. This is reminiscent of episodic memory.

After NMF decomposition, the columns in W act as basis vectors into which the columns of X are represented. What we want the NMF to produce is a decomposition of each utterance in terms of word-like entities. In the current experiments, where X is based on utterances (in linguistic term: sequences of words), each column of W should therefore ideally be related to a single word-like entity. A necessary condition to enable this utterance–to-word decomposition is provided by the way in which new utterances are mapped to a new column of X. This mapping will be denoted *map* below. If s_1 and s_2 are two arbitrary speech segments, and $s_1 \oplus s_2$ is the concatenation of these segments, then *map* must meet the following property:

$$map(s1 \oplus s2) = map(s1) + map(s2)$$

where the right-hand side '+' sign refers to the usual vector-addition of two vectors.
The matrix H contains the corresponding weights (activations) in the decompositions. If a certain entry in H (h_{ij}) is positive, it means that the corresponding column of W is required with weight h_{ij} to explain the corresponding column in X.
In this decomposition, the number of columns in W (and so the number of rows in H) is a model parameter. This number must be estimated on the basis of the number of input stimuli observed so far. In the current implementation, this number is initialised to 5 and increases with the number of observed stimuli. The way how this number increases is basically heuristically determined. The number of columns in W must be larger than the number of concepts that must be distinguished. Else, it would not be possible to account for different linguistic contexts of the target words (the acoustic context of the target words in the utterances). This implies a tendency for the learner to overestimate the number of things in the environment that must be distinguished.
In the current implementation, the NMF factorisation takes place for the first time after K stimuli have been observed, where K is a user-defined number. This corresponds to the assumption that reorganization towards a more efficient storage of episodic traces in the memory is only necessary after a certain number of observations. After this initialisation, the resulting W and H are updated after each following stimulus. As a result, W and H evolve gradually as long as more utterances are being observed.
To *decode* a new (not yet observed) utterance U, the *map* operation is applied on U, and a vector h is sought such that the difference between

$$map(U) \text{ and } W\,h$$

with W the *current* internal representation, is minimised. As a result, the vector h encodes the utterance in terms of activations of the columns of W: The winning column is the one corresponding to the highest value in h.
As said above, the multimodal stimulus contains a tag corresponding to visual input; this tag can be coded into W such that each column of W is statistically associated with a tag. In combination with the information in the vector h, this association allows the learner to respond with the corresponding tag, in combination with the corresponding value in h.

NMF minimisation

The minimisation of the NMF cost function leads to the overall closest match between prediction and observation, and so to an overall minimisation of the recognition errors made by the learner. Hoyer (2004) presents two different NMF algorithms, each related to a particular distance that is to be minimised. In the case of minimisation of the Euclidean distance (Frobenius norm) between X and WH, the cost function that is minimised reads (see Hoyer, 2004 for details)

$$F_1(X, WH) = \sum_{i,j} (X_{ij} - [WH]_{ij})^2 / 2$$

while in case of the Kullback-Leibler divergence, this cost function reads

$$F_2(X, WH) = \sum_{ij} (X .* \log(X ./ (WH) - X + WH)_{ij}$$

In this formula, .* and ./ denote component-wise multiplication and division, respectively. The structure of the expressions at the right-hand side indicates that in both cases the error between prediction and observation is an accumulated sum over all tokens that are available in X (and so processed during training). Splitting the 2-way sum in two separate one-way sums (using i and j, respectively), the terms $(X_j\text{-}[WH]_j)^2$ and $(X .* \log(X./WH) - X + WH)_j$ can be interpreted as the internal target function that is to be minimized on a token-by-token basis. In these token-related expressions, X, WH and H are now column vectors rather than matrices; W is a matrix.

The second expression, related to the Kullback-Leibler distance between reference (X) and hypothesis (WH), can be regarded as the log-likelihood of the model (XH) predicting the observation (X). This, in turn, can be interpreted as a measure for the quality of the parse of the utterance associated to X, in terms of the concepts that are associated with the columns in matrix W.

Interestingly, the learning model does not need to segment utterances in order to hypothesize the presence of target words. Nor is the ordering of the words in the utterance used. This is so because the *map* function is symmetric:

$$map(s1 \oplus s2) = map(s1) + map(s2) = map(s2 \oplus s1)$$

This implies that word ordering is not reflected after the map operation. We come back to this property in the discussion section (section 6).

Figure 2 shows how the use of NMF relates to the use of abstraction, which is necessary to facilitate access to the internal representations of speech signals. It shows how information in X (at a low level of abstraction) is factorised to obtain more abstract information (in W and H). Thus, abstraction is mathematically modelled as factorisation of the data matrix X.

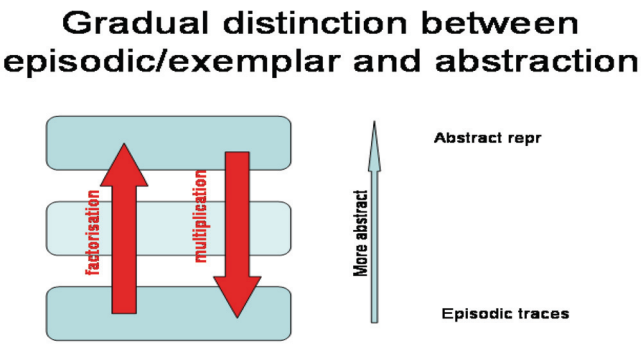


Fig. 2. This figure shows a multi-layered representation of the contents of the memory of the learner model. On the lowest level, data are represented in unreduced form. The higher the level, the more abstract the corresponding representation is. The picture shows the general idea of having multiple different levels of abstractness in parallel. In the current computational model of the learner, just two levels are used, an 'episodic' one (here: actual sequences of feature vectors obtained from the feature extraction module), and an abstract one (here: basis vectors in a vector space representing words, in combination with activation strengths). By using NMF, the conceptual bottom-up process of abstraction is translated into explicit matrix factorisations, while the top-down process is represented by matrix multiplications. These top-down and bottom-up processes can interact in a natural way since they use the same paradigm of algebraic matrix manipulation.

5. Experiments

5.1 Materials

For the experiments discussed here, we use three comparable databases collected in the ACORNS project: one Dutch database (NL), a Finnish database (FIN), and a Swedish database (SW). For each language, the databases contain utterances from 2 male and 2 female speakers. Each speaker produced 1000 utterances in two speech modes (adult-directed, ADS, and infant-directed, IDS). For the infant-directed style, all speakers were asked to act as if they addressed a child of about 8-12 months old. The resulting speech has the well-known characteristics of infant-directed speech, such as a more exaggerated intonation, clear pronunciation, and low speaking rate.

The set of 1000 uterances contains 10 repetitions of combinations of target words and 10 carrier sentences. Within a database, not all target words are uniformly distributed. While all 4 speakers share the same target words, the proper name they use to address the learner is different for each speaker. For example, the NL database (8000 utterances) contains 800 tokens of ecologically relevant target words such as luier (diaper), auto (car), but only 200 of the proper names mirjam, isabel, damian, otto. In total, there are 13 different target words per language.

5.2 Experimental set-up

In each experiment the training is based on a specific list of utterances selected from the available pool of 8000 utterances. The ordering in which the utterances are presented is one of the experimental parameters (for example, this ordering can be random or speaker-blocked). During a training the utterances are always processed utterance-by-utterance. It must be emphasized that in our approach there is no essential difference between training and test: each NMF update is based on the history observed so far (matrix X), while each new utterance is recognised (decoded) on the basis the stored representations (W) of the learner learned so far. In the experiments reported here, the length of the history-update window used in each NMF step is a parameter. One training session of the computational model consists in presenting (by the carer) the next not yet observed multimodal stimulus. The learner attempts to decode the audio part of this new input stimulus, and replies by providing its most active word hypothesis in combination with a confidence score. Then, exactly as in a real-life carer-child interaction, it is up to the carer-model to give feedback: by providing the next stimulus, or by correcting the model's reply.

5.3 Experiment 1

Experiment 1 aims at showing that the learner is able to create representations of target words, and that when a new speaker is encountered, these representations must be adapted towards the characteristics of the new speaker.

To that end, the pool of 8000 Dutch utterances was blocked by speaker, and randomized within speaker. The resulting utterance list contained 2000 utterances by a female speaker, followed by 2000 utterances produced by a male speaker, followed again by the utterances from another female and another male speaker.

The results of this word detection experiment are shown in Figure 3. The plot shows the performance of the learner, measured as average accuracy over the most recent 50 stimuli. The horizontal axis shows the number of stimuli (tokens) presented so far. The vertical axis shows the corresponding accuracy in terms of percentages correct responses. Each time a

new speaker starts, a drop in performance of about 20-30 percent points can be seen. This performance drop is mainly due to the fact that the word representations learned so far are inadequate to correctly parse the utterances by the new speaker. The dip shows that representations are dependent on the speakers previously encountered during training.
Given the learning settings, the learner is able to create adequate internal representations for 10 target words as produced by the first female speaker within about 1000 tokens (that is, approximately 100 tokens per word). For each new speaker, the performance is back on its previous high level within about 100 tokens per word. Results for Finnish and Swedish are very similar.
During the first few hundred utterances the learner does not have any repreentatin available and so does not respond in a meaningful manner; this explaines why the accuracy is zero.

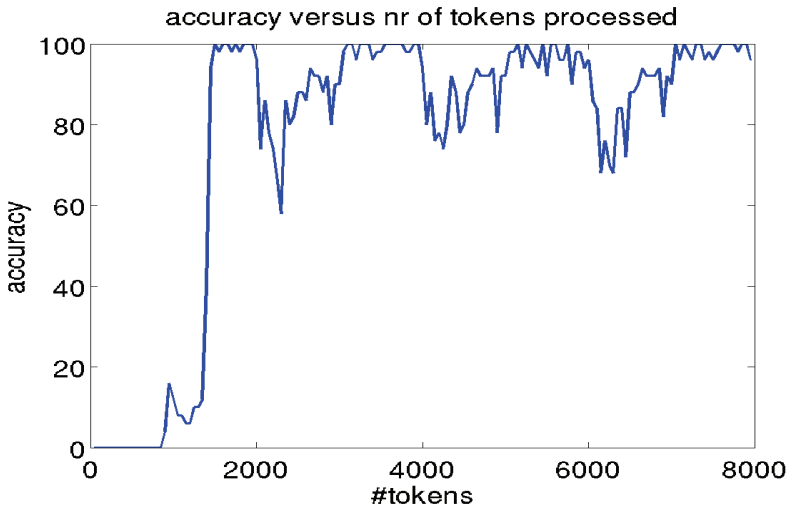


Fig. 3. Results of word detection experiment (for Dutch, speaker-blocked). The plot shows the performance of the learner, measured as average accuracy over the most recent 50 stimuli. The horizontal axis shows the number of stimuli (tokens) presented so far. The vertical axis shows the corresponding accuracy in terms of percentages. A drop in performance of about 20-30 percent point can be seen each time when a new speaker starts.

5.4 Experiment 2

During the training, the NMF update takes place after each utterance. Tus, there are two parameters in the model that affect the eventual performance of the learner. These parameters specify the update scheme for the internal representations: how many utterances are to be used in the update of the internal representations, and when the initialisation of the internal representation should occur. The first parameter (number of utterances used in each NMF step) is referred to by memory length (indicated by 'ml') – this parameter specifies something that might be called 'effective memory length'. The second parameter deals with the initialisation and denotes the number of stimuli before the first NMF decompistion ('nsbt').
In this experiment, we focus on the 2000 utterances of one Dutch female speaker. Figure 4a shows the dependency of the eventual performance of the memory length. Four values for

ml are shown (20, 100, 500, inf). The value 'inf' means that all utterances that are observed so far are used in the NMF updates. In this experiment, the value of nsbt is fixed to 100, which means that the very first NMF factorisation occurs after 100 utterances, after which recognition takes place.
The plot shows that the eventual performance largely depends on the memory length. Values of 500 and 'inf' do lead to results that are almost indistinguishable; a value of 100, however, leads to considerably lower performance. Translating this to the level of individual words, this implies that 50 tokens per word suffice, but 9 to 10 tokens are insufficient to yield adequate representations.
As shown in Fig. 4b the effect of the parameter nbst is much less dramatic. The most interesting observation is that there is no need to delay the first decomposition until after a large number of input stimuli have been observed. Delaying the first decomposition does not buy improvements in later learning. But in a real learning situation it might cost a baby dearly, because the carer might become frustrated by the lack of meaningful responses.

5.5 Experiment 3

In this experiment, the aim is to show that internal representations are changing continuously, and that we can exploit structure in the representation space by statistical means. This shows how abstraction may follow as a result of competition in crowded collections of representations on a lower level. For example, we would like to know whether speaker-dependent word representations can be grouped in such a way that the common characteristics of these representations combine into one higher-level word representation. We investigate this by first creating speaker-dependent word representations, followed by a clustering to arrive at speaker-*in*dependent word representations.

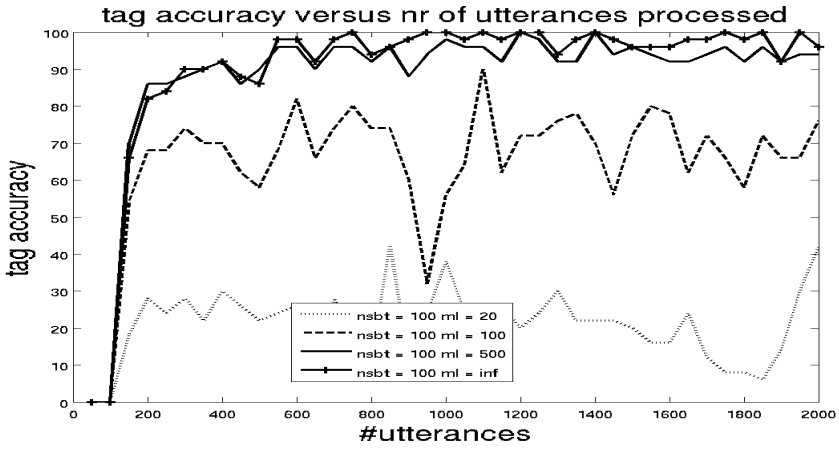


Fig. 4a. This figure shows the dependency of the eventual performance of the memory length. Three values for memory length (indicated by 'ml') are shown (20, 100, 500, inf). The value 'inf' means that all utterances that are observed so far are used in each NMF update. The number of stimuli that are processed before the first NMF-step ('nsbt') is fixed to 100. For further explanation see the text.

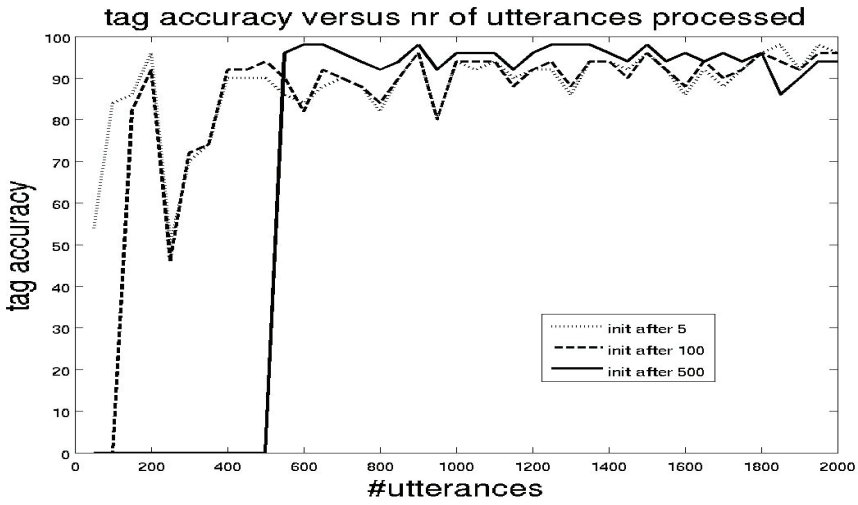


Fig. 4b. In this figure, the performance of the learner is shows as a function of the number of utterances used for the first NMF update ('init'). For the sake of comparison, the memory length is chosen to be equal to 500. The dashed curve in this figure is comparable to the solid curve in figure 4a (ml = 500, number of stimuli used in first NMF factorisation = 100). One observes that the eventual learner result is only slightly dependent on the amount of data used in the initialisation of W and H.

The training data are taken from the Dutch database and consists of 2000 utterances, 500 utterances randomly chosen from each speaker. The visual tags that are associated to the utterances now differ from the tags used in the two previous experiments. While in those experiments the tag was a unique reference to an object, such as 'ball', the tags in this experiment are a combination of the object referred to (ball) *and* the speaker. That means that the leaner has to create and distinguish speaker-dependent representations for all 'words', leading to 36 different columns in the W matrix (the nine common words x four different speakers). As a result, each column encodes a speaker-dependent variant of a target word. For example, for the single target word 'luier' (diaper), 4 columns in W represent the speaker-dependent acoustic realisations as produced by the four speakers.
The question in this experiment is to what extent the W columns can be clustered such that the speaker-dependent variants of a single word can be interpreted as belonging to one cluster.
All representations are one-to-one with columns in W. The metric of the vector space in which these columns reside is defined by the symmetrised Kullback-Leibler divergence. This means that for any vector pair (v_1, v_2) the distance KL(v_2, v_2) can be used as a dissimilarity measure, resulting in a KL-distance matrix M_{KL}. A 10-means clustering using M_{KL} then yields 10 clusters (where each cluster contains one or more word-speaker representations).
Eventually, we obtained clusters that correspond almost perfectly to speaker-independent word representations. Figure 5 shows how the between-cluster distance increases while the

average within-cluster variance decreases during training. This implies that clusters do emerge from the entire set of representations, which indicates that NMF is able to group speaker-dependent word representations one more abstract representation.
One interesting aspect to address here is the precise evaluation of the within and between-cluster variances. This is not trivial, since the KL divergence in the vector space spanned by the columns of W is not Euclidean, meaning that the concept of 'mean' vector is problematic. To circumvent this, the symmetrised KL divergence was first used to define a distance between any two vectors in the space spanned by the columns of W. Next, evaluation of the mean vector was avoided by making use of the following property:

$$\sum_i (x_i - < x >)(x_i - < x >)^t = 0.5 \sum_{ij} (x_i - x_j)(x_i - x_j)^t$$

Application of this expression for both within and between cluster variances leads to the results as shown in Figure 5.

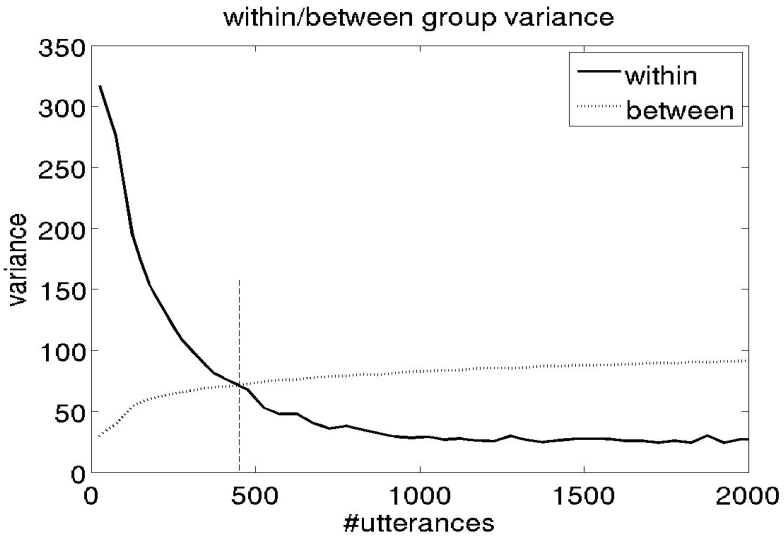


Fig. 5. Values of the between-cluster variance (dotted line) and within-cluster variance (bold line) during training. The ratio of the within-variance and between-variance decreases. This shows that the speaker-dependent word representations can indeed be clustered into groups that become increasingly more distinct.

6. Discussion

The computational model presented here shows that words (and word-like entities) can be discovered without the need for a lexicon that is already populated. This discovery mechanism uses two very general learning principles that also play a role in language acquisition: the repetitive character of infant-directed speech on the one hand, and cross-modal associations in the speech and visual input on the other hand.

The use of the term 'word' in the context of discovery may be a bit misleading, due to the meanings of the term in linguistics. Throughout this chapter 'word' means an entity of which an acoustic realisation is present across utterances as a stretch of speech.

Given a database consisting of 8000 utterances, we showed that our learning model is able to build and update representations of 13 different target words. Experiment 1 shows that these representations are speaker dependent: When the learner is confronted with a new speaker, the model must adapt its internal representation to the characteristics of the speech of the new speaker. A computational model like we are building allows us to look inside the representation space and to investigate the dynamic behaviour of representations during learning.

Experiment 2 showed that the actual performance of the learner depends on two parameters that determine when and how the internal representations are updated. The amount of utterances that is used for each update of internal representations relates to the amount of memory that can be kept active during training. The result in experiment 2 suggests that 10 to 50 observations must be kept in memory for building adequate representations of words. The second result of experiment 2 shows that the amount of data used for bootstrapping the NMF decomposition is not crucial for the eventual performance of the learner. This means that learning can be considered as a truly ongoing process, operating directly from the first stimulus.

The third experiment showed that the representations are changing continuously, and that the representation space can be investigated in detail. A clustering of the columns of W showed how speaker-dependent word representations can be grouped into clusters that correspond almost 1-1 with speaker-independent word representations.

The conceptual consequences of this result are very interesting. In the literature on mental representations of words and the status of phonemes in the prelexical representation (see e.g. McQueen, 2007) there is considerable discussion about the level of abstractness that must be assumed in the word representations. Although based on a simple database and a simple word discovery scheme, the result in experiment 3 suggests how abstraction may follow as a result of competition between crowded collections of representations on a lower level. If needed, speaker-dependent word representations can be clustered such that the common characteristics of these representations combine into one unique word representation.

The current word discovery approach does not use the ordering of the words in an utterance. The utterances 'the ball is red' and 'the red is ball' would be mapped onto the same vector (there are small differences that are not relevant for this discussion). This seems an undesirable property of a word discovery algorithm, especially when the acquisition of syntax is a next step in language acquisition (cf. Saffran and Wilson, 2003). Current research shows that NMF *is* able to recover information about word order by augmenting the output of the map function with additional components related to the relative position of words in the input. A discussion about this approach is outside the scope of this paper.

Since the computational model aims at simulating word discovery as it could happen in human language acquisition, the cognitive plausibility of the model is an important evaluation criterion. The literature on language learning and word acquisition discusses a number of phenomena.

Firstly, the number of words that young infants understand increases over time, with a 'word spurt' between the age 1y and 2y. This word spurt is generally attributed to various factors such as effective reuse of existing representations (but other factors may play a role, see McWhinney, 1998). In the current experiments, a word spurt effect is not yet shown. The way in which internal representations are built, however, paves the way to investigate whether a word spurt effect can be (at least partly) explained by the efficient reuse of already-trained internal representations. If the representations space becomes too crowded, this may be a trigger for the learner to look for a more efficient encoding of the stored information, with a better (more efficient) decoding of new words as a possible result.

In the language acquisition literature, a few more characteristics of language learning are discussed of which the modelling will be a challenge for all models that are ultimately based on statistics. One of these characteristics is that infants reach a stage in which they need just a few examples to learn a new word. Apparently, a reliable representation can be built on the basis of a few tokens only. Our model is in principle able to do that, but to what extent this is dependent on other factors remains to be investigated. Investigations about how a training could be performed on the basis of single tokens (or just a few tokens) will help to understand to what extent the human speech decoding process deviates from a purely Bayesian model.

Another characteristic of first language acquisition is a phenomenon referred to as *fast mapping*. Broadly speaking, fast mapping means that children learn that 'new' (unobserved) words are likely to refer to 'so far unobserved' objects. Apparently the formation of form-referent pairs is a process that might be controlled by some economic rules (in combination with statistically motivated updates of representations). For example, it may imply that an utterance that cannot be understood (fully parsed) given the current representations inspires the learner to postulate a new word-referent pair. However, we want to avoid an ad-hoc approach, in the sense that we want to avoid that the computational model is able to reproduce the effects due to a pre-thought scheme in the implementation. Instead, the fast mapping may result from the use of an underlying rule e.g. based on efficient reuse of representations or on efficient interpretation of the stimulus. The phenomenon of fast mapping will be topic of experiments in the near future.

Our last discussion point relates to the use of visual/semantic tags in the multimodal databases. In the experiments reported in this chapter, tags serve as an abstract representation of the object in the scene that the utterance relates to. The tags are now interpreted by the computational model as they are, without any uncertainty that might obscure its precise interpretation. This might be regarded as undesirable, since it favours the visual information compared to the auditory input (which is subject to variation and uncertainty). Moreover, it is not realistic to assume that the visual system is able to come up with unambiguous and invariant tags.

In the near future the computational model will be extended with a component that allows us to present 'truly' multimodal stimuli, comprising of an audio component and 'visual/semantic' component. The visual/semantic component will then replace the tag that was used in the current databases. For example: the tag 'ball' will be replaced by a vector of binary components, each of them indicated the presence or absence of a certain primitive visual feature (such as red-ness, blue-ness, round-ness).

7. Conclusion

We presented a computational model of word discovery as the first step in language acquisition. The word representations emerge during training without being specified a priori. Word-like entities are discovered without the necessity to first detect sub-word units. The results show that 13 target words can be detected with an accuracy of 95-98 percent by using a database of 8000 utterances spoken by 4 speakers (2000 utterances per speaker).

Future research will enhance the model such that information about word ordering can be obtained. Also the multi-modal information in the stimuli will be enriched to encode visual/semantic information in a cognitively more plausible way.

8.Acknowledgements

This research was funded by the European Commission under contract FP6-034362 (ACORNS).

9. References

Aslin, R.N., Saffran, J.R., Newport, E.L. (1998). Computation of probability statistics by 8-month-old infants. Psychol Sci 9, pp. 321-324.

Bacchiani, M. (1999). Speech recognition system design based on automatically derived units. PhD Thesis, Boston University (Dept. of Electrical and Computer Engineering) (available on-line).

Baddeley, A.D. (1986). Working Memory Clarendon Press, Oxford.

Bosch, L. ten (2006). Speech variation and the use of distance metrics on the articulatory feature space. ITRW Workshop on Speech Recognition and Intrinsic Variation, Toulouse.

Bosch, L. ten, and Cranen, B. (2007). An unsupervised model for word discovery. Proceedings Interspeech 2007, Antwerp, Belgium.

Bosch, L. ten, Van hamme, H., Boves, L. (2008). A computational model of wlanguage acquisition: focus on word discovery. Proceedings Interspeech 2008, Brisbane, Australia.

Bourlard, H., Hermansky, H., Morgan, N. (1996). Towards increasing speech recognition error rates. Speech Communication, Volume 18, Issue 3 (May 1996), 205--231.

Boves, L., ten Bosch, L. and Moore, R. (2007). ACORNS - towards computational modeling of communication and recognition skills , in Proc. IEEE conference on cognitive informatics, pages 349-356, August 2007.

Gaskell, M. G. (2007). Statistical and connectionist models of speech perception and word recognition. In M. G. Gaskell (Ed.), The Oxford Handbook of Psycholinguistics, pp. 55-69, Oxford University Press, Oxford, 2007.

George, D. and Hawkins, J. (2005) A Hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex. Proceedings of the International Joint Conference on Neural Networks (IJCNN 05).

Gerken, L., and Aslin, R.N. (2005). Thirty years of research in infant speech perception: the legacy of Peter Jusczyk. Language Learning and Development, 1: 5-21.

Goldinger, S.D. (1998). Echoes of echoes? An episodic theory of lexical access. Psychological Review, Vol. 105, 251-279.

Gopnik, A., Meltzoff, A.N., and Kuhl, P K. (2001). The Scientist in the Crib, New York: William Morrow Co. Hawkins, J. (2004) On Intelligence. New York: Times Books.

Hoyer, P. (2004). Non-negative matrix factorisation with sparseness constraints. Journal of Machine Learning Research 5. Pp. 1457-1469.

Johnson, E.K., Jusczyk, P.W. (2001). Word segmentation by 8-month-olds: when speech cues count more than statistics. J Mem Lang 44:548-567.

Johnson, S. (2002). Emergence. New York: Scribner.

Jusczyk, P.W. (1999). How infants begin to extract words from speech. TRENDS in Cognitive Science, 3: 323-328.

Kuhl, P.K. (2004). Early language acquisition: cracking the speech code. Nat. Rev. Neuroscience, 5: 831 843.

Lee, C.-H. (2004). From Knowledge-Ignorant to Knowledge-Rich Modeling: A New Speech Research Paradigm for Next Generation Automatic Speech Recognition. Proc. ICSLP.

Lippmann, R. (1997). Speech Recognition by Human and Machines. Speech Communication, 22: 1-14.

Luce, P.A and Lyons, E.A. (1998) Specificity of memory representations for spoken words, Mem Cognit.,26(4): 708-715.

Maslow, A. (1954). Motivation and Personality New York: Harper & Row.

McClelland, J. L. and Elman, J. L. (1986). The TRACE model of speech perception. Cognitive Psychology, Vol. 18, 1986, pp. 1-86.

McQueen, J. M. (2007). Eight questions about spoken-word recognition. In M. G. Gaskell (Ed.), The Oxford Handbook of Psycholinguistics, pp. 37-53, Oxford University Press, Oxford, 2007.

Moore R. K. (2003). A comparison of the data requirements of automatic speech recognition systems and human listeners, Proc. EUROSPEECH'03, Geneva, pp. 2582-2584, 1-4.

Moore, R. K. and Cunningham, S. P. (2005). Plasticity in systems for automatic speech recognition: a review, Proc. ISCA Workshop on 'Plasticity in Speech Perception, pp. 109-112, London, 15-17 June (2005).

Newport, E.L. (2006). Statistical language learning in human infants and adults. Plenary addressed at Interspeech 2006, Pittsburgh, USA (Sept. 2006).

Norris, D. (1994). Shortlist: A connectionist model of continuous speech recognition. Cognition, Vol. 52, 1994, pp. 189-234.

Norris, D. and McQueen, J. M. (2008). Shortlist B: A Bayesian model of continuous speech recognition. Psychological Review 115(2), pp.357-395.

Ostendorf, M. (1999). Moving beyond the beads-on-a-string model of speech. In: Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop. Vol. 1. Keystone, Colorado, USA, pp. 79-83.

Park A., and Glass, J. (2006). Unsupervised word acquisition from speech using pattern discovery. Proceedings ICASSP-2006, Toulouse, France, pp. 409--412.

Pisoni, D. B. and Levi, S. V. (2007). Representations and representational specificity in speech perception and spoken word recognition. In M. G. Gaskell (Ed.), The Oxford Handbook of Psycholinguistics, pp. 3-18, Oxford University Press, Oxford, 2007.

Pitt, M.A., Myung, I. J. and Zhang, S. (2002). Toward a method of selecting among computational models of cognition. Psychological Review, Vol. 109, 2002, pp. 472-491.

Prince C.G. and Hollich, G. J. (2005). Synching infants with models: a perceptual-level model of infant synchrony detection. The Journal of Cognitive Systems Research, 6, pp. 205-228.

Roy, D., and Pentland, A. (2002). Learning words from sights and sounds: A computational model. Cognitive Science, 26 (1), 113-146.

Saffran, J.R., Aslin, R.N., and Newport, E.L. (1996). Statistical learning in 8-month-old infants, Science, 274, December, pp. 1926-28.

Saffran, J.R., Wilson, D.P. (2003). From syllables to syntax: multilevel statistical learning by 12-month-old infants. Infancy 4:273--284.

Scharenborg O., Norris, D., ten Bosch, L., and McQueen, J. M. (2005). How should a speech recognizer work? Cognitive Science, Vol. 29, pp. 867-918.

Shi, R., Oshima-Takane, Y., and Marquis A. (2008). Word-meaning association in early language development. Brain and Cognition. Volume 67, Supplement 1, June 2008, Pages 38-39.

Smith, L.B. and Yu C. (2008). Infants rapidly learn word-referent mappings via cross-situational statistics. Cognition 106(3). Pp 1558-1568.

Snow, C. and Ferguson, C. (1977). Talking to children: language input and acquisition. Cambridge, New York: Cambridge University Press.

Sroka, J. J. and Braida, L. D. (2005). Human and machine consonant recognition, Speech Communication: 44, 401-423.

Stouten, V., Demuynck, K., and Van hamme, H. (2007). Automatically Learning the Units of Speech by Non-negative Matrix Factorisation. In Proc. European Conference on Speech Communication and Technology, pages 1937-1940, Antwerp, Belgium.

Stouten, V., Demuynck, K., and Van hamme, H. (2008). Discovering Phone Patterns in Spoken Utterances by Non-negative Matrix Factorisation. IEEE Signal Processing Letters, volume 15, pages 131-134.

Swingley, D. (2005). Statistical clustering and the contents of the infant vocabulary. Cognitive Psychology, 50: 86-132.

Thiessen, E. D., Hill, E.A., and Saffran J.R. (2005). Infant-Directed Speech Facilitates Word Segmentation. Infancy, 7(1), 53--71

Thiessen, E.D., Saffran, J.R. (2003) When cues collide: use of stress and statistical cues to word boundaries by 7- to 9-month-old infants. Dev Psychol 39:706--716.

Wang, Y. (2003). Cognitive Informatics: A new transdiciplinary research field. Brain and Mind, 4: 115-127.

Wang, Y. (2007). On cognitive informatiocs foundations of knowledge and formal knowledge systems. 6th international conference on cognitive informatics, Lake Tahoe, CA, USA, August 6-8, 2007. pp. 263-272.

Werker, J.F. and Yeung, H.H. (2005). Infant speech perception bootstraps word learning. TRENDS in Cognitive Science, 9: 519-527.

Wesker, T., Meyer, B., Wagener, K., Anemueller, J., Mertins, A. and Kollmeier, B. (2005). Oldenburg logatome speech corpus (ollo) for speech recognition experiments with humans and machines. Proc. of Interspeech, Lisboa.

12

Automatic Speech Recognition via N-Best Rescoring using Logistic Regression

Øystein Birkenes[1], Tomoko Matsui[2],
Kunio Tanabe[3] and Tor André Myrvoll[1]
[1]*Norwegian University of Science and Technology (NTNU), Trondheim,*
[2]*The Institute of Statistical Mathematics, Tokyo,*
[3]*Waseda University, Tokyo,*
[1]*Norway*
[2,3]*Japan*

1. Introduction

Automatic speech recognition is often formulated as a statistical pattern classification problem. Based on the optimal Bayes rule, two general approaches to classification exist; the generative approach and the discriminative approach. For more than two decades, generative classification with hidden Markov models (HMMs) has been the dominating approach for speech recognition (Rabiner, 1989). At the same time, powerful discriminative classifiers like support vector machines (Vapnik, 1995) and artificial neural networks (Bishop, 1995) have been introduced in the statistics and the machine learning literature. Despite immediate success in many pattern classification tasks, discriminative classifiers have only achieved limited success in speech recognition (Zahorian et al., 1997; Clarkson & Moreno, 1999). Two of the difficulties encountered are 1) speech signals have varying durations, whereas the majority of discriminative classifiers operate on fixed-dimensional vectors, and 2) the goal in speech recognition is to predict a sequence of labels (e.g., a digit string or a phoneme string) from a sequence of feature vectors without knowing the segment boundaries for the labels. On the contrary, most discriminative classifiers are designed to predict only a single class label for a given feature.

In this chapter, we present a discriminative approach to speech recognition that can cope with both of the abovementioned difficulties. Prediction of a class label from a given speech segment (speech classification) is done using logistic regression incorporating a mapping from varying length speech segments into a vector of regressors. The mapping is general in that it can include any kind of segment-based information. In particular, mappings involving HMM log-likelihoods have been found to be powerful.

Continuous speech recognition, where the goal is to predict a sequence of labels, is done with N-best rescoring as follows. For a given spoken utterance, a set of HMMs is used to generate an N-best list of competing sentence hypotheses. For each sentence hypothesis, the probability of each segment is found with logistic regression as outlined above. The segment probabilities for a sentence hypothesis are then combined along with a language model score in order to get a new score for the sentence hypothesis. Finally, the N-best list is reordered based on the new scores.

The chapter is organized as follows. In the next section, we introduce some notation and present logistic regression in a general pattern classification framework. Then, we show how logistic regression can be used for speech classification, followed by the use of logistic regression for continuous speech recognition with N-best rescoring. Finally, we present experimental results on a connected digit recognition task before we give a short summary and state the conclusions.

2. Pattern classification and logistic regression

In pattern classification, we are interested in finding a decision rule h, which is a mapping from the set of observations $\mathcal{X}$ to the set of labels $\mathcal{Y}$. Depending on the application, an observation $x \in \mathcal{X}$ can be a vector of features, or it can have a more complex form like a sequence of feature vectors. The latter is the most common way of representing a speech segment (Rabiner, 1989). A label y is usually denoted as a natural number in the finite set $\mathcal{Y} \in \{1,\ldots,K\}$ of class labels. In speech classification, for example, there are typically $K = 39$ class labels representing phonemes.

If the joint probability distribution $p(x,y)$ of observations and labels were known, the optimal decision rule would be the Bayes decision rule (Berger, 1985), which is

$$\begin{aligned} \hat{y} &= \arg\max_{y \in \mathcal{Y}} p(y \mid x) \\ &= \arg\max_{y \in \mathcal{Y}} p(x \mid y) p(y). \end{aligned} \tag{1}$$

In practical applications, however, we usually do not know any of the above probability distributions. One way to proceed is to estimate the distributions from a set $\mathcal{D} = \{(x_1, y_1),\ldots,(x_L, y_L)\}$ of samples referred to as training data. Bayes decision rule can then be approximated in two ways. The first way is to estimate the two distributions $p(x \mid y)$ and $p(y)$, and substitute these into the second line in (1), an approach called the generative approach. The second way is to estimate $p(y \mid x)$, and substitute this into the first line in (1), an approach called the discriminative approach.

Logistic regression is a statistically well-founded discriminative approach to classification. The conditional probability of a class label given an observation is modeled with the multivariate logistic transform, or softmax function, defined as (Tanabe, 2001a,b)

$$\hat{p}(y = k \mid x, W, \Lambda) = \frac{e^{f_k(x,W,\Lambda)}}{\sum_{i=1}^{K} e^{f_i(x,W,\Lambda)}}. \tag{2}$$

In the above equation, f_i is a linear combination (plus a bias term) of M regressors $\phi_1(x,\lambda_1),\ldots,\phi_M(x,\lambda_M)$, with hyperparameters $\Lambda = \{\lambda_1,\ldots,\lambda_M\}$, i.e.,

$$\begin{aligned} f_i(x, W, \Lambda) &= w_{0i} + w_{1i}\phi_1(x,\lambda_1) + \cdots + w_{Mi}\phi_M(x,\lambda_M) \\ &= w_i^{\mathrm{T}} \phi(x, \Lambda), \end{aligned} \tag{3}$$

with $\phi(x,\Lambda) = [1, \phi_1(x,\lambda_1),\ldots,\phi_M(x,\lambda_M)]^{\mathrm{T}}$ and $w_i = [w_{0i},\ldots,w_{Mi}]^{\mathrm{T}}$. The parameters of the model are the elements of the $(M+1) \times K$ dimensional weight matrix

$$W = \begin{pmatrix} | & & | \\ w_1 & \cdots & w_K \\ | & & | \end{pmatrix}. \tag{4}$$

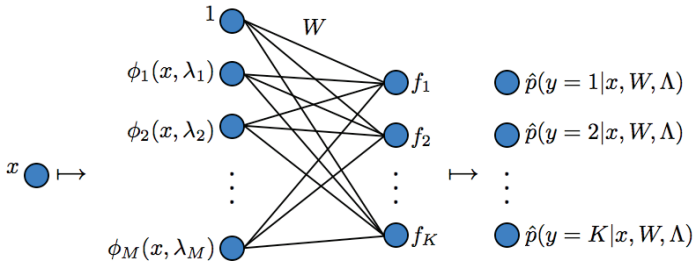


Fig. 1. The logistic regression model.

Due to the probability constraint $\sum_{\kappa=1}^{K} \hat{p}(y=k\,|\,x,W,\Lambda)=1$, the weight vector for one of the classes, say w_K, need not be estimated and can be set to all zeros. Here however, we follow the convention in (Tanabe, 2001a,b) and keep the redundant representation with K non-zero weight vectors. As explained in (Tanabe, 2001a,b), this is done for numerical stability reasons, and in order to treat all the classes equally.

We can think of the model for the conditional probability of each class k given an observation x as a series of transforms of x as illustrated in Fig. 1. First, x is transformed into a vector $\phi(x,\Lambda)$ of M regressors augmented with a "1". Then a linear transform $f = W^{\mathrm{T}}\phi(x,\Lambda)$ gives the elements of the K-dimensional vector f, which are subsequently used in the multivariate logistic transform in order to obtain the conditional probabilities $\hat{p}(y=k\,|\,x,W,\Lambda)$.

The classical way to estimate W from a set of training data $\mathcal{D}$ is to maximize the likelihood, or equivalently, minimize the negative log-likelihood

$$l(W;\mathcal{D}) = -\sum_{l=1}^{L} \log \hat{p}(y=y_l\,|\,x_l,W,\Lambda). \tag{5}$$

However, the maximum likelihood estimate does not always exist (Albert & Anderson, 1984). This happens, for example, when the mapped data set $\{(\phi(x_1;\Lambda),y_1),\ldots,(\phi(x_L;\Lambda),y_L)\}$ is linearly separable. Moreover, even though the maximum likelihood estimate exists, overfitting to the training data may occur, which in turn leads to poor generalization performance. For this reason, we introduce a penalty on the weights and find an estimate $\hat{W}$ by minimizing the penalized negative log-likelihood (Tanabe, 2001a,b)

$$pl_{\delta}(W;\mathcal{D}) = -\sum_{l=1}^{L} \log \hat{p}(y=y_l\,|\,x_l,W,\Lambda) + \frac{\delta}{2}\mathrm{trace}\Gamma W^{\mathrm{T}}\Sigma W, \tag{6}$$

where $\delta \geq 0$ is a hyperparameter used to balance the likelihood and the penalty factor. The $K \times K$ diagonal matrix Γ compensates for differences in the number of training examples from each class, as well as include prior probabilities for the various classes. If we let L_k

denote the number of training examples from class k, and $\hat{p}(y=k)$ denote our belief in the prior probability for class k, we let the k th element of Γ be

$$\gamma_k = \frac{L_k}{L\hat{p}(y=k)}. \tag{7}$$

The $(M+1)\times(M+1)$ matrix Σ is the sample moment matrix of the transformed observations $\phi(x_l;\Lambda)$ for $l=1,\ldots,L$, that is,

$$\Sigma = \Sigma(\Lambda) = \frac{1}{L}\sum_{l=1}^{L}\phi(x_l;\Lambda)\phi^{\mathrm{T}}(x_l;\Lambda)\,. \tag{8}$$

It can be shown (Tanabe, 2001a) that $pl_\delta(W;\mathcal{D})$ is a matrix convex function with a unique minimizer W^*. There is no closed-form expression for W^*, but an efficient numerical method of obtaining an estimate was introduced in (Tanabe, 2001a,b, 2003). In this algorithm, which is called the penalized logistic regression machine (PLRM), the weight matrix is updated iteratively using a modified Newton's method with stepsize α_i, where each step is

$$W_{i+1} = W_i - \alpha_i \Delta W_i\,, \tag{9}$$

where ΔW_i is computed using conjugate gradient (CG) methods (Hestenes & Stiefel, 1952; Tanabe, 1977) by solving the equation (Tanabe, 2001a,b)

$$\sum_{l=1}^{L}\phi_l\phi_l^{\mathrm{T}}\Delta W_i(\operatorname{diag} p_l - p_l p_l^{\mathrm{T}}) + \delta\Sigma\Delta W_i\Gamma = \Phi(P^{\mathrm{T}}(W_i) - Y^{\mathrm{T}}) + \delta\Sigma W_i\Gamma\,. \tag{10}$$

In the above equation, Φ is the $(M+1)\times L$ matrix whose l th column is $\phi_l = \phi(x_l;\Lambda)$, $P(W)$ is a $K\times L$ matrix whose l th column is $p_l = [\hat{p}(y=1\,|\,x_l,W,\Lambda),\ldots,\hat{p}(y=K\,|\,x_l,W,\Lambda)]^{\mathrm{T}}$, and Y is a $K\times L$ matrix where the l th column is a unit vector with all zeros except y_l which is 1.

2.1 Adaptive regressor parameters

Additional discriminative power can be obtained by treating Λ as a set of free parameters of the logistic regression model instead of a preset fixed set of hyperparameters (Birkenes et al., 2006a). In this setting, the criterion function can be written

$$pl_\delta(W,\Lambda;\mathcal{D}) = -\sum_{l=1}^{L}\log\hat{p}(y=y_l\,|\,x_l,W,\Lambda) + \frac{\delta}{2}\operatorname{trace}\Gamma W^{\mathrm{T}}\Sigma(\Lambda)W\,, \tag{11}$$

which is the same as the criterion in (6), but with the dependency on Λ shown explicitly. The goal of parameter estimation is now to find the pair (W^*,Λ^*) that minimizes the criterion in (11). This can be written mathematically as

$$(W^*,\Lambda^*) = \arg\min_{(W,\Lambda)} pl_\delta(W,\Lambda;\mathcal{D})\,. \tag{12}$$

As already mentioned, the function in (11) is convex with respect to W if Λ is held fixed. It is not guaranteed, however, that it is convex with respect to Λ if W is held fixed. Therefore, the best we can hope for is to find a local minimum that gives good classification performance.
A local minimum can be obtained by using a coordinate descent approach with coordinates W and Λ. The algorithm is initialized with Λ_0. Then the initial weight matrix is found as

$$W_0 = \arg\min_W pl_\delta(W, \Lambda_0; \mathcal{D}) . \tag{13}$$

The iteration step is as follows:

$$\begin{aligned} \Lambda_{i+1} &= \arg\min_\Lambda pl_\delta(W_i, \Lambda; \mathcal{D}) \\ W_{i+1} &= \arg\min_W pl_\delta(W, \Lambda_{i+1}; \mathcal{D}). \end{aligned} \tag{14}$$

The coordinate descent method is illustrated in Fig. 2.

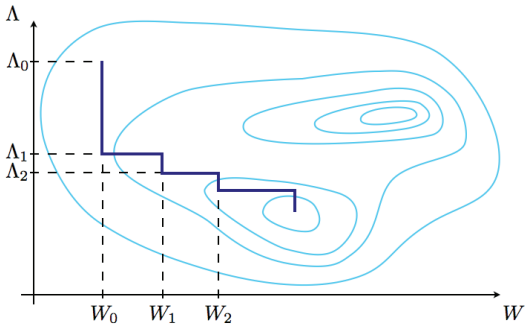


Fig. 2. The coordinate descent method used to find the pair (W^*, Λ^*) that minimizes the criterion function $pl_\delta(W, \Lambda; \mathcal{D})$.

For the convex minimization with respect to W, we can use the penalized logistic regression machine (Tanabe, 2001a,b). As for the minimization with respect to Λ, there are many possibilities, one of which is the RProp method (Riedmiller and Braun, 1993). In this method, the partial derivatives of the criterion with respect to the elements of Λ are needed. These calculations are straightforward, but tedious. The interested reader is referred to (Birkenes, 2007) for further details.
When the criterion function in (11) is optimized with respect to both W and Λ, overfitting of Λ to the training data may occur. This typically happens when the number of free parameters in the regressor functions is large compared to the available training data. By keeping the number of free parameters in accordance with the number of training examples, the effect of overfitting may be reduced.

2.2 Garbage class

In some applications, the classifier will be presented with observations x that do not correspond to any of the classes in the label set $\mathcal{Y}$. In this situation, the classifier should return a small probability for every class in $\mathcal{Y}$. However, this is made impossible by the fact

that the total probability should sum to 1, that is, $\sum_{y\in\mathcal{Y}} p(y \mid x) = 1$. The solution to this problem is to introduce a new class $y = K+1 \in \mathcal{Y}_o = \mathcal{Y} \cup \{K+1\}$, called a garbage class, that should get high conditional probability given observations that are unlikely for the classes in $\mathcal{Y}$, and small probability otherwise (Birkenes et al., 2007).

In order to train the parameters of the logistic regression model with such a garbage class, a set of observations labeled with a garbage label, or garbage observations, are needed. For applications with a low-dimensional observation set $\mathcal{X}$, these garbage observations can be drawn from a uniform distribution over $\mathcal{X}$. For many practical applications however, $\mathcal{X}$ has a very high dimensionality, so an unreasonably high number of samples must be drawn from the uniform distribution in order to achieve good performance. In such cases, prior knowledge of the nature or the generation of the possible garbage observations that the classifier will see during prediction is of great value. We will soon see how we can use N-best lists to generate garbage observations for continuous speech recognition.

3. Classification of speech segments with logistic regression

In this section we will be concerned with the modeling of the conditional distribution $p(y \mid x)$ using the logistic regression model, where each observation $x = (o_1, \ldots, o_{T_x})$ is a sequence of feature vectors extracted from a speech segment and y is a word label. Since the observation x is here a sequence of feature vectors that can vary in length, the logistic regression mapping $\phi : \mathcal{X} \to \mathbb{R}^{M+1}$ is a map from the set $\mathcal{X}$ of all such observations x into the Euclidean space $\mathbb{R}^{M+1}$ containing all regressor vectors $\phi(x;\Lambda)$. The mapping should be able to map observations of varying lengths into fixed dimensional vectors while preserving the discriminative information embedded in the observations.

A mapping that has been found to be effective for speech classification makes use of $M = K$ hidden Markov models (HMMs), one for each word in the vocabulary, and is defined as (Birkenes et al., 2006a)

$$\phi(x;\Lambda) = \begin{pmatrix} 1 \\ \frac{1}{T_x}\log \hat{p}(x;\lambda_1) \\ \vdots \\ \frac{1}{T_x}\log \hat{p}(x;\lambda_M) \end{pmatrix}, \tag{15}$$

where $\hat{p}(x;\lambda_m)$ is the Viterbi-approximated likelihood (i.e., the likelihood computed along the Viterbi path) of the mth HMM with parameter vector λ_m. Specifically, if we let $\lambda = (\pi, A, \eta)$ be the set of parameters for an HMM, where π denotes the initial state probabilities, A is the transition matrix, and η is the set of parameters of the state-conditional probability density functions, then

$$\begin{aligned} \hat{p}(x;\lambda) &= \max_q \hat{p}(x,q;\lambda) \\ &= \max_q \pi_{q_1} \prod_{t=2}^{T_x} a_{q_{t-1},q_t} \prod_{t=1}^{T_x} \hat{p}(o_t \mid q_t;\eta_{q_t}), \end{aligned} \tag{16}$$

where $q = (q_1, \ldots, q_{T_x})$ denotes a state sequence. Each state-conditional probability density function is a Gaussian mixture model (GMM) with a diagonal covariance matrix, i.e.,

$$\begin{aligned} \hat{p}(o \mid q; \eta_q) &= \sum_{h=1}^{H} c_{qh} \mathcal{N}(\mu_{qh}, \Sigma_{qh}) \\ &= \sum_{h=1}^{H} c_{qh} (2\pi)^{-D/2} \left(\prod_{d=1}^{D} \sigma_{qhd} \right)^{-1} e^{-\frac{1}{2} \sum_{d=1}^{D} \left(\frac{o_d - \mu_{qhd}}{\sigma_{qhd}} \right)^2}, \end{aligned} \tag{17}$$

where H is the number of mixture components, c_{qh} is the mixture component weight for state q and mixture h, D is the vector dimension, and $\mathcal{N}(\mu, \Sigma)$ denotes a multivariate Gaussian distribution with mean vector μ and diagonal covariance matrix Σ with elements σ_d. The hyperparameter vector of the mapping in (15) consists of all the parameters of all the HMMs, i.e., $\Lambda = (\lambda_1, \ldots, \lambda_M)$.

We have chosen to normalize the log-likelihood values with respect to the length T_x of the sequence $x = (o_1, \ldots, o_{T_x})$. The elements of the vector $\phi(x; \Lambda)$ defined in (15) are thus the average log-likelihood per frame for each model. The reason for performing this normalization is that we want utterances of the same word spoken at different speaking rates to map into the same region of space. Moreover, the reason that we use the Viterbi-approximated likelihood instead of the true likelihood is to make it easier to compute its derivatives with respect to the various HMM parameters. These derivatives are needed when we allow the parameters to adapt during training of the logistic regression model.

With the logistic regression mapping ϕ specified, the logistic regression model can be trained and classification can be performed as explained in the previous section. In particular, classification of an observation x is accomplished by selecting the word $\hat{y} \in \mathcal{Y}$ having the largest conditional probability, that is,

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \hat{p}(y \mid x, W, \Lambda), \tag{18}$$

where

$$\hat{p}(y = k \mid x, W, \Lambda) = \frac{e^{w_k^{\mathrm{T}} \phi(x, \Lambda)}}{\sum_{i=1}^{K} e^{w_i^{\mathrm{T}} \phi(x, \Lambda)}}. \tag{19}$$

Although in this section we only considered probabilistic prediction of words given a speech segment, the theory is directly applicable to subword units such as phones.

4. N-best rescoring using logistic regression

In this section, we consider the continuous speech recognition problem, which amounts to finding the best sequence of subwords, or sentence hypothesis, given a whole utterance of a sentence. A problem we have to deal with in this context is that the segment boundaries are not known. We propose a two step approach: 1) generate an N-best list using a set of HMMs and the Viterbi algorithm (Viterbi, 1983), and 2) rescore the N-best list and select the

sentence hypothesis with the highest score. Rescoring of a sentence hypothesis is done by obtaining probabilities of each subword using logistic regression, and combining the subword probabilities into a new sentence score using a geometric mean. These sentence scores are used to reorder the sentence hypotheses in the N-best list. The recognized sentence hypothesis of an utterance is then taken to be the first one in the N-best list, i.e., the sentence hypothesis with the highest score.

In the following, let us assume that we have a set of HMMs, one for each subword (e.g., a digit in a spoken digit string, or a phone). We will refer to these HMMs as the baseline models and they will play an important role in both the training phase and the recognition phase of our proposed approach for continuous speech recognition using logistic regression. For convenience, we let $z = (o_1,\ldots,o_{T_z})$ denote a sequence of feature vectors extracted from a spoken utterance of a sentence $s = (y_1,\ldots,y_{L_s})$ with L_s subwords. Each subword label y_l is one of $(1,\ldots,K)$, where K denotes the number of different subwords. Given a feature vector sequence z extracted from a spoken utterance s, the baseline models can be used in conjunction with the Viterbi algorithm in order to generate a sentence hypothesis $\hat{s} = (\hat{y}_1,\ldots,\hat{y}_{L_{\hat{s}}})$, which is a hypothesized sequence of subwords. Additional information provided by the Viterbi algorithm is the maximum likelihood (ML) segmentation on the subword level, and approximations to the subword likelihoods. We write the ML

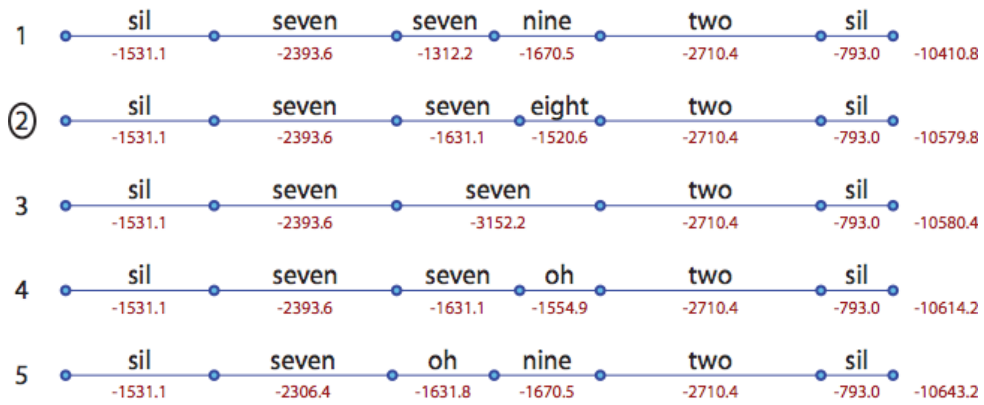


Fig. 3. A 5-best list where the numbers below the arcs are HMM log-likelihood values corresponding to the segments. The total log-likelihood for each sentence hypothesis is shown at the right. The list is sorted after decreasing log-likelihood values for the sentences. The circle around sentence number 2 indicates that this is the correct sentence.

segmentation as $z = (x_1,\ldots,x_{L_{\hat{s}}})$, where x_l denotes the subsequence of feature vectors associated with the l th subword $\hat{y}_l$ of the sentence hypothesis.

For a given utterance, we can use the baseline models to generate an N-best list of the N most likely sentence hypotheses (Schwartz and Chow, 1990). An example of a 5-best list is shown in Fig. 3. The list is generated for an utterance of the sentence "seven, seven, eight, two", with leading and trailing silence. The most likely sentence hypothesis according to the

HMMs appears at the top of the list and is the sentence "seven, seven, nine, two". This sentence differs from the correct sentence, which is the second most likely sentence hypothesis, by one subword. The segmentation of each sentence hypothesis in the list is the most likely segmentation given the sentence hypothesis. Each segment is accompanied with the HMM log-likelihood.
The reason for generating N-best lists is to obtain a set of likely sentence hypotheses with different labeling and segmentation, from which the best sentence hypothesis can be chosen based on additional knowledge. In the following we will first consider how we can obtain reliable subword probabilities given speech segments appearing in N-best lists. We suggest using a garbage class for this purpose. Then, we introduce a method for rescoring N-best lists using these estimated subword probabilities.

4.1 Logistic regression on segments in N-best lists

Provided that the baseline models are reasonably good, many of the segments in the N-best lists are good in the sense that they correspond to a complete utterance of exactly one subword. However, it is inherent that N-best lists frequently contain segments that do not correspond to a complete utterance of exactly one subword. Some segments, for example, correspond to only a part of an utterance of a subword, or even an utterance of several subwords together. Consider again the 5-best list in Fig. 3, where the correct sentence hypothesis appears in position 2. Let us assume that the correct unknown segmentation coincides with the ML segmentation in position 2. Then, the third segment in sentence hypothesis 3 actually corresponds to an utterance of the two connected digits "seven" and "eight" spoken in a sequence. Moreover, for hypotheses 1 and 5, the third segment may not correspond to a complete utterance of "seven", whereas the fourth segment corresponds to an utterance of the last part of "seven" and the whole of "eight". Thus, the segments of an N-best list can be roughly divided into two: good segments and garbage segments.
The role of logistic regression in our N-best rescoring approach is to provide conditional probabilities of subword labels given a segment. Obviously, we want a correct subword label to get high conditional probability given a good segment. This implies that incorrect subword labels will get low probabilities for good segments since the total probability should sum to one. Furthermore, garbage segments should result in low probabilities for all subword labels. For this reason we introduce a garbage class, whose role is to aggregate large probability for garbage segments and low probability otherwise. In the training of the model, we need two sets of training examples; 1) a set of good segments each labeled with the correct subword label, and 2) a set of garbage segments labeled with the garbage label.
Let us first discuss how we can obtain segments from the former set. If the training utterances were segmented on the subword level, i.e., if we knew the segment boundaries of each subword, we could simply use these subword-labeled segments as the training set for the logistic regression model. In most training databases for speech however, the segment boundaries are not known, only the orthographic transcription, i.e., the correct subword sequence. Then, the most straightforward thing to do would be to estimate the segment boundaries. For this, we will make use of the baseline models to perform Viterbi forced alignment (FA) segmentation on the training data. From a pair (z, s) in the training

database, FA segmentation gives us a set $\{(x_1, y_1), \ldots, (x_{L_s}, y_{L_s})\}$ of subword labeled segments. Doing this for all the pairs (z, s) in the training database yields a set

$$\mathcal{D}_{\text{FA}} = \{(x_l, y_l)\}_{l=1,\ldots,L_{\text{FA}}} \tag{20}$$

of all FA-labeled segments.

Extracting garbage segments to be used in the training of the logistic regression model is more difficult. In the rescoring phase, segments that differ somehow from the true unknown segments should give small probability to any class in the vocabulary, and therefore high probability to the garbage class. In order to achieve this, we generate an N-best list for each training utterance, and compare all segments within the list with the corresponding forced alignment generated segments, or the true segments if they are known. The segments from the N-best list that have at least ε number of frames not in common with any of the forced alignment segments, are labeled with the garbage label *K*+1 and used as garbage segments for training. This gives us a set

$$\mathcal{D}_{\text{gar}} = \{(x_l, K+1)\}_{l=1,\ldots,L_{\text{gar}}} \tag{21}$$

of all garbage-labeled segments. The full training data used to train the logistic regression model is therefore

$$\mathcal{D} = \mathcal{D}_{\text{FA}} \bigcup \mathcal{D}_{\text{gar}}. \tag{22}$$

4.2 The rescoring procedure

Now that we have seen how logistic regression can be used to obtain the conditional probability of a subword given a segment, we will see how we can use these probability estimates to rescore and reorder sentence hypotheses of an N-best list.

For a given sentence hypothesis $\hat{s} = (\hat{y}_1, \ldots, \hat{y}_{L_{\hat{s}}})$ in an N-best list with corresponding segmentation $z = (x_1, \ldots, x_{L_{\hat{s}}})$, we can use logistic regression to compute the conditional probabilities $\hat{p}_{\hat{y}_l} = \hat{p}(y = \hat{y}_l \mid x_l, W, \Lambda)$. A score for the sentence hypothesis can then be taken as the geometric mean of these probabilities multiplied by a weighted language model score $\hat{p}(\hat{s})$ as in

$$v_{\hat{s}} = \left(\prod_{l=1}^{L_{\hat{s}}} \hat{p}_{\hat{y}_l} \right)^{1/L_{\hat{s}}} \left(\hat{p}(\hat{s}) \right)^{\beta}, \tag{23}$$

where β is a positive weight needed to compensate for large differences in magnitude between the two factors. In order to avoid underflow errors caused by multiplying a large number of small values, the score can be computed as

$$v_{\hat{s}} = \exp\left\{ \frac{1}{L_{\hat{s}}} \sum_{l=1}^{L_{\hat{s}}} \log \hat{p}_{\hat{y}_l} + \beta \log \hat{p}(\hat{s}) \right\}. \tag{24}$$

When all hypotheses in the N-best list have been rescored, they can be reordered in descending order based on their new score. Fig. 4 shows the 5-best list in Fig. 3 after rescoring and reordering. Now, the correct sentence hypothesis "seven, seven, eight, two" has the highest score and is on top of the list.

Additional performance may be obtained by making use of the log-likelihood score for the sentence hypothesis already provided to us by the Viterbi algorithm. For example, if $\hat{p}(z \mid \hat{s})$ denotes the sentence HMM likelihood, we can define an interpolated logarithmic score as

$$\tilde{v}_{\hat{s}} = (1-\alpha)\frac{1}{L_{\hat{s}}}\sum_{l=1}^{L_{\hat{s}}} \log \hat{p}_{\hat{y}_l} + \alpha \log \hat{p}(z \mid \hat{s}) + \beta \log \hat{p}(\hat{s}) \,, \tag{25}$$

where $0 \le \alpha \le 1$.

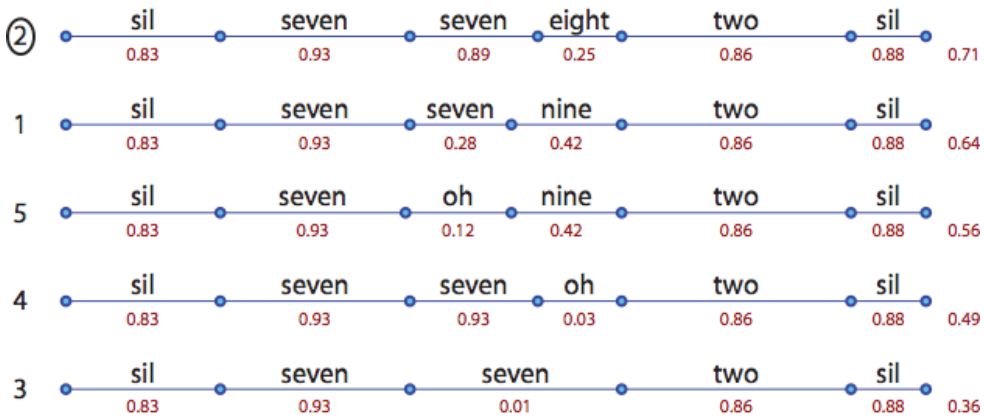


Fig. 4. The 5-best list in Fig. 3 after rescoring using penalized logistic regression with HMM log-likelihood regressors. The hypotheses have been re-ordered according to sentence scores computed from geometric means of the segment probabilities. Sentence number 2, which is the correct one, is now at the top of the list.

5. Experimental results

We performed rescoring of 5-best lists generated by an HMM baseline speech recognizer on the Aurora2 database (Pearce and Hirsch, 2000). We tried both rescoring without a garbage class, and with a garbage class. In the latter experiment, we also interpolated the logistic regression score and the HMM score. In all experiments, a flat language model was used.

5.1 The Aurora2 database and the baseline system

The Aurora2 connected digits database (Pearce & Hirsch, 2000) contains utterances, from different speakers, of digit strings with lengths 1–7 digits. We used only the clean data in both training and testing. The clean data corresponds to the data in the TI-digits database (Leonard, 1984) downsampled to 8 kHz and filtered with a G712 characteristic.

There are 8440 training utterances and 4004 test utterances in the training set and the test set, respectively. The speakers in the test set are different from the speakers in the training set.

From each speech signal, a sequence of feature vectors were extracted using a 25 ms Hamming window and a window shift of 10 ms. Each feature vector consisted of 12 Mel-frequency cepstral coefficients (MFCC) and the frame energy, augmented with their delta and acceleration coefficients. This resulted in 39-dimensional vectors.

Each of the digits 1–9 was associated with one class, while 0 was associated with two classes reflecting the pronunciations "zero" and "oh". The number of digit classes was thus $C = 11$. For each of the 11 digit classes, we used an HMM with 16 states and 3 mixtures per state. In addition, we used a silence (sil) model with 3 states and 6 mixtures per state, and a short pause (sp) model with 1 state and 6 mixtures. These HMM topologies are the same as the ones defined in the training script distributed with the database. We refer to these models as the baseline models, or collectively as the baseline recognition system. The sentence accuracy on the test set using the baseline system was 96.85%.

5.2 Rescoring 5-best lists without a garbage class

Before training the logistic regression model, the training data was segmented using the baseline models with forced alignment. We updated only the means of the HMMs while keeping the other HMM parameters fixed. For each of the coordinate descent iterations we used the Rprop method (Riedmiller & Braun, 1993) with 100 iterations to update the HMM means Λ and the Newton method with 4 iterations to update W. After 30 coordinate descent iterations, the optimization was stopped.

We used the trained logistic regression model to rescore 5-best lists that were generated on the test set by the baseline recognition system. The upper bound on the sentence accuracy inherent in the 5-best lists, i.e., the sentence accuracy obtainable with a perfect rescoring method, was 99.18%. We chose to rescore only those sentence hypotheses in each 5-best list that had the same number of digits as the first hypothesis in the list (Birkenes et al., 2006b). The resulting sentence accuracy was 97.20%.

5.3 Rescoring 5-best lists with a garbage class

We now present results that we achieved with 5-best rescoring with the use of a garbage class in the logistic regression model. The 5-best lists used in the rescoring phase were the same as above. This time the training was done using two sets of segments; correct segments with the correct class label, and garbage segments with the garbage label. The former set was generated using the baseline recognition system with forced alignment on the training data. The garbage segments were generated from 5-best lists on the training data, with $\varepsilon = 10$. Again, we updated only the mean values of the HMMs while keeping the other HMM parameters fixed. The training with the coordinate descent approach was done in the same way as above. Also this time we stopped the optimization after 30 iterations.

The sentence accuracies for $\delta \in \{10^3, 10^4, 10^5, 10^6\}$ are shown in Fig. 5. The baseline accuracy and the accuracy of the 5-best rescoring approach without a garbage class are also shown. We see that our approach with a garbage class gives the best accuracy for the four values of the regularization parameter δ we used in our experiments. For lower values of δ, we

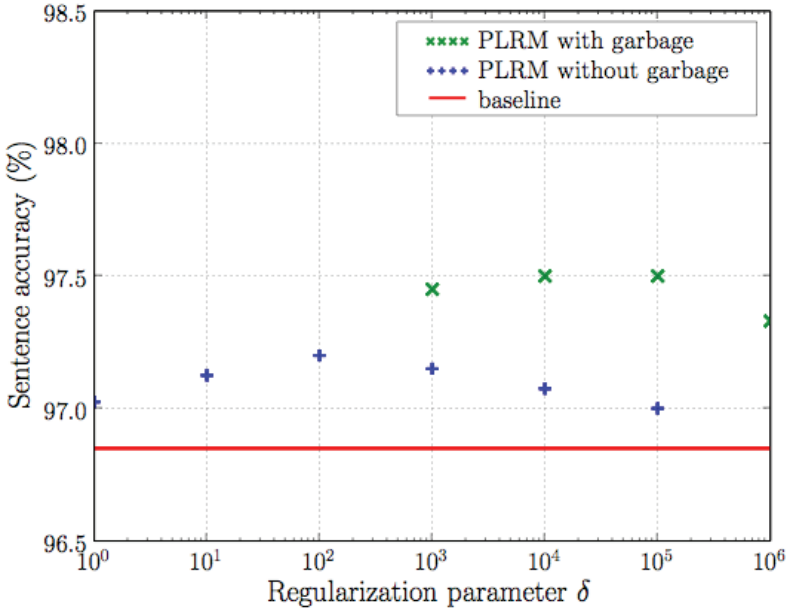


Fig. 5. Sentence accuracy on the test set for various δ.

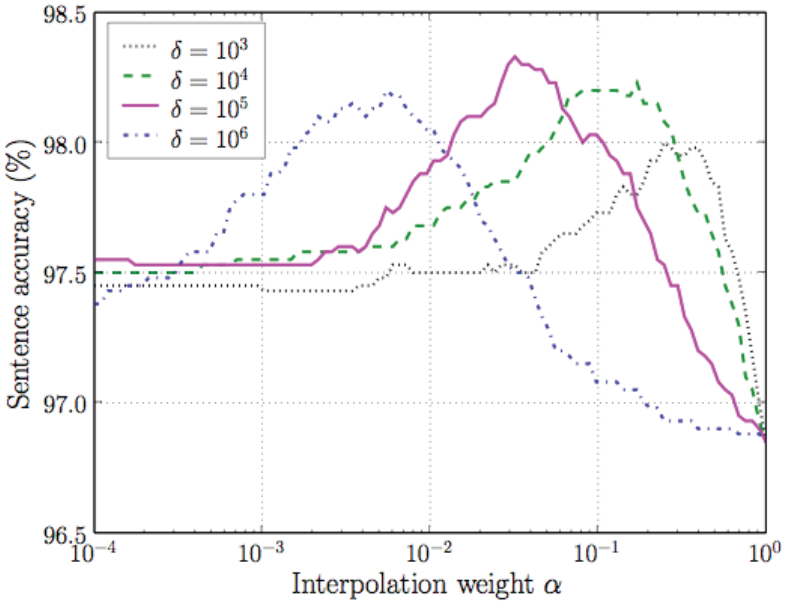


Fig. 6. Sentence accuracy using interpolated scores.

expect a somewhat lower sentence accuracy due to overfitting. Very large δ values are expected to degrade the accuracy since the regression likelihood will be gradually negligible compared to the penalty term.

Fig. 6 shows the effect of interpolating the HMM sentence likelihood with the logistic regression score. Note that with $\alpha = 0$, only the logistic regression score is used in the rescoring, and when $\alpha = 1$, only the HMM likelihood is used. The large gain in performance when taking both scores into account can be explained by the observation that the HMM score and the logistic regression score made very different sets of errors.

6. Summary

A two-step approach to continuous speech recognition using logistic regression on speech segments has been presented. In the first step, a set of hidden Markov models (HMMs) is used in conjunction with the Viterbi algorithm in order to generate an N-best list of sentence hypotheses for the utterance to be recognized. In the second step, each sentence hypothesis is rescored by interpolating the HMM sentence score with a new sentence score obtained by combining subword probabilities provided by a logistic regression model. The logistic regression model makes use of a set of HMMs in order to map variable length segments into fixed dimensional vectors of regressors. In the rescoring step, we argued that a logistic regression model with a garbage class is necessary for good performance.

We presented experimental results on the Aurora2 connected digits recognition task. The approach with a garbage class achieved a higher sentence accuracy score than the approach without a garbage class. Moreover, combining the HMM sentence score with the logistic regression score showed significant improvements in accuracy. A likely reason for the large improvement is that the HMM baseline approach and the logistic regression approach generated different sets of errors.

The improved accuracies observed with the new approach were due to a decrease in the number of substitution errors and insertion errors compared to the baseline system. The number of deletion errors, however, increased compared to the baseline system. A possible reason for this may be the difficulty of sufficiently covering the space of long garbage segments in the training phase of the logistic regression model. This needs further study.

7. References

Albert, A. and Anderson, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika,* 71(1):1-10

Berger, J. (1985). *Statistical Decision Theory and Bayesian Analysis,* Springer-Verlag, 2 edition

Birkenes, Ø. (2007). *A Framework for Speech Recognition using Logistic Regression,* PhD thesis, Norwegian University of Science and Technology (NTNU)

Birkenes, Ø. ; Matsui, T. & Tanabe, K. (2006a). Isolated-word recognition with penalized logistic regression machines, *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Toulouse, France

Birkenes, Ø. ; Matsui, T. ; Tanabe, K. & Myrvoll, T. A. (2006b). Continuous speech recognition with penalized logistic regression machines, *Proceedings of IEEE Nordic Signal Processing Symposium*, Reykjavik, Iceland

Birkenes, Ø. ; Matsui, T. ; Tanabe, K. & Myrvoll, T. A. (2007). N-best rescoring for speech recognition using penalized logistic regression machines with garbage class, *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, USA

Bishop, C. (1995). *Neural Networks for Pattern Recognition,* Oxford University Press

Clarkson, P. & Moreno, P. (1999). On the use of support vector machines for phonetic classification, *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, pp. 585-588

Hestenes, M. R. & Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards,* 49: 409-436

Leonard, R. (1984). A database for speaker independent digit recognition, *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Vol. 3, pp. 42.11

Pearce, D. & Hirsch, H.-G. (2000). The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions, *In ISCA ITRW ASR*, pp. 181-188, Paris, France

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition, *Proceedings of the IEEE*, 77(2):257-286

Riedmiller, M. & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm, *Proceedings of the IEEE Intl. Conf. on Neural Networks*, pp. 586-591, San Francisco, CA

Schwartz, R. & Chow, Y. (1990). The N-best algorithm: an efficient and exact procedure for finding the N most likely sentence hypotheses, *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 81-84, Albuquerque, New Mexico, USA

Tanabe, K. (1977). Conjugate-gradient method for computing the moore-penrose inverse and rank of a matrix. *Journal of Optimization Theory and Applications,* 22(1):1-23

Tanabe, K. (2001a). Penalized logistic regression machines: New methods for statistical prediction 1. *ISM Cooperative Research Report 143,* pp. 163-194

Tanabe, K. (2001b). Penalized logistic regression machines: New methods for statistical prediction 2, *Proceedings of Information-based Induction Sciences (IBIS)*, pp. 71-76, Tokyo, Japan

Tanabe, K. (2003). Penalized logistic regression machines and related linear numerical algebra, *In KOKYUROKU 1320, Institute for Mathematical Sciences*, pp. 239-250, Kyoto, Japan

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory,* Springer, 2 edition

Viterbi, A. (1983). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Information Theory,* 13(4):179-190

Zahorian, S.; Silsbee, P. & Wang, X. (1997). Phone classification with segmental features and a binary-pair partitioned neural network classifier, *Proceedings of IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP)*, Vol. 2, pp. 1011-1014

13

Knowledge Resources in Automatic Speech Recognition and Understanding for Romanian Language

Inge Gavat, Diana Mihaela Militaru and Corneliu Octavian Dumitru
Faculty of Electronics Telecommunications and Information Technology,
University POLITEHNICA Bucharest
Romania

1. Introduction

In this chapter are presented the results obtained in automatic speech recognition and understanding (ASRU) experiments made for Romanian language in the statistical framework, concerning the performance enhancement of two important knowledge resources, namely the acoustical models and the language model. If the ASRU process is for simplicity seen as a two stage process, in the first stage automatic speech recognition (ASR) is done and in the second stage the understanding is accomplished. The acoustical models incorporate knowledge about features statistic in different speech units composing the words and are mostly responsible for the performance of the recognition stage, judged after the WRR (word recognition rate). The language models incorporate knowledge about the word statistic in the phrase and determine mostly the performance of the understanding stage, judged after the PRR (phrase recognition rate). The two considered stages are interrelated and the named performance criteria are interdependent, enhanced WRR leads to PRR enhancement too. In this chapter are exposed methods to enhance the WRR, based on introducing of contextual models like triphones instead monophones or building of gender specialized models (for men, women and children) instead of global models. The methods applied to enhance the PRR are based on introducing of a restrictive finite state grammar instead the permissive word loop grammar or a bigram based language model.

1.1 Short history

Speech recognition and understanding has in Romania also a long history and begins with recognition and synthesis of vowels, done in the University Politehnica from Bucharest around 1963 (Draganescu, 2003). Digit recognizers were built around 1970 in hardware form and in 1976 as software models and the first recognition experiments for continuous speech were successful around 1980 in the Institut for Linguistics of the Romanian Academy. The researches in this new domain of speech technology were extended after 1980 also in other universities and technical universities in cities like Iasi, Cluj – Napoca and Timisoara. To bring researchers together, starting with the year 1999 each two years an international conference namely SPED is organized under the aegis of the Romanian Academy. It is also to be mentioned participation of Romanian researchers in international research programs,

in international conferences, in bilateral cooperations. In 2002 a special issue for Romanian contributions was dedicated by the International Journal of Speech Technology.
Our research group comes from the University Politehnica Bucharest, Chair of Applied Electronics and Information Engineering, Faculty of Electronics, Telecommunications and Information Technology. Our research interests concern mainly ASRU, but also other topics like speech recognition based on neural networks (Valsan et al., 2002), (Gavat & Dumitru, 2002-1), (Gavat et al., 2002-2), (Dumitru & Gavat, 2008) on fuzzy technics (Gavat & Zirra, 1996-2), (Gavat et al., 1997), (Gavat et al., 2001-1), (Gavat et al., 2001-2), or on Support Vector Machines (SVM) (Gavat et al., 2005-2), (Gavat & Dumitru, 2008), speech synthesis, TTS systems, speech and music retrieval (Gavat et al., 2005-1), speech prosody, multimodal systems, could be mentioned. Our main realization is the Automatic Speech Recognition System for Romanian Language, ASRS_RL (Dumitru, 2006), a research platform in order to implement and enhance different methods for ASRU in Romanian language (Gavat and Dumitru, 2008). Recognizers for phonemes, digits and continuous speech acting under the statistic paradigm based on hidden Markov models, under the connectionist paradigm of artificial neural networks or under fuzzy principles, but also under combined principles were experimented.
The system presented in this chapter has as main objective refinement of the statistic paradigm for ASRU in Romanian language, by enhancement of two important aspects, acoustical modeling and language modeling.

1.2 The proposed system

In the statistical approach, for the mathematical formulation of the problem, the recognition process can be modeled as a communication system, depicted in Fig. 1, consisting in four stages: text generation, speech production, acoustic processing, and linguistic decoding.

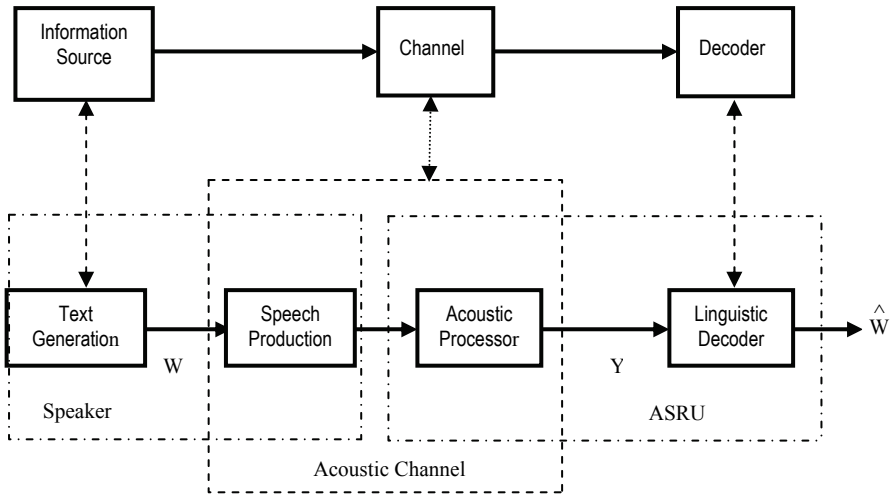


Fig. 1 Structure of continuous speech recognition system

A speaker is assumed a transducer that transforms into speech the text of thoughts to communicate. The delivered word sequence W is converted into an acoustic observation sequence *Y*, with probability *P(W, Y)*, through a noisy acoustical transmission channel, into an

acoustic observations sequence Y which is then decoded to an estimated sequence $\hat{W}$. The goal of recognition is then to decode the word string, based on the acoustic observation sequence, so that the decoded string has the maximum a posteriori probability (Huang et al., 2001):

$$\hat{W} = \arg\max_{W} P(W \mid Y) \tag{1}$$

Using Bayes' rule can be written as:

$$\hat{W} = \arg\max_{W} P(Y \mid W) * P(W) / P(Y) \tag{2}$$

Since $P(Y)$ is independent of W, the maximum a posteriori decoding rule is:

$$\hat{W} = \arg\max_{W} P(Y \mid W) * P(W) \tag{3}$$

The term $P(Y|W)$ is generally called the acoustic model as it estimates the probability of sequence of acoustic observations conditioned on the word string (Rabiner, 1989).
The term $P(Y)$ is generally called the language model since it describes the probability associated with a postulated sequence of words. Such language models can incorporate both syntactic and semantic constraints. When only syntactic constraints are used, the language model is called a grammar.
The block diagram of the system, based on the pattern recognition paradigm, and applied in a continuous speech recognition task is presented in Fig. 2. The speech signal is analysed resulting sequence of feature vectors grouped in linguistic unit patterns. Each obtained pattern is compared with reference patterns, pre-trained and stored with class identities.

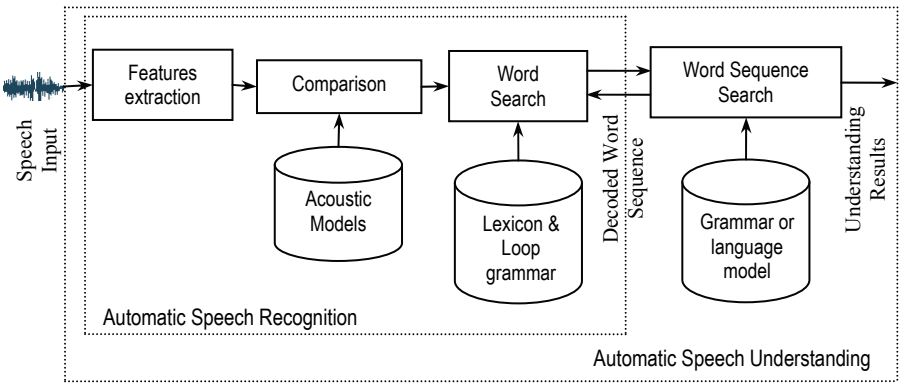


Fig. 2. Block diagram for an automatic speech recognition and understanding system.

These pre-trained patterns, obtained in a learning process are in our system the acoustical models for phonemes, with or without context, and represent a first knowledge source for the word sequence search.
Further, based on the dictionary the words are recognized and a simple loop grammar leads to the estimated word sequence. The important outcome of this stage is the word

recognition rate, the phrase recognition rate having low levels. To enhance the phrase recognition rate, a restrictive grammar or a language model must be applied. Like it is represented in Fig. 1, it is possible to separate the part of automatic speech recognition (ASR) as a first stage in the automatic speech recognition and understanding (ASRU) process (Juang et Furui, 2000).

1.3 Chapter structure

The remainder of this paper will be structured as follows. Section two will be dedicated to the acoustic models as first knowledge source in the ASRU process. In the third section is presented the second knowledge resource, in form of language models and restrictive grammars. Comments and discussions on the experimental results presented in the previous two sections will be made in section four. The final section will be dedicated to conclusions about the work done and to plan future activities.

2. Acoustic models

The acoustic models developed in our experiments are the hidden Markov models (HMM), basic entities in the statistical framework.

2.1 Hidden Markov models

2.1.a Basics monophones and triphones

HMMs are finite automata, with a given number of states; passing from one state to another is made instantaneously at equally spaced time moments. At every pass from one state to another, the system generates observations, two processes taking place: the transparent one represented by the observations string (features sequence), and the hidden one, which cannot be observed, represented by the state string (Gavat et al., 2000).

In speech recognition, the left - right model (or the Bakis model) is considered the best choice. For each symbol, such a model is constructed; a word string is obtained by connecting corresponding HMMs together in sequence (Huang et al., 2001).

For limited vocabulary, word models are widely used, since they are accurate and trainable. In the situation of a specific and limited task they become valid if enough training data are available, but they are typically not generalizable. Usually for not very limited tasks are preferred phonetic models based on monophones (which are phonemes without context), because the phonemes are easy generalizable and of course also trainable.

Monophones constitute the foundation of any training method and we also started with them (as for any language). But in real speech the words are not simple strings of independent phonemes, because each phoneme is affected through the immediately neighboring phonemes by co-articulation. Therefor to monophones context was added leading for example to triphones like monophones with left and right context, that became actually the state of the art in automatic speech recognition and understanding for the large vocabularies (Young, 1992).

Based on the SAMPA (Speech Assessment Methods Phonetic Alphabet) in Romanian language there are 34 phonemes and for each a model is to be trained. For triphones the situation is more complicated because the number of them is large, around 40000, and the control of the training could be lost. To solve this problem, tying of acoustically similar states of the models built for triphones corresponding to each context is an efficient solution.

In the realized continuous speech recognition and understanding task we modelled intra-word triphones and also cross- words triphones. We adopted the state tying procedure, conducting to a controllable situation.

2.1.b HMM Types

The hidden Markov model incorporates the knowledge about feature constellation corresponding to each of the distinct phonetic units to be recognized. In our experiments we used continuous and semi-continuous models.

To describe HMMs, we start for simplicity reasons with the discrete model (Gold, Morgan, 2002).

Discrete HMMs

A discrete HMM is presented in Fig.3 in a Bakis form. The basic parameters of the model are:

- N –The number of states $S = \{s_1, s_2, ...s_N\}$; a state to a certain time is denominated as q_t, $(q_t \in S)$.
- M – The number of distinct symbols observable in each state. The observations are $V = \{v_1, v_2, ...v_M\}$; one element O_t from V is a symbol observed at moment t.
- A – The transition matrix containing the probabilities a_{ij} of the transition from state i in state j:

$$a_{ij} = A(i,j) = P\left(q_{t+1} = s_j \middle| q_t = s_i\right) \quad 1 \le i, j \le N \text{ , } t \in [1,T] \text{, } a_{ij} \ge 0 \text{ , } \sum a_{ij} = 1 \tag{4}$$

- B – Matrix of observed symbols in each state of the model: $b_j(k)$ represents the probability to observe a symbol v_k in state j:

$$b_j(k) = P\left(O_t = v_k \middle| q_t = s_j\right) \quad 1 \le j \le N \text{ , } 1 \le k \le M \text{ , } t \in [1,T] \text{, } b_j(k) \ge 0 \text{ , } \sum b_j(k) = 1 \tag{5}$$

- Π – The matrix of initial probabilities

$$\pi_i = P(q_1 = s_i) \text{, } \pi_i \ge 0 \text{ , } \sum \pi_i = 1 \tag{6}$$

In a compact mode a discrete HMM can be symbolized with $\lambda = (\Pi, A, B)$.

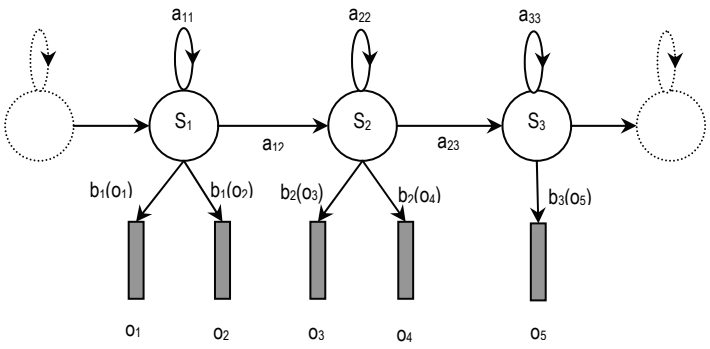


Fig 3. Bakis model with three states

Continuous densities hidden Markov models (CDHMM)
În the HMMs defined as $\lambda = (\Pi, A, B)$ the observations at a certain moment present a continuous probability density function, usually a Gaussian density or a mixture of Gaussian densities. For this last case we have:

$$b_i(O_t) = \sum_{m=1}^{M} c_{im} b_{im}(O_t), \; i = \overline{1, N} \tag{7}$$

c_{im} obey the restrictions: $c_{im} \geq 0$, $\sum_{m=1}^{M} c_{im} = 1$.

$b_{im}(O_t)$ is a K-dimensional Gaussian density with covariance matrix σ_{im} and mean μ_{im}:

$$b_{im}(O_t) = \frac{1}{\sqrt{(2\pi)^K |\sigma_{im}|}} \exp\left[-\frac{1}{2} (O_t - \mu_{im})^T \frac{1}{\sigma_{im}} (O_t - \mu_{im}) \right] \tag{8}$$

Semicontinuous hidden Markov models (SCHMM)
SCHMM are an intermediate form between the discrete and the continuous density hidden Markov models. An acoustical observation is described by a weighted combination of a number of probabilities, so that:

$$b_i(O_t) = \sum_{k=1}^{M} b_i(k) f_k(O_t) \text{ for } i = \overline{1, N} \tag{9}$$

$f_k(O_t)$ is a Gaussian density, with the covariance matrix Σ_k and mean vector μ_k.
Because speech is a signal with a high degree of variability, the most appropriate model capable to capture its complicated dependencies is the continuous one. But often also semicontinuous or discrete models are applied in simpler speech recognition tasks.

2.1.c Problems that can be solved with HMMs

Based on HMM's the statistical strategies has many advantages, among them being recalled: rich mathematical framework, powerful learning and decoding methods, good sequences handling capabilities, flexible topology for statistical phonology and syntax. The disadvantages lie in the poor discrimination between the models and in the unrealistic assumptions that must be made to construct the HMM's theory, namely the independence of the successive feature frames (input vectors) and the first order Markov process (Goronzy, 2002).
The algorithms developed in the statistical framework to use HMM are rich and powerful, situation that can explain well the fact that today, hidden Markov models are the widest used in practice to implement speech recognition and understanding systems.
The main problems that can be solved with HMMs are:

- The evaluation problem, in which given the model the probability to generate a observation sequence is calculated. This probability is the similarity measure used in recognition (decoding) to assign a speech segment to the model of highest probability. This problem can be solved with the forward or the backward algorithm
- The training problem, in which given a set of data, models for this data must be developed. It is a learning process, during which the parameters of the model are

estimated to fit the data. For each phonetic unit a model can be developed, such phonetic units can be monophones or intra- word or inter-word triphones. Utterances result through concatenation of these phonetic units. Training of the models is achieved with the Baum-Welch algorithm.

- The evaluation of the probability of the optimal observation sequence that can be generated by the model. This problem can be solved with the Viterbi algorithm. Often this algorithm is used instead the forward or backward procedure, because it is acting faster and decode easier the uttered sequence.

2.2 Recognition experiments based on monophones and triphones models

First we will define the conditions under that our experiments were conducted and further the obtained experimental results will be displayed.

To solve the problem of continuous speech recognition we used the specialized recognition tool based on hidden Markov models (HMMs), namely the HTK-Toolkit (Young et al., 2006).

2.2.1 Experiments conditions

To conduct this experiments, we choose the speech material contained in the first self-made in the university data base, called OCDRL, meaning Old Continuous Database for Romanian Language. The OCDRL is constituted by two databases: the training database contains 500 phrases, spoken by 10 speakers (8 males and 2 females), each speaker reading 50 phrases; the testing database contains 350 phrases spoken by the same speakers. The speech material was recorded in a laboratory environment, sampled with 16 kHz and quantized with 16 bits, the speakers were students, not professionals (Gavat et al., 2003), (Dumitru, 2006).

As speech data, the utterances of the data base were processed by phonetical transcription after the SAMPA standard (Sampa), conducting to the phonetic dictionary of the system. Each word is decomposed in constituent monophones (34 for Romanian language) or triphones (34^3 for Romanian language) and for each monophone or triphone a model must be trained. Of course for monophones the number of necessary models is small, there are sufficient training data, so that the models will be good trained in a short time. For triphones the situation is more complicated because there number is huge and the training data become insufficient (Young, 1994). Therefore tying procedure must be adopted, combining in a model similar triphones. Beam searching is the solution adopted in our experiments to realize the tying.

The digitizated data are further analysed in order to extract characteristic cues, called features. By short term analysis a set of features is obtained for each speech frame, extracted by a windowing process. The frame duration is chosen making a compromise between a long time (20-40 ms) imposed in order to detect the periodic parts of speech and the short time during which the speech can be considered a stationary random process (around 20 ms.). The experimental results further presented are obtained with a Hamming window, with duration 25 ms and the overlapping factor of the windows ½.

The features that can be obtained in our system to characterize speech segments are from two kinds: static features obtained for each window and dynamic features, calculated over a number of windows and representing derivatives of first, second and third degree. The static features type (Gavat et al., 2003), (Dumitru & Gavat, 2006) we have extracted to use in our experiments are:

- Perceptive linear prediction coefficients (PLP)
- Mel-frequency cepstral coefficients (MFCC)
- Linear prediction coefficients (LPC)
- Linear prediction reflexion coefficients (LPREFC)
- Linear prediction cepstral coefficients (LPCEPC)

All this features are 12-dimensional.
Energy and zero crossing rate were also calculated for each frame and are one-dimensional.
To each of this kind of features we can add the first, second and third degree derivatives, in order to capture the dynamic of speech.
To prepare features for training of the models, we perform normalization applying two algorithms:

- Cepstral mean normalization (CMN)
- Global variance normalization (GVN)

The sequences of features vectors obtained from the OCDRL training database are used to train the acoustical models, like monophones and triphones (intra-word and inter-word or cross word).
Further, we will evaluate the efficiency of the trained models by the word recognition rate (WRR), the accuracy and the phrase recognition rate (PRR) in a task of continuous speech recognition for Romanian language using for that the OCDRL test database.
At the end of the first tests we compared the word recognition rates, and could establish a first ranking of the best feature set for recognition. The results are displayed in Table 1.

FEM	PLP	MFCC	LPC	LPREFC	LPCEPC
WRR (%)	58.96	54.98	39.04	39.57	47.81
Accuracy (%)	56.97	52.59	36.25	36.17	45.82

Table 1. WRR and accuracy for the basic feature extraction methods (FEM).

It is to be seen that the best scores were obtained with the PLP coefficients (Hermansky, 1990), so that we will display bellow only results for feature vectors having as components PLP coefficients with first order derivatives (D) and second order ones (A) with or without energy (E).

2.2.2 Experimental results

We conducted recognition tests on the OCDRL test database, proving the generalization capability of the models in the following situations:

- Training of the models with global initialization (TGI)
- Retraining of the models with global initialization (RGI)
- Retraining of the models with individual initialization (RII)

Detailed is analyzed TGI. For RGI and RII some comparative results are given.
Training with global initialization (TGI)
We applied first the training procedure with global initialization, meaning that the training starts with all initial models having zero mean and unitary variance.
We have trained and than tested along our experiments the following types of continuous density models:

- Simple mixture monophones (SMM)
- Simple mixture intra- word triphones (SMIWT)

- Simple mixture cross- word triphones (SMCWT)
- Multiple mixtures monophones (MMM)
- Multiple mixtures intra-word triphones (MMIWT)
- Multiple mixtures cross-word triphones (MMCWT)

We have also trained and tested semicontinuous density models based on monophones and triphones.

The obtained results for continuous densities models are displayed in Table 2 for simple mixtures and in Table 3 for multiple mixtures. The results for semicontinuous densities are displayed in Table 4.

CDHMMs	PLP + E + D + A			PLP + D + A		
	WRR	Accuracy	PRR	WRR	Accuracy	PRR
SMM	84.47	83.16	40.00	84.74	84.74	44.00
SMM+CMN	87.37	87.37	42.00	87.89	87.89	52.00
SMIWT	97.37	97.37	84.00	98.16	98.16	88.00
SMIWT+CMN	97.63	97.63	88.00	96.32	96.32	80.00
SMCWT	91.84	91.58	52.00	91.32	90.79	50.00
SMCWT+CMN	89.21	88.42	38.00	90.79	90.53	48.00

Table 2. Recognition performance for singular mixture trained monophones and triphones in continuous density hidden Markov models (CDHMM).

CDHMMs/ number of mixtures		PLP + E + D + A			PLP + D + A		
		WRR	Accuracy	PRR	WRR	Accuracy	PRR
MMM	5	96.58	96.32	80.00	97.37	97.37	86.00
	10	97.37	97.37	86.00	97.37	97.37	88.00
	15	98.16	98.16	90.00	97.89	97.89	88.00
	20	98.16	98.16	90.00	98.42	98.42	90.00
MMIWT	2	98.68	98.68	92.00	98.42	98.42	90.00
	4	98.68	98.68	92.00	98.95	98.95	92.00
	6	98.68	98.68	92.00	99.21	99.21	94.00
	8	98.68	98.68	92.00	99.21	99.21	94.00
	10	98.95	98.95	94.00	98.95	98.95	92.00
	12	99.21	99.21	96.00	98.42	98.42	90.00
MMCWT	2	93.68	92.89	58.00	94.21	93.95	68.00
	4	93.42	92.63	56.00	95.26	95.00	70.00
	6	93.68	93.16	58.00	94.74	94.47	64.00
	8	95.00	94.21	62.00	94.74	94.47	62.00
	10	95.53	94.74	64.00	95.26	94.74	68.00
	12	94.74	93.95	62.00	94.74	94.21	62.00

Table 3. Recognition performance for multiple mixture trained monophones and triphones in continuous density hidden Markov models (CDHMM).

HMMs	PLP + E + D + A			PLP + D + A		
	WRR	Accuracy	PRR	WRR	Accuracy	PRR
monophones	96.58	95.26	76.00	97.11	97.11	82.00
monophones +CMN	96.51	96.24	79.59	97.11	98.58	84.00
triphones	97.89	97.63	88.00	98.42	98.42	88.00
triphones +CMN	98.42	97.89	88.00	98.68	98.68	92.00

Table 4. Recognition performance for semicontinuous hidden Markov models (SCHMMs).

Detailed discussions and comments concerning these results will be made in section 4.
Some global observations can be made:

- Triphones are in all cases more effective models than monophones
- Increasing the mixture number is helpful only below certain limits: for monophones this limit is around 20 mixtures, for inter-word triphones around 12 mixtures, for cross-word triphones around 10 mixtures
- Due to the poor applied grammar, WRR is always higher than PRR
- SCHMMs are slightly less more effective than CDHMMs
- CMN is slightly more effective for semicontinuous models, producing increases in the same time for WRR, accuracy and PRR
- In all cases, the best performance is obtained with the feature set (PLP + D + A)

For applications, not only the recognition rates, but also training and testing durations are important. Training of models is done off line, so that the training duration is not critical. The testing time is important to be maintained low, especially for real time applications.
Training and testing of the models were done on a standard PC with 1 GHZ Pentium IV processor and a dynamic memory of 1 GB. The obtained training and testing durations are detailed in Table 5 for different categories of models.

HMMs		Training duration (sec.)	Average testing duration/word (sec.)
CHMM	SMM	157	0.092
	30 MMM	2.323	0.291
	SMIWT	263	0.098
	12 MMIWT	1.087	0.219
	SMCWT	220	0.129
	12 MMCWT	1.106	0.223
SCHMM	Monophones	3.779	0.125
	Triphones	2.887	0.831

Table 5. Training and testing durations

As general observation we can say that the processing durations depend of the model complexity for both the training duration and testing duration. The training time takes values between 157s and 3.779s. The testing duration/word is less than 0.3s, so that real time applications are possible with this system.

Retraining with global initialization (RGI) and with individual initialization (RII)

The performance of the recognition system can be enhanced by retraining, with global initialization or individual initialization. In the first case, the prototypes of the retraining are to be globally initialized with the means and variances extracted from the data trained with

that monophone mixtures conducting to the best recognition results. In the second case, the initialization is to be done with individual means and variances, extracted from the trained data with a high mixtures number.
Bellow are displayed the comparative results of the recognition performance obtained by training with global initialisation (TGI), retraining with global initialization (RGI) and retraining with individual initialization (RII) for SMCWT in Table 6 and for MMCWT in Table 7.

Training type/ normalization		PLP + E + D + A			PLP + D + A		
		WRR	Accuracy	PRR	WRR	Acuracy	PRR
TGI	PS	97.37	97.37	84.00	98.16	98.16	88.00
	CNM	97.63	97.63	88.00	96.32	96.32	80.00
	RGV	96.84	96.84	82.00	95.79	95.79	76.00
RGI	PS	98.68	98.68	92.00	97.63	97.63	86.00
	CNM	98.68	98.68	92.00	98.68	98.68	92.00
	RGV	97.63	97.63	84.00	97.89	97.89	84.00
RII	PS	96.32	96.32	76.00	97.11	97.11	80.00
	CNM	98.68	98.68	92.00	98.16	98.16	88.00
	RGV	97.63	97.37	86.00	97.37	97.37	84.00

Table 6. Comparative results for TGI. RGI and RII for SMCWT

Training type/ number of mixtures		PLP + E + D + A			PLP + D + A		
		WRR	Accuracy	PRR	WRR	Acuracy	PRR
TGI	2	96.68	98.68	92.00	98.42	94.42	90.00
	4	98.68	98.68	92.00	98.95	98.95	92.00
	6	98.68	98.68	92.00	99.21	99.21	94.00
	8	98.68	98.68	92.00	99.37	97.37	88.00
	10	98.95	98.95	94.00	98.95	98.95	92.00
	12	99.21	99.21	96.00	98.42	98.42	90.00
RGI	2	98.68	98.42	90.00	98.68	98.68	92.00
	4	99.21	98.95	92.00	99.95	98.95	92.00
	6	98.68	98.68	92.00	99.21	99.21	94.00
	8	98.42	98.16	86.00	98.68	98.68	92.00
	10	98.42	98.16	86.00	98.68	98.68	9000
	12	99.21	99.21	96.00	98.42	98.42	90.00
RII	2	98.95	98.95	92.00	98.95	98.68	90.00
	4	99.21	98.95	94.00	98.68	98.68	90.00
	6	99.21	98.95	94.00	99.21	99.21	94.00
	8	99.47	98.95	94.00	99.21	99.21	94.00
	10	99.74	99.21	94.00	99.21	99.21	94.00
	12	99.47	98.95	92.00	98.95	98.95	92.00

Table 7. Comparative results for TGI. RGI and RII for MMCWT

The training durations are increasing by retraining. Some comparative results are presented in Table 8.

HMMs		TGI Training time (sec.)	TGI Average testing time/word (sec.)	RGI Training time (sec.)	RGI Average testing time/word (sec)	RII Training time (sec)	RII Average testing time/word (sec.)
CHMM	SMM	157	0.092	211	0.096	360	0.11

Table 8. Comparative results for training and testing durations for TGI, RGI and RII

As global remark, we can conclude that retraining procedures enhance the recognition performance, but also are raising the training durations.

2.4 Gender trained models

In speaker independent speech recognition for large vocabularies the training strategies for the acoustical models are very important: a well trained model has high generalization properties and leads to acceptable word and phrase recognition rates, even without special speaker adaptation procedures. This purpose can be realized in the simplest way by speaker selection in the training phase.

In our experiments made we have assessed the speech recognition performance configuring the training database in three manners: only with female speakers (FS), only with male speakers (MS), combining male and female speakers (MS and FS). In order to find out which training strategy ensures the highest generalization capacity, the tests were made with two kinds of databases: only with female speakers (FS), only with male speakers (MS).

For continuous speech recognition there are two databases namely CDRL (Continuous Database for Romanian Language) and SCDRL (Second Continuous Database for Romanian Language).

The characteristics (Dumitru, Gavat, 2007) for the first database CDRL are the following: the database is constituted for training by 3300 phrases, spoken by 11 speakers, 7 males and 4 female speakers, each speaker reading 300 phrases, and for testing by 880 phrases spoken by the same speakers, each of them reading 80 phrases. The training database contains over 3200 distinct words, while the testing database contains 1500 distinct words.

The second database, SCDRL, contain 2000 phrases, spoken by 5 males speakers and 5 females speakers; each of them reading 200 phrases for training and 100 phrases, 20 phrases spoken by 5 speakers (3 males speakers and 2 females speakers) for testing. The numbers of the distinct words are: 11000 words for training and 760 for testing.

The data are sampled for CDRL by 44.1 kHz and for SCDRL by 16 kHz, quantified with 16 bits, and recorded in a laboratory environment.

In order to carie out our experiments the database was reorganized as follows: one database for male speakers (MS), one database for female speakers (FS) and one database for male and female speakers (MS and FS). In the case of independent speaker we have excluded one MS and one FS from the training and we used for testing (Dumitru, 2006).

To assess the progresses made with our ASRS_RL system we initiated comparative tests for the performance expressed in word recognition rate (WRR) to establish the values under the

new conditions versus the starting ones. The comparison (CDRL *vs.* SCDRL) is made for the following situations (in Table 9):

- Gender based training/mixed training;
- MFCC_D_A (36 mel-frequency cepstral coefficients with the first and second order variation);
- HMM - monophone modeling.

Training	Testing	CDRL	SCDRL
Training MS	Testing MS	56.33	55.45
	Testing FS	40.98	50.72
Training FS	Testing MS	53.56	43.91
	Testing FS	56.67	64.18
Training MS & FS	Testing MS	57.44	53.53
	Testing FS	49.89	63.22

Table 9. Comparison between CDRL and SCDRL for the case of independent speaker.

Similar results are obtained in the case of dependent speaker, (tested speaker was used in the training too) for example in Table 10 is presented the results for SCDRL.

Training	Testing	SCDRL
Training MS	Testing MS	71.62
	Testing FS	53.58
Training FS	Testing MS	55.07
	Testing FS	73.30
Training MS & FS	Testing MS	68.58
	Testing FS	67.79

Table 10. WRR for SCDRL for the case of dependent speaker.

Newly, trying to improve the word recognition rate, we chose the triphone modeling and we extended the area of extracted parameters (features) from the speech signal to PLP. The results obtained for triphone using two parameterizations, MFCC_D_A with 36 coefficients and PLP with only 5 coefficients are displayed in Table 11 for CDRL database (Gavat, Dumitru, 2008).

Training	Testing	MFCC_D_A		PLP	
		Monophone	Triphone	Monophone	Triphone
Training MS	Testing MS	56.33	81.02%	34.02	68.10
	Testing FS	40.98	72.86	25.12	59.00
Training FS	Testing MS	53.56	69.23	23.78	53.02
	Testing FS	56.67	78.43	34.22	58.55
Training MS & FS	Testing MS	57.44	78.24	47.00	70.11
	Testing FS	49.89	74.95	41.22	69.65

Table 11. WRR (for CDRL) in the case of monophone *vs.* triphone for MFCC_D_A and PLP coefficients.

The obtained results show that gender training is effective only if testing is done for the same gender. Than the results are better as in the case of mixed training.

3. Language models

The language model is an important knowledge source, constraining the search for the word sequence that has produced the analyzed observations, in form of succession of feature vectors. In the language model are included syntactic and semantic constraints. If only syntactic constraints are expressed, the language model is reduced to a grammar. In the following we will present first some basic aspects concerning the language modelling and further experimental results obtained in ASRU experiments on a database of natural, spontaneous speech, constituted by broadcasted meteorological news.

3.1 Basics about language modeling

Language models intend to capture the interrelations between words in order to gain understanding of the phrases, their meaning. The language models best known are from two kinds: rule – based and statistical. Rule-based models conduct to a certain word sequence based on a set of rules. Statistical models determine the word sequence based on a statistical analysis of large amounts of data (Huang et al., 2001), (Juang et Furui, 2000).

3.1.a N-gram statistical models

From statistical point of view the language model is represented in relation (10) by the probability $P(W)$, that can be written in the form:

$$P(W) = P(w_1, w_2, ..., w_n) = P(w_1)P(w_2|w_1)\cdots P(w_n|w_1, w_2, ..., w_{n-1}) = \\ = \prod_{i=1}^{n} P(w_i|w_1, w_2, ..., w_{i-1}) \tag{10}$$

$P(w_i|w_1, w_2, ..., w_{i-1})$ is the probability that the word w_i follows after the word sequence w_1, w_2, $\ldots w_{i-1}$. The choice of w_i depends on the whole input history. For a vocabulary having as dimension v there are possible v^{i-1} different histories; it is a huge number, making practically impossible to estimate the probabilities even for not big i values.

To find a solution, shorter histories are considered and the most effective one is based of a history of two preceeding words, called the *trigram* model $P(w_i|w_{i-1}, w \quad i-2)$). In a similar way could be introduced the *unigram* ($P(w_i)$), or the *bigram* ($P(w_i|w_{i-1})$). Our language model is *bigram* based

3.2 Parsing technics

Parsing algorithms are applied to search the desired word sequence in an utterance, based on rules or based on statistics.

A parser based on rules is represented in Fig. 4, the statistical one is depicted in Fig.5.

Based on rules, parsing becomes dependent from linguists specialized knowledge in order to establish this rules. Based on learning to create from a training corpus the language model, the statistical parser is flexible and also more independent from specialized expertize. Efficiency of the statistical parsing can be enhanced by the so called boot-strapp training (Huang et al., 2001), consisting in developping a model for a part of the training corpus and refining this model successivelly in completing the whole trainig material.

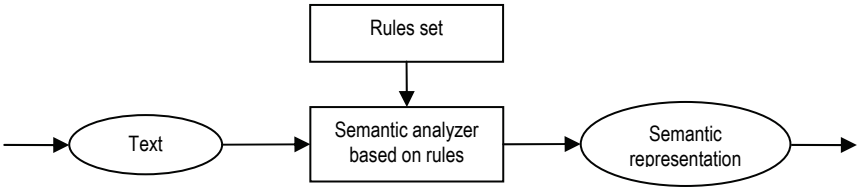


Fig. 4. Rule based parser

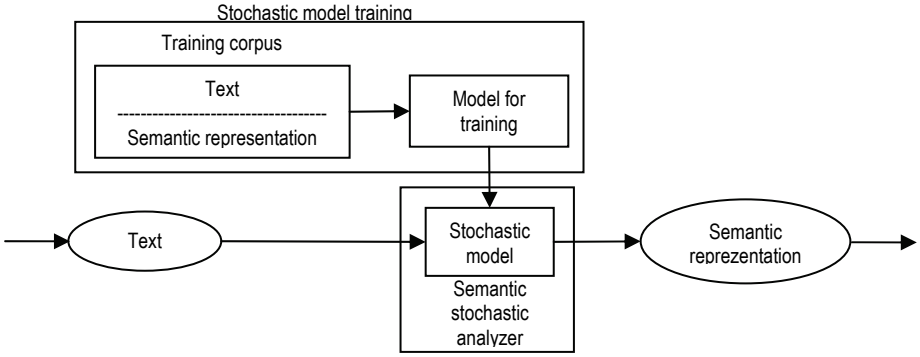


Fig. 5. Statistical parser

3.3 Experimental results

The language models applied in our research are:

- a simple word loop grammar, (WLG), permitting ever word sequence, without restrictions
- a restrictive finite state grammar (FSG), allowing only certain word sequences
- a bigram statistical model, extracting valid word sequences based on the bigram probabilities

The further displayed results were obtained on the MeteoRL database, constituted by registration of broadcasted meteorological news. The database contains 700 phrases, given a dictionary of 534 words. There are different speakers, male and females, not adnotated in our data.

The experiments had as objective to determine the influence of the language model on the ASRU performance, expressed in WRR, accuracy and PRR.

The conducted experiments were realized under the above listed conditions:

- Single mixtures for monophones, intra-word triphones and cross-word triphones (Table 12)
- Multiple mixtures for monophones, intra-word triphones and cross-word triphones (Table 13)
- Semi continuous models in form of monophones and triphones (Table 14)

Investigation of training duration and average testing duration/word were also done and the obtained results are displayed in Table 15. It is to notice the increase of the training but also of the average testing durations/word

HMMs/ number of mixtures		PLP + E + D + A			PLP + D + A		
		WRR	Accuracy	PRR	WRR	Acuracy	PRR
SMM	WLG	67.28	65.76	2.08	65.98	64.79	4.17
	FSG	95.89	95.11	55.32	96.33	95.67	53.19
	Bigram	97.40	96.97	72.92	98.70	98.37	81.25
SMIWT	WLG	87.87	81.15	12.50	89.38	85.70	14.58
	FSG	96.67	95.56	65.96	97.33	96.11	68.09
	Bigram	99.35	99.24	89.58	99.24	99.02	85.42
SMCWT	WLG	81.15	79.20	2.08	85.98	81.74	2.64
	FSG	96.11	95.56	59.57	97.67	96.56	61.70
	Bigram	99.35	99.24	87.50	99.13	98.92	85.42

Table 12. Comparative recognition performance for WLG, FSG and bigram model in the case of singular mixtures of monophones, intra-word triphones and cross-word triphones for continuous models.

HMMs/ number of mixtures		PLP + E + D + A			PLP + D + A		
		WRR	Accuracy	PRR	WRR	Acuracy	PRR
10 MMM	WLG	87.64	86.44	10.42	80.15	80.04	10.42
	FSG	97.22	96.44	65.96	96.89	96.00	59.57
	Bigram	98.70	98.59	83.33	98.37	98.16	83.33
6 MMIWT	WLG	87.87	81.15	12.50	89.11	87.11	17.02
	FSG	98.05	97.14	71.74	97.00	96.33	70.21
	Bigram	99.35	99.13	85.42	99.35	99.02	81.25
6 MMCWT	WLG	82.20	77.69	6.25	81.78	78.89	4.26
	FSG	97.71	97.37	69.57	96.67	96.22	68.09
	Bigram	99.44	99.44	89.36	99.00	99.00	87.23

Table 13. Comparative recognition performance for WLG, FSG and bigram model in the case of multiple mixtures of monophones, intra-word triphones and cross-word triphones for continuous models.

HMMs	PLP + E + D + A			PLP + D + A		
	WRR	Accuracy	PRR	WRR	Acuracy	PRR
Monophones						
WLG	83.42	80.82	6.25	78.01	77.03	4.17
FSG	97.00	96.22	61.70	96.89	96.22	59.57
Bigram	98.81	98.59	83.33	99.13	98.92	85.42
Triphones						
WLG	92.74	88.52	22.92	90.15	88.42	16.67
FSG	97.89	96.78	65.96	98.33	97.11	68.09
Bigram	98.81	98.37	77.08	98.22	98.00	76.60

Table 14. Comparative recognition performance for WLG, FSG and bigram model in the case of monophones and triphones of semicontinuous models.

HMMs		Training duration (sec.)	Average testing duration/word (sec.)		
			WLG	FSG	Bigram
CHMM	SMM	162	3.220	1.18	0.102
	10 MMM	672	3.930	1.43	0.130
	SMIWT	259	2.979	0.037	0.072
	6 MMIWT	361	1.250	0.069	0.097
	SMCWT	261	3.289	0.078	0.122
	6 MMCWT	652	7.390	0.143	0.205
SCHMM	Monophones	3.250	4.004	0.902	0.108
	Triphones	8.040	4.031	1.552	1.002

Table 15. Training duration and average testing duration/word for different language modeling and acoustical modeling techniques.

4. Discussion and comments of the experimental results

In sections 2 and 3 of this chapter we presented results obtained in continuous speech recognition and understanding experiments for Romanian language concerning the efficiency of:

- acoustical modeling based on monophones and triphones in continuous and semicontinuous models, with singular and multiple gaussian mixtures
- training with global initialization and retraining with global and individual initialization
- gender based training
- introduction of language models based on finite state grammars and bigram modeling

Some discussions and comments of this results could be usefull to conclude about the done work and future work directions.

4.1 Monophone and triphone models

All the experiments were carried out on the OCDRL database.

Comparing the results obtained for CDHMM models with singular mixtures (Table 2) it is obvious that triphone modeling enhance the recognition performance: WRR is increasing from 84.47% for monophones to 91.84% for CWT and to 97.37% for IWT, a maximum enhancement of more than 12%. Applying CMN the WRR marks again a slight increase.

The results obtained for CDHMM models with multiple mixtures (Table 3) show a WRR enhancement of around 12% for monophones with singular mixtures to monophones with five mixtures, with slight increase by increase of mixture numbers. For triphones, the WRR enhancement from single mixtures to multiple ones is not so spectacular: around 2% for IWT and 3% for CWT. Increasing the number of mixtures only slight increase in the WRR can be noticed. But because training time increases for multiple mixtures (Table 3) from 157s to 2323s for monophones, from 263s to 1087s for IWT and from 220s to 1106s for CWT it is better to not increase too much the mixtures number.

For SCHMM models, WRR increases of more than 1% can be remarqued by passing from monophones to triphones (Table 4).

4.2 Training and retraining

Training of the models was done with global initialization, retraining with global and individual initialization on the OCDRL database.
Of course an increase of the training and testing/word durations is to be noticed from 157s/0.092s for TGI to 211s/0.096s for RGI and 360s/0.11s for RII. (Table 8).
Increases, but not spectacular can be reported for WRR: for SMCWT for example, from 97.63% for TGI with CMN to 98.68% for RGI and RII with CMN (Table 6). Slightly higher increases can be reported for MMCWT for example: from 96.68% for TGI to 98.68% for RGI and 98.95% for RII for the case of two mixtures (Table 7).

4.3 Gender based training

The experiments were carried out on two databasis, CDRL and SCDRL for monophones and triphones, using as features MFCCs with first and second order variations and PLP coefficients, training with MS, FS and mixed and testing with MS and FS.
Gender based training and testing enhance the WRR. For example, training MS and testing MS leads to a WRR from 56.33% for the CDRL data base and 55.45% on the SCDRL data base for monophones, and 81.02% for triphones; testing with FS leads respectively to 40.98%, 50.72% and 72.86%, sensible lower values as for MS (Table9 and Table 11).
But it is to notice that in mixed training, the testing results are only slightly worser than for the gender based case: for training MS and testing MS WRR is 71.62%, training FS and testing FS WRR is 73.3%, but for mixed training WRR is 68.58% for testing MS and 67.79% testing FS (Table 11).

4.4 Language modelling

For the experiments a natural spontaneous spoken language database was used, namely MeteoRL. It is a way to explain why the results obtained on this database for WRR, accuracy and PRR are sensible lower than on the OCDRL database in which prompted, read text is used as speech material. For the SLG, in case of SMM for example, WRR, accuracy and PRR are respectively 84,47%, 83,16%, 40% for the OCDRL database (Table 2) and only 67,28%, 65,76%, 2,08% for the MeteoRL database (Table 12). Improving the language model, this data become respectively 95,89%, 95,11, 55,32% for FSG and 97,40%, 96,97%, 72,92% for the bigram model (Table 12), so that spectacular improving in ASRU performance is achieved, It is to notice that globally, the results obtained for the MeteoRL database follow the same trends as for the OCDRL data base. The known hierarchies are preserved: the WRR and PRR are higher for triphone models than for monophones, for multiple mixtures models than for single mixtures ones. For this experiments it is to highlite the improvement resulted by enhancing language modeling: starting for WLG with WRR, accuracy and PRR having the values of 82,20%, 77,69%, 6,25% for 6MMCWT, they became 97,71%, 97,37%, 69,57% for FSG and 99,44%, 99,44%, 89,36% for the bigram model (Table 13). Enhancements in ASRU performance can also be reported for semicontinuous models (Table 14)

5. Conclusions and future work

The done experiments helped us to obtain a deeper insight in the ASRU technics based on the statistical framework. The progress done in this work mainly consists in enhancing the language model applying for the first time in ASRU experiments for Romanian language

more elaborated language models as the simple WLG in form of the FSG and the bigram model. It is a work that in the future has to be further continued and improved.

Our major concern for future work is to obtain a standard database for Romanian language to validate the results obtained in ASRU experiments. The databases we have used were done in the laboratory of our university, carefully and with hard work, but still not fulfilling all standard requirements in audio quality and speech content.

6. References

Draganescu, M., (2003). Spoken language Technology, *Proceedings of Speech Technology and Human-Computer-Dialog (SPED2003)*, Bucharest, Romania, pp. 11-12.

Dumitru, C.O. and Gavat, I. (2006). Features Extraction and Training Strategies in Continuous Speech Recognition for Romanian Language, *International Conference on Informatics in Control, Automation & Robotics – ICINCO 2006*, Setubal, Portugal, pp. 114-121.

Dumitru, O. (2006). *Modele neurale si statistice pentru recunoasterea vorbirii*, Ph.D. thesis.

Dumitru, C.O. and Gavat, I. (2007). Vowel, Digit and Continuous Speech Recognition Based on Statistical, Neural and Hybrid Modelling by Using ASRS_RL, *Proceedings EUROCON 2007*, Warsaw, Poland, pp. 856-863.

Dumitru, C.O. and Gavat, I. (2008). NN and Hybrid Strategies for Speech Recognition in Romanian Language, *ICINCO 2008 – Workshop ANNIIP*, Funchal–Portugal, pp. 51-60

Gavat, I., Zirra, M. and Enescu, V. (1996-1). A Hybrid NN-HMM System for Connected Digit Recognition over Telephone in Romanian Language. *IVTTA '96 Proceedings*, Basking Ridge, N.J., pp. 37-40.

Gavat, I. and Zirra, M. (1996-2). Fuzzy models in Vowel Recognition for Romanian Language, *Fuzzy-IEEE '96 Proceedings*, New Orleans, pp. 1318-1326.

Gavat, I., Grigore, O., Zirra, M. and Cula, O. (1997). Fuzzy Variants of Hard Classification Rules, *NAFIPS'97 Proceedings*, New York, pp. 172-176.

Gavat, I., Zirra, M. and Cula, O. (1998). Hybrid Speech Recognition System with Discriminative Training Applied for Romanian Language, *MELECON '98 Proceedings*, Tel Aviv, Israel, pp. 11-15.

Gavat, I., & all. (2000). *Elemente de sinteza si recunoasterea vorbirii*, Ed. Printech, Bucharest.

Gavat, I., Valsan, Z., Sabac, B., Grigore, O. and Militaru, D. (2001-1). Fuzzy Similarity Measures - Alternative to Improve Discriminative Capabilities of HMM Speech Recognizers, *ICA 2001 Proceedings*, Rome, Italy, pp. 2316-2317.

Gavat, I., Valsan, Z. and Grigore, O. (2001-2). Fuzzy-Variants of Hidden Markov Models Applied in Speech Recognition, *SCI 2001 Proceedings, Invited Session: Computational Intelligence in Signal and Image Processing,* Orlando, Florida, pp. 126-130.

Gavat, I. and Dumitru, C.O. (2002-1). Continuous Speech Segmentation Algorithms Based on Artificial Neural Networks*, The 6th World Multiconference on Systemics, Cybernetics and Informations - SCI 2002*, Florida, SUA, Vol. XIV, pp. 111-114.

Gavat, I., Dumitru, C.O., Costache, G. (2002-2). Application of Neural Networks in Speech Processing for Romanian Language, *Sixth Seminar on Neural Network Applications in Electrical Engineering - Neurel 2002*, Belgrade, Yugoslavia, pp. 65-70.

Gavat, I., Dumitru, C.O., Costache, G., Militaru, D. (2003). Continuous Speech Recognition Based on Statistical Methods, *Proceedings of Speech Technology and Human-Computer-Dialog (SPED2003)*, Bucharest, pp. 115-126.

Gavat, I., Costache, G., Iancu, C., Dumitru, C.O. (2005-1). SVM-based Multimedia Classifier, *WSEAS Transactions on Information Science and Applications*, Issue 3, Vol. 2, pp. 305-310.

Gavat, I., Dumitru, C.O., Iancu, C., Costache, G. (2005-2). Learning Strategies in Speech Recognition, *The 47th International Symposium - ELMAR 2005*, Zadar, Croatia, pp. 237-240.

Gavat,I., Dumitru, C.O. (2008). The ASRS_RL - a Research Platform, for Spoken Language Recognition and Understanding Experiments, *Lecture Notes in Computer Science (LNCS)*, Vol. 5073, Part II, pp. 1142-1157.

Gold, B., Morgan, N. (2002). *Speech and audio signal processing*, John Wiley&Sons, N. Y.

Goronzy, S. (2002). *Robust Adaptation to Non-Native Accents in Automatic Speech Recognition*, Springer – Verlag, Berlin.

Hermansky, H. (1990). Perceptual Linear Predictive (PLP) Analysis of Speech, *Journal Acoustic Soc. America*, Vol. 87, No. 4, pp. 1738-1752.

Huang, X., Acero, A., Hon, H.W. (2001). *Spoken Language Processing–A Guide to Theory, Algorithm, and System Development*, Prentice Hall, 2001.

Juang, B.H., Furui, S. (2000). Automatic Recognition and Understanding of Spoken Language–A First Step Toward Natural Human–Machine Communication, *Proc. IEEE*, Vol. 88, No. 8, pp. 1142-1165.

Rabiner, L.R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition , *Proceedings of the IEEE*, Vol. 77, No. 8, pp. 257-286 .

Sampa. http://www.phon.ucl.ac.uk/home/sampa

Valsan, Z., Gavat, I., Sabac, B., Cula, O., Grigore, O., Militaru, D., Dumitru, C.O., (2002). Statistical and Hybrid Methods for Speech Recognition in Romanian, *International Journal of Speech Technology*, Kluwer Academic Publishers, Vol. 5, Number 3, pp. 259-268.

Young, S.J. (1992). The general use of tying in phoneme-based HMM speech recognizers, *Proceedings ICASSP'92*, Vol. 1, San Francisco, pp. 569-572.

Young, S.J., Odell, J.J., Woodland, P.C. (1994). Tree based state tying for high accuracy modelling, *ARPA Workshop on Human Language Technology*, Princeton.

Young, S., Kershaw, D., Woodland, P. (2006). *The HTK- Book*, U.K.

14

Construction of a Noise-Robust Body-Conducted Speech Recognition System

Shunsuke Ishimitsu
Hiroshima City University
Japan

1. Introduction

In recent years, speech recognition systems have been used in a wide variety of environments, including internal automobile systems. Speech recognition plays a major role in a dialogue-type marine engine operation support system (Matsushita & Nagao,2001) currently under investigation. In this system, speech recognition would come from the engine room, which contains the engine apparatus, electric generator, and other equipment. Control support would also be performed within the engine room, which means that operations with a 0-dB signal-to-noise ratio (SNR) or less are required. Noise has been determined to be a portion of speech in such low SNR environments, and speech recognition rates have been remarkably low. This has prevented the introduction of recognition systems, and up till now, almost no research has been performed on speech recognition systems that operate in low SNR environments. In this chapter, we investigate a recognition system that uses body-conducted speech, that is, types of speech that are conducted within a physical body, rather than speech signals themselves (Ishimitsu et al. 2001).

Since noise is not introduced into body-conducted signals that are conducted in solids, even within sites such as engine rooms which are low SNR environments, it is necessary to construct a system with a high speech recognition rate. However, when constructing such systems, learning data consisting of sentences that must be read a number of times is required for creation of a dictionary specialized for body-conducted speech. In the present study we applied a method in which the specific nature of body-conducted speech is reflected within an existing speech recognition system with a small number of vocalizations. Because two of the prerequisites for operating within a site such as an engine room where noise exists are both "hands-free" and "eyes-free" operations, we also investigated the effects of making such a system wireless.

2. Dialogue-type marine engine operation support system using body-conducted speech

Since the number of Japanese sailors has decreased dramatically in recent years, there is a shortage of skilled maritime engineers. Therefore, a database which stores the knowledge used by skilled engineers has been constructed (Matsushita & Nagao,2001).

In this study, this knowledge database is accessed by speech recognition. The system can be used to educate sailors and make it possible to check the ship's engines.

Figure 1 shows a conceptual diagram of a dialogue-type marine engine operation support system using body-conducted speech. The signals are detected with a body-conducted microphone and then wirelessly transmitted, and commands or questions from the speech-recognition system located in the engine control room are interpreted. A search is made for a response to these commands or questions speech recognition results and confirmation on the suitability of entering such commnads into the control system is made. Commands suitable for entry into the control system are speech-synthesized and output to a monitor. The speech-synthesized sounds are replayed in an ear protector/speaker unit, and while continuing communication, work can be performed while safety is continuously confirmed.
The present research is concerned with the development of the body-conducted speech recognition portion of this system. In this portion of the study, a system was created based on a recognition engine that is itself based on a Hidden Markov Model (HMM) incidental to a database (Itabashi, 1991). With this system, multivariate normal distribution is used as the output probability density function, and a mean vector μ that takes an n-dimensional vector as the frame unit of speech feature quantities and a covariance matrix Σ are used; these are expressed as follows: (Baum,1970)

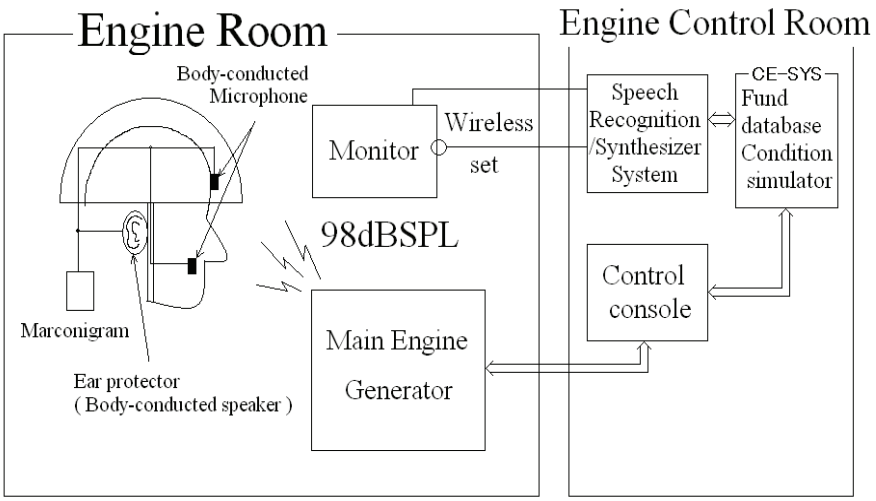


Fig. 1. Dialogue-type marine engine operation support system using body-conducted speech.

$$b(o,\mu,\Sigma)=\frac{1}{(2\pi)^{n/2}\mid\Sigma\mid^{1/2}}e^{-\frac{1}{2}(o-\mu)'\Sigma^{-1}(o-\mu)} \tag{1}$$

HMM parameters are shown using the two parameters of this output probability and the state transition probability. To update these parameters using conventional methods, utterances repeated at least 10-20 times would be required. To perform learning with only a few utterances, we focused on the relearning of the mean vector μ within the output probability, and thus created a user-friendly system for performing adaptive processing.

3. Investigation into identifying sampling locations for body-conducted speech

3.1 Investigation through frequency characteristics

Figure 2 shows candidate locations for body-conducted speech during this experiment. Three locations - the lower part of the pharynx, the upper left part of the upper lip and the front part of the zygomatic arch - were selected as signal sampling locations. The lower part of the pharynx is an effective location for extracting the fundamental frequency of a voice and is often selected by electroglottograph (EGG). Since the front part of the zygomatic arch is where a ship's chief engineer has his helmet strapped to his chin, it is a meaningful location for sound-transmitting equipment. The upper left part of the upper lip is the location that was chosen by Pioneer Co., Ltd. for application of a telecommunication system in a noisy environment. This location is confirmed to have very high voice clarity (Saito et al., 2001). Figure 3 indicates the amplitude characteristics of body-conducted speech signals at each location, and also shows the difference between a body-conducted signal on the upper lip and the voice when a 20-year-old male reads "Denshikyo Chimei 100" (this is the Japan Electronics and Information Technology Industries Association (JEITA) Data Base selection of 100 locality names). Tiny accelerometers were mounted on the above-mentioned locations with medical tape. Figure 3 indicates that the amplitudes of body-conducted speech at the zygomatic arch and the pharynx are 10-20 dB lower than body-conducted speech at the upper left part of the upper lip.

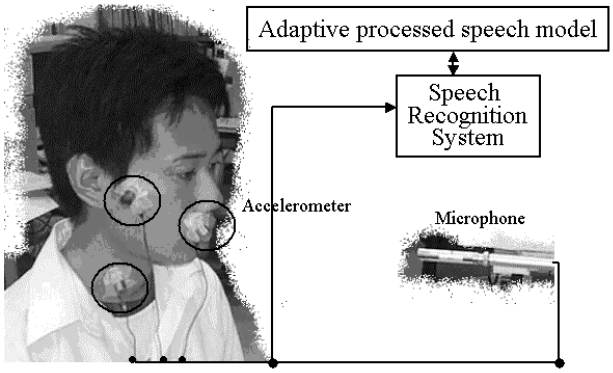


Fig. 2. Sampling location for body-conducted speech.

The clarity of vibration signals from body-conducted speech was poorer using signals from all sites except the upper left part of the upper lip in the listening experiment. Some consonant sounds that were not captured at other locations were extracted at the upper left part of the upper lip. However, compared to the speech signals shown in Figure 4, the amplitude characteristics at the upper left part of the upper lip appear to be about 10 dB lower than those of the voice.

Based on frequency characteristics, we believe that recognition of a body-conducted signal will be difficult utilizing an acoustic model built using acoustic speech signals. However, by using the upper left part of the upper lip, the site with the highest clarity signals, we think it will be possible to recognize body-conducted speech with an acoustic model built from acoustic speech using adaptive signal processing or speaker adaptation.

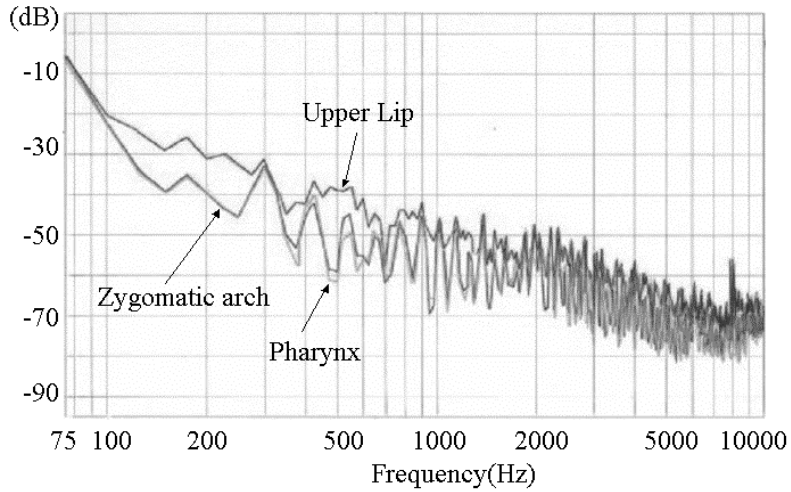


Fig. 3. Frequency characteristics of body-conducted speech.

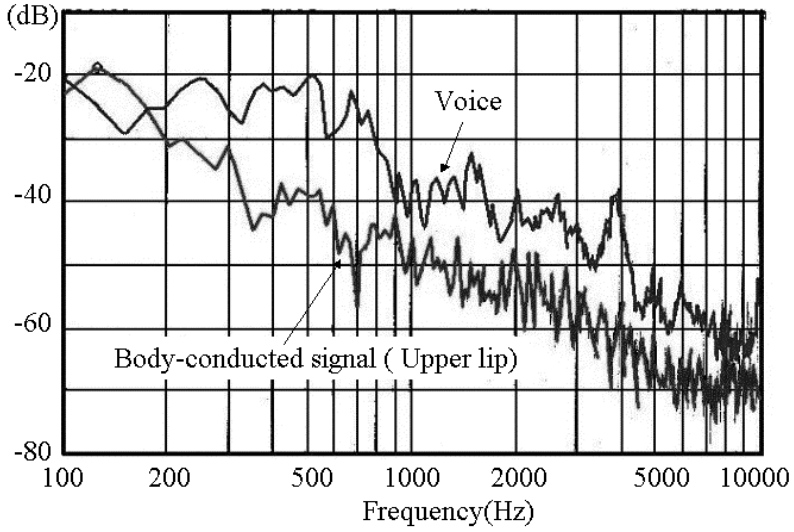


Fig. 4. Frequency characteristics of body-conducted speech and speech.

3.2 Comparison by recognition parameters

To investigate the effectiveness of a body-conducted signal model, we examined the characteristics of feature vectors. We are using LPC (Linear Predictive Coding) mel-cepstrum as the feature vectors to build an HMM. This system is widely used for parameters of speech recognition (Baum,1970). The first to the thirteenth coefficients were used as the feature vectors. The analysis conditions were: 12 kHz sampling, analysis frame length 22 msec, frame period 7 msec, analysis window hamming window.

In this study, we examined a word recognition system. To investigate the possibility of building a body-conducted speech recognition system with a speech model without building an entirely new body-conducted speech model, we compared sampling locations for body-conducted speech parameters at each location, and parameter differences amongst words. Figure 5 shows the difference on mel-cepstrum between speech and body-conducted speech at all frame averages. Body-conducted speech concentrates energy at low frequencies so that it converges on energy at lower orders like the lower part of the pharynx and the zygomatic arch, while the mel-cepstrum of signals from the upper left part of the upper lip shows a resemblance to the mel-cepstrum of speech. They have robust values at the seventh, ninth and eleventh orders and exhibit the outward form of the frequency property unevenly.

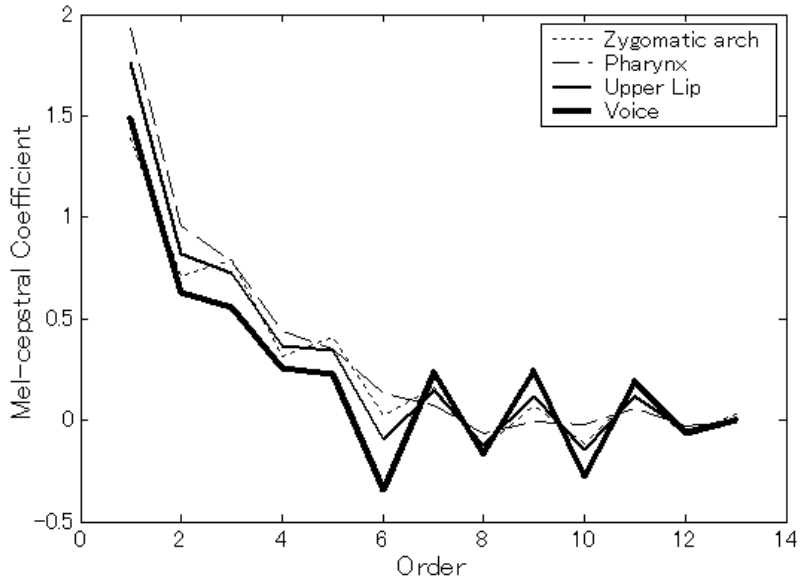


Fig. 5. Mel-cepstrum difference between speech and body-conducted speech.

Although the upper left part of the upper lip has the closest proximity to voice characteristics, it does not capture all of the characteristics of speech. This caused us to conclude that it is difficult to build a body-conducted speech model solely with a voice model.

We concluded that it might be possible to build a body-conducted speech recognition system by building a model at the upper left part of the upper lip and optimizing speech-conducted speech signals based on a voice model.

4. Recognition experiments

4.1 Selection of the optimal model

The experimental conditions are shown in Table 1. For system evaluation, we used speech extracted in the following four environments:

- Speech within an otherwise silent room
- Body-conducted speech within an otherwise silent room
- Speech within the engine room of the Oshima-maru while the ship was running
- Body-conducted speech within the engine room of the Oshima-maru while the ship was running

Noise within the engine room of the Oshima-maru when the ship was running was 98 dB SPL (Sound Pressure Level), and the SNR when a microphone was used was -25 dB. This data consisted of 100 terms read by a male aged 20, and the terms were read three times in each environment.

Valuation method	Three set utterance of 100 words
Vocabulary	JEITA 100 locality names
Microphone position	From the month to about 20cm
Accelerator position	The upper left part of the upper lip

Table 1. Experimental conditions

	anchorage		cruising	
	Speech	Body	Speech	Body
Anechoic room	45%	14%	2%	45%
Anechoic room + noise	64%	10%	0%	49%
Cabin	35%	9%	1%	42%
Cabin + noise	62%	4%	0%	48%

Table 2. The result of preliminary testing

Extractions from the upper left part of the upper lip were used for the body-conducted speech since the effectiveness of these signals was confirmed in previous research (Ishimitsu et al, 2001, Haramoto et al, 2001). the effectiveness of which has been confirmed in previous research. The initial dictionary model to be used for learning was a model for an unspecified speaker created by adding noise to speech extracted within an anechoic room. This model for an unspecified speaker was selected through preliminary testing. The result of preliminary testing is shown in Table 2.

4.2 Examination of the body-conducted speech recognition system using a voice model with pretreatment

We have shown that noise-robust speech recognition is possible using body-conducted speech which spreads through the inside of the body. However, the rate of speech recognition for body-conducted speech under the same calm conditions is slightly poorer than the rate of recognition for acoustic speech.

As a result, it was determined desirable to use a dictionary that had not been through an adaptation processing to the environment with a speaker. To that end, we examined how body-conducted speech quality could be improved to that of acoustic speech quality as the next step in our experiments. Specifically, the transfer function between speech and body-conducted speech was computed with adaptation signal processing and a cross-spectral method with the aim of raising the quality of body-conducted speech to that of speech by collapsing the body-conducted speech input during the transfer function. By using this filtering as a pretreatment, we hoped to improve the articulation score and recognition rate of body-conducted speech.

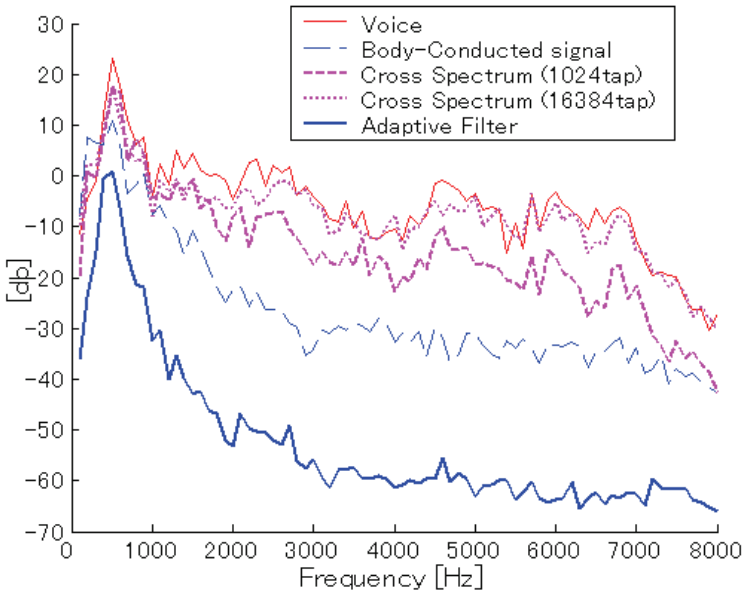


Fig. 6. The frequency characteristics of each method.

First, ten words were selected from the list of 100 place names, and then we analyzed the results using an adaptation filter and a cross-spectral method. The adaptation filter length was set to 1024, and the convergence coefficient was set to 0.01. In the cross-spectral method, the filter length was set to 1024 and 16384.

The frequency characteristics of a speech sound, a body-conducted speech sound, and the results generated by the use of an adaptation filter and a cross-spectral method are shown in Fig. 6. The characteristics of the pretreatment filter when each technique was used are shown in Fig. 7. This pretreatment filter was calculated with an adaptation algorithm using a reverse filter. The characteristics best approached the sound the speech when cross-spectral compensation was applied, and the transfer function took the form of a highpass filter. With an adaptation filter, a LMS algorithm was not able to lapse into a partial solution and the optimal solution could not be calculated this time.

Next, we describe the articulation score on reproduction; applying this pretreatment filter to body-conducted speech. With the adaptation filter, the processing result became a blurred sound. Although it seldom faded in the cross-spectral method, an echo occurred. When the

adaptation filter was applied to body-conducted speech, the results were closer when the filter length approached 16834 than when the filter length approached 1024. However, the echo also became stronger. For this reason (as a result of the speech recognition experiment by the free speech recognition software Julius) we were not able to check the predominance difference. In addition, adaptation to a speaker and environment were not taken into account in this application.

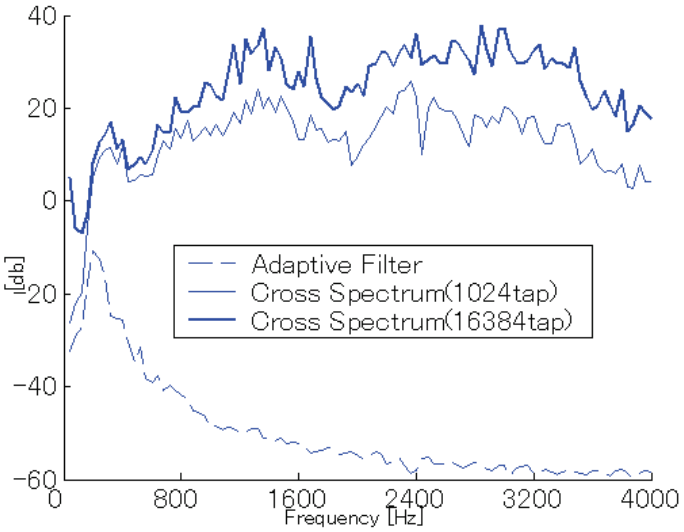


Fig. 7. The characteristics of the pretreatment filter.

Speech	Body	Filtered body
71%	57%	61%

Table 3. The result of preliminary testing with Julius

The highpass filter was designed based on the transfer function calculated with the cross-spectral method. The 16 tap FIR filter (Finite Impulse Response Filter) has a short filter length of a grade that does not generate the echo realized by the highpass filter. The recognition rate of filtered body-conducted speech is shown in Table 3. The recognition rate of body-conducted speech improved by 4% after filtering. Although the sound quality was clear and easy to hear when speech was filtered with the highpass filter, the noise in the high-frequency region was emphasized. For this reason, we concluded that the effect of the filter on the speech recognition rate was inadequate.

4.3 The effect of adaptation processing

The speech recognition test results in the cases where adaptive processing (Ishimitsu & Fujita, 1998) was performed for room interior speech and engine-room interior speech are shown in Table 4, and in Figures 8 and 9. The underlined portions show the results of the tests performed in each stated environment. In tests of recognition and signal adaptation via speech within the machine room, there was almost no operation whatsoever. That result is

shown in Figure 8, and it is thought that extraction of speech features failed because the engine room noise was louder than the speech sounds. Conversely, with room interior speech, signal adaptation was achieved. When environments for performing signal adaptation and recognition were equivalent, an improvement in the recognition rate of 27.66% was achieved, as shown in Figure 9. There was also a 12.99% improvement in the recognition rate for body-conducted speech within the room interior. However, since that recognition rate was around 20% it would be unable to withstand practical use. Nevertheless, based on these results, we found that using this method enabled recognition rates exceeding 90% with just one iteration of the learning samples.

The results of cases where adaptive processing was performed for room-interior body-conducted speech and engine-room interior body-conducted speech are shown in Table 5, and in Figures 10 and 11. Similar to the case where adaptive processing was performed using speech, when the environment where adaptive processing and the environment where recognition was performed were equivalent, high recognition rates of around 90% were obtained, as shown in Figure 10. In Figure 11. It can be observed that signal adaptation using engine-room interior body-conducted speech and speech recognition results were 95% and above, with 50% and above improvements, and that we had attained the level needed for practical usage.

Valuation	Candidate for adaptation		
	Room	Engine Room	No adaptation
Speech(Room)	**90.66**	1.33	63.00
Body(Room)	22.66	1.33	9.67
Speech(Engine)	1.00	**1.50**	0.67
Body(Engine)	46.50	1.50	45.00

Table 4. Result of adaptation processing with speech (%)

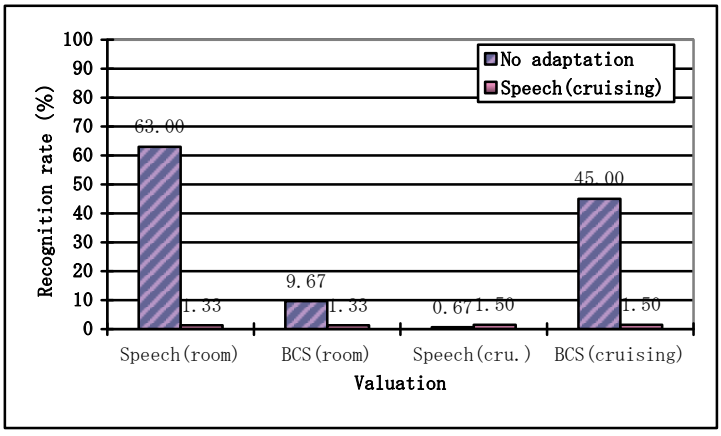


Fig. 8. Signal adaptation with speech (crusing).

Valuation	Candidate for adaptation		
	Room	Engine Room	No adaptation
Speech(Room)	40.67	46.17	63.00
Body(Room)	**86.83**	26.83	9.67
Speech(Engine)	1.50	1.00	0.67
Body(Engine)	49.00	**95.50**	45.00

Table 5. Result of adaptation processing with body-conducted speech (%)

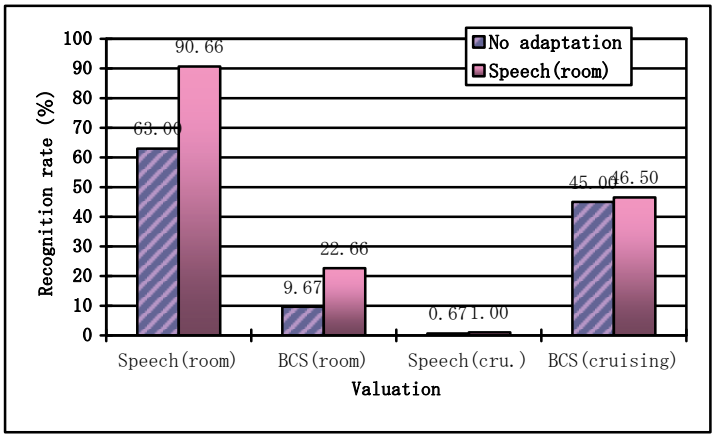


Fig. 9. Signal adaptation with speech (room).

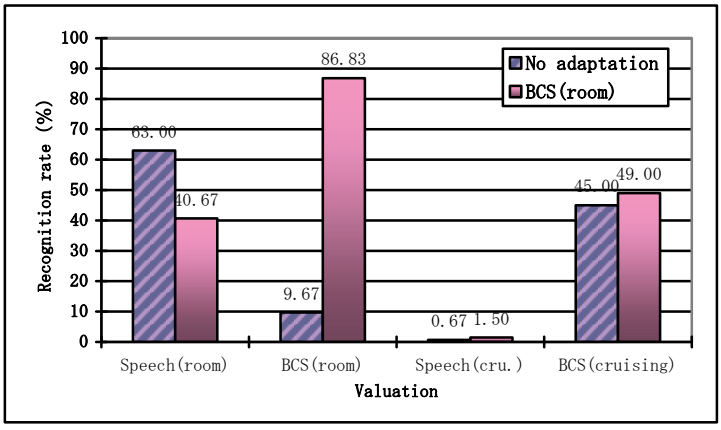


Fig. 10. Signal adaptation with body-conducted speech (room).

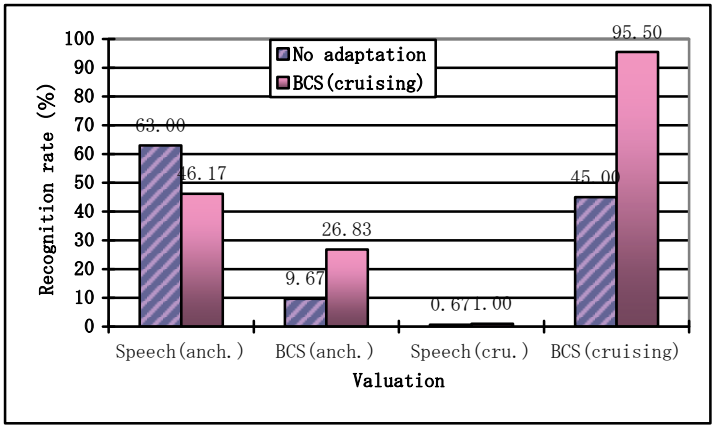


Fig. 11. Signal adaptation with body-conducted speech (crusing).

4.4 Investigation of the effects of making the system wireless

We next investigated the effects of making the system wireless. The specification of the wireless system used for this experiment is shown in Table 6. This data consisted of 100 terms read by a male aged 20 (a different man from the person who read the terms in the earlier experiment), and the terms were spoken three times in each one environment. The man who read the terms wore a helmet during analysis of body-conducted speech and the terms were extracted from the top of his head (calvaria). The effectiveness of this position has been confirmed in documentation (Saito et al. 2001). The initial dictionary model to be used for learning was, as in the previous tests, a model for an unspecified speaker. Here, the noise was white noise generated by a speaker, and was set in the vicinity of 0 dB SNR. The results for this experiment are shown in Table 7. From these results we concluded that if adaptive processing is performed when wired, the recognition rate becomes high, and thus the usefulness is confirmed. However, for speech transmitted wirelessly, the recognition rate was lower. This is thought to be becuase when the wireless type system was used, the noise was in the same frequency bandwidth as speech. The spectrogram analysis results of speech /hachinohe/ using cable and wireless are shown in Figures 12 and 13, respectively. From these figures it can be seen that although speech remained at less than 4000 Hz, noise overlap on the whole zone could be observed. This result suggests that it is necessary to test another method such as wireless LAN instead of a walkie-talkie.

Manufacturer	MOTOROLA
Part number	GL2000
Frequency	154.45-154.61MHz
Transmitting output	1W/5W

Table 6. Specifications for a wireless system

Conditions			No adaptation	adaptation
Cable	Quiet	speech	53.33	98.33
		body	43.66	97.00
wireless	Quiet	speech	3.33	77.00
		body	5.00	79.33
wireless	Noisy	speech	1.60	57.66
		body	2.00	62.00

Table 7. Results of a wireless vs. a cable system (%)

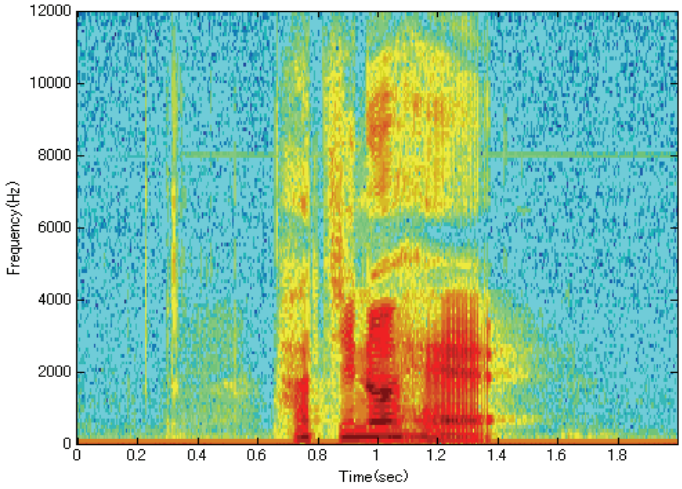


Fig. 12. /hachinohe/ with cable.

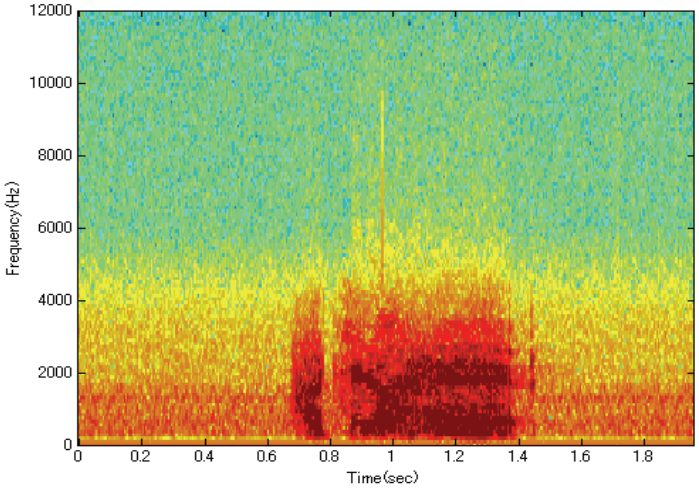


Fig. 13. /hachinohe/ with wireless.

7. Conclusion

We investigated a body-conducted speech recognition system for the establishment of a usable dialogue-type marine engine operation support system that is robust in noisy conditions, even in a low SNR environment such as an engine room. By bringing body-conducted speech close to audio quality, we were able to examine ways to raise the speech recognition rate. However, in an examination by pretreatment, we could not obtain optimal results when using an adaptation filter and a cross-spectral method. We introduced an adaptive processing method and confirmed the effectiveness of adaptive processing via small repetitions of utterances. In an environment of 98 dB SPL, improvements of 50% or above of recognition rates were successfully achieved within one utterance of the learning data and speech recognition rates of 95% or higher were attained. From these results, it was confirmed that this method will be effective for establishment of the present system.

In a wireless version of the system, the results showed a worsening of recognition rates because of noise in the speech bandwidth. Even when adaptive processing was performed, a sufficient speech recognition rate could not be obtained. Although more testing of this wireless system within the actual environment of the Oshima-maru will be necessary, it will also be necessary to investigate other wireless methods.

8. References

Matsushita, K. and Nagao, K. (2001). Support system using oral communication and simulator for marine engine operation. , *Journal of Japan Institude of Marine Engineering*, Vol.36, No.6, pp.34-42, Tokyo.

Ishimitsu, S., Kitakaze, H., Tsuchibushi, Y., Takata, Y., Ishikawa, T., Saito Y., Yanagawa H. and Fukushima M. (2001). Study for constructing a recognition system using the bone conduction speech, *Proceedings of Autumn Meeting Acoustic Society of Japan* pp.203-204, Oita, October, 2001, Tokyo.

Haramoto, T. and Ishimitsu, S. (2001). Study for bone-conducted spcceh recognition system under noisy environment, *Proceedings of 31st graduated Student Mechanical Society of Japan*, pp.152, Okayama, March, 200, Hiroshima.

Saito, Y., Yanagawa, H., Ishimitsu, S., Kamura K. and Fukushima M.(2001), Improvement of the speech sound quality of the vibration pick up microphone for speech recognition under noisy environment, *Proceedings of Autumn Meeting Acoustic Society of Japan* I, pp.691～692, Oita, October, 2001, Tokyo.

Itabashi S. (1991), *Continuous speech corpus for research*, Japan Information Processing Development Center, Tokyo.

Ishimitsu, S., Nakayama M. and Murakami, Y.(2001), Study of Body-Conducted Speech Recognition for Support of Maritime Engine Operation, *Journal of Japan Institude of Marine Engineering*, Vol.39, No.4, pp.35-40, Tokyo.

Baum, L.E., Petrie, T., Soules, G. and Weiss, N. (1970), A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *Annals of Mathematical Statistics*, Vol.41, No.1, pp.164-171, Oxford.

Ishimitsu, S. and Fujita, I.(1998), *Method of modifying feature parameter for speech recognition*, United States Patent 6,381,572, US.

Multi-modal ASR systems

15

Adaptive Decision Fusion for Audio-Visual Speech Recognition

Jong-Seok Lee[1] and Cheol Hoon Park[2]
[1] *Signal Processing Institute, Ecole Polytechnique Fédérale de Lausanne (EPFL)*
[2]*School of Electrical Engineering and Computer Science, KAIST*
[1]*Switzerland*
[2]*Korea*

1. Introduction

While automatic speech recognition technologies have been successfully applied to real-world applications, there still exist several problems which need to be solved for wider application of the technologies. One of such problems is noise-robustness of recognition performance; although a speech recognition system can produce high accuracy in quiet conditions, its performance tends to be significantly degraded under presence of background noise which is usually inevitable in most of the real-world applications.

Recently, audio-visual speech recognition (AVSR), in which visual speech information (i.e., lip movements) is used together with acoustic one for recognition, has received attention as a solution of this problem. Since the visual signal is not influenced by acoustic noise, it can be used as a powerful source for compensating for performance degradation of acoustic-only speech recognition in noisy conditions. Figure 1 shows the general procedure of AVSR: First, the acoustic and the visual signals are recorded by a microphone and a camera, respectively. Then, salient and compact features are extracted from each signal. Finally, the two modalities are integrated for recognition of the given speech.

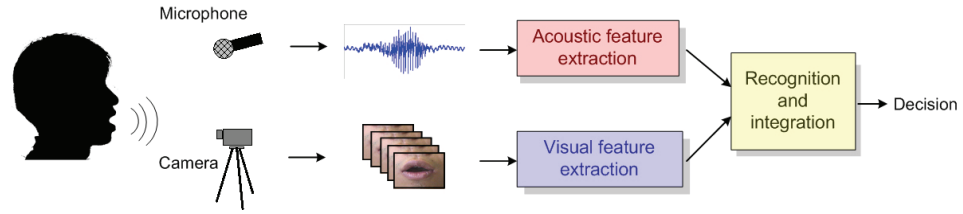


Fig. 1. General procedure of audio-visual speech recognition

In this chapter, we focus on the problem of audio-visual information fusion in AVSR, i.e., how to combine the two modalities effectively, which is an important issue for noise-robust AVSR. A good method of audio-visual fusion should exploit complementary characteristics of the two modalities efficiently so that we can obtain robust recognition performance over various noisy environments.

First, we give a review of methods for information fusion in AVSR. We present biological and psychological backgrounds of audio-visual information fusion. Then, we discuss existing fusion methods. In general, we can categorize such methods into two broad classes: feature fusion (or early integration) and decision fusion (or late integration). In feature fusion, the features from the two information sources are concatenated first and then the combined features are fed into a recognizer. In decision fusion, the features of each modality are used for recognition separately and the outputs of the two recognizers are integrated for the final recognition result. Each approach has its own advantages and disadvantages, which are explained and compared in detail in this chapter.
Second, we present an adaptive fusion method based on the decision fusion approach. Between the two fusion approaches explained above, it has been shown that decision fusion is more preferable for implementing noise-robust AVSR systems than feature fusion. In order to construct a noise-robust AVSR system adopting decision fusion, it is necessary to measure relative reliabilities of the two modalities for given speech data and to control the amounts of the contribution of the modalities according to the measured reliabilities. Such an adaptive weighting scheme enables us to obtain robust recognition performance consistently over diverse noise conditions. We compare various definitions of the reliability measure which have been suggested in previous researches. Then, we introduce a neural network-based method which is effective for generating appropriate weights for given audio-visual speech data of unknown noise conditions and thereby producing robust recognition results in a wide range of operating conditions.

2. Audio-visual speech recognition and information fusion

The ultimate goal of the AVSR technology would be to construct a recognizer whose performance is comparable to that of humans. Thus, understanding how humans perceive audio-visual speech will be helpful for constructing AVSR systems showing good performance. In this section, we review theories of modality integration in humans' bimodal speech perception in the viewpoint of biology and psychology, and presents approaches of information fusion for AVSR.

2.1 Bimodal nature of speech perception

The process of humans' speech production is intrinsically bimodal: The configuration of the tongue, the jaw, the teeth and the lips determines which specific sound is produced. Many of such articulatory movements are visible. Therefore, the mechanism of humans' speech perception is also bimodal. In a face-to-face conversation, we listen to what others say and, at the same time, observe their lip movements, facial expressions, and gestures. Especially, if we have a problem in listening due to environmental noise, the visual information plays an important role for speech understanding (Ross et al., 2007). Even in the clean condition speech recognition performance is improved when the talking face is visible (Arnold & Hill, 2001). Also, it is well-known that hearing-impaired people often have good lipreading skills.
There exist many researches proving the bimodality of speech perception and showing interesting results of audio-visual interaction due to the bimodality: The McGurk effect demonstrated the bimodality of the humans' speech perception by showing that, when the acoustic and the visual speech is incongruent, listeners recognize the given speech as a sound which is neither the acoustic nor the visual speech (McGurk & MacDonald, 1976). It was shown that many phonemes which are acoustically confusable are easily distinguished

using visual information (for example, /b/ and /g/) (Summerfield, 1987). Psychological experiments showed that seeing speakers' lips enhances the ability to detect speech in noise by decreasing auditory detection threshold of speech in comparison to the audio-only case, which is called "bimodal coherence masking protection" meaning that the visual signal acts as a cosignal assisting auditory target detection (Grant & Seitz, 2000; Kim & Davis, 2004). Such improvement is based on the correlations between the acoustic signal and the visible articulatory movement. Moreover, the enhanced sensitivity improves the ability to understand speech (Grant & Seitz, 2000; Schwartz et al., 2004).

A neurological analysis of the human brain shows an evidence of humans' multimodal information processing capability (Sharma et al., 1998): When different senses reach the brain, the sensory signals converge to the same area in the superior colliculus. A large portion of neurons leaving the superior colliculus are multisensory. In this context, a neurological model of sensor fusion has been proposed, in which sensory neurons coming from individual sensors are fused in the superior colliculus (Stein & Meredith, 1993). Also, it has been shown through positron emission tomography (PET) experiments that audio-visual speech perception yields increased activity in multisensory association areas such as superior temporal sulcus and inferior parietal lobule (Macaluso et al., 2004). Even silent lipreading activates the primary auditory cortex, which is shown by neuroimaging researches (Calvert et al., 1997; Pekkola et al., 2005; Ruytjens et al., 2007).

The nature of humans' perception demonstrates a statistical advantage of bimodality: When humans have estimates of an environmental property from two different sensory systems, any of which is possibly corrupted by noise, they combine the two signals in the statistically optimal way so that the variance of the estimates for the property is minimized after integration. More specifically, the integrated estimate is given by the maximum likelihood rule in which the two unimodal estimates are integrated by a weighted sum with each weight inversely proportional to the variance of the estimate by the corresponding modality (Ernest & Banks, 2002).

The advantage of utilizing the acoustic and the visual modalities for human speech understanding comes from the following two factors. First, there exists "complementarity" of the two modalities: The two pronunciations /b/ and /p/ are easily distinguishable with the acoustic signal, but not with the visual signal; on the other hand, the pronunciations /b/ and /g/ can be easily distinguished visually, but not acoustically (Summerfield, 1987). From the analysis of French vowel identification experiments, it has been shown that speech features such as height (e.g., /i/ vs. /o/) and front-back (e.g., /y/ vs. /u/) are transmitted robustly by the acoustic channel, whereas some other features such as rounding (e.g., /i/ vs. /y/) are transmitted well by the visual channel (Robert-Ribes et al., 1998). Second, the two modalities produce "synergy.": Performance of audio-visual speech perception can outperform those of acoustic-only and visual-only perception for diverse noise conditions (Benoît et al., 1994).

2.2 Theories of bimodal speech perception

While the bimodality of speech perception has been widely demonstrated as shown above, its mechanism has not been clearly understood yet because it would require wide and deep psychological and biological understanding about the mechanisms of sensory signal processing, high-level information processing, language perception, memory, etc. In this subsection, we introduce some existing psychological theories to explain how humans perform bimodal speech perception, some of which conflict with each other.

There exists a claim that visual speech is secondary to acoustic speech and affects perception only when the acoustic speech is not intelligible (Sekiyama & Tohkura, 1993). However, the McGurk effect is a counterexample of this claim; the effect is observed even when the acoustic speech is not corrupted by noise and clearly intelligible.

The direct identification model by Summerfield is an extension of Klatt's lexical-access-from-spectra model (Klatt, 1979) to a lexical-access-from-spectra-and-face-parameters model (Summerfield, 1987). The model assumes that the bimodal inputs are processed by a single classifier. A psychophysical model based on the direct identification has been derived from the signal detection theory for predicting the confusions of audio-visual consonants when the acoustic and the visual stimuli are presented separately (Braida, 1991).

The motor theory assumes that listeners recover the neuromotor commands to the articulators (referred to as "intended gestures") from the acoustic input (Liberman & Mattingly, 1985). The space of the intended gestures, which is neither acoustic nor visual, becomes a common space where the two signals are projected and integrated. A motivation of this theory is the belief that the objects of speech perception must be invariant with respect to phonemes or features, which can be achieved only by neuromotor commands. It was argued that the motor theory has a difficulty in explaining the influence of higher-order linguistic context (Massaro, 1999).

The direct realist theory also claims that the objects of speech perception are articulatory rather than acoustic events. However, in this theory the articulatory objects are actual, phonetically structured vocal tract movements or gestures rather than the neuromotor commands (Fowler, 1986).

The TRACE model is an interactive activation model in which excitatory and inhibitory interactions among simple processing units are involved in information processing (McClelland & Elman, 1986). There are three levels of units, namely, feature, phoneme and word, which compose of a bidirectional information processing channel: First, features activate phonemes, and phonemes activate words. And, activation of some units at a level inhibits other units of the same level. Second, activation of higher level units activates their lower level units; for example, a word containing the /a/ phoneme activates that phoneme. Visual features can be added to the TRACE of the acoustic modality, which produces a model in which separate feature evaluation of acoustic and visual information sources is performed (Campbell, 1988).

The fuzzy logical model of perception (FLMP) is one of the most appealing theories for humans' bimodal speech perception. It assumes perceiving speech is fundamentally a pattern recognition problem, where information processing is conducted with probabilities as in Bayesian analysis. In this model, the perception process consists of the three stages which are successive but overlapping, as illustrated in Figure 2 (Massaro, 1987; Massaro, 1998; Massaro, 1999): First, in the evaluation stage, each source of information is evaluated to produce continuous psychological values for all categorical alternatives (i.e., speech classes). Here, independent evaluation of each information source is a central assumption of the FLMP. The psychological values indicate the degrees of match between the sensory information and the prototype descriptions of features in memory, which are analogous to the fuzzy truth values in the fuzzy set theory. Second, the integration stage combines these to produce an overall degree of support for each alternative, which includes multiplication of the supports of the modalities. Third, the decision stage maps the outputs of integration into some response alternative which can be either a discrete decision or a likelihood of a given response.

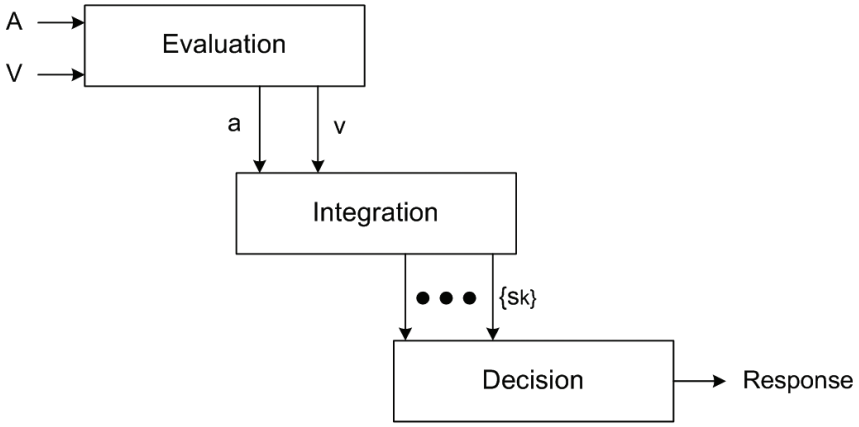


Fig. 2. Illustration of the processes of perception in FLMP. A and V represent acoustic and visual information, respectively. a and v are psychological values produced by evaluation of A and V, respectively. s_k is the overall degree of support for the speech alternative k.

It is worth mentioning about the validity of the assumption that there is no interaction between the modalities. Some researchers have argued that interaction between the acoustic and the visual modalities occurs, but it has also argued that very little interaction occurs in human brains (Massaro & Stork, 1998). In addition, the model seems to successfully explain several perceptual phenomena and be broadening its domain, for example, individual differences in speech perception, cross-linguistic differences, distinction between information and information-processing. Also, it has been shown that the FLMP gives better description of various psychological experiment results than other integration models (Massaro, 1999).

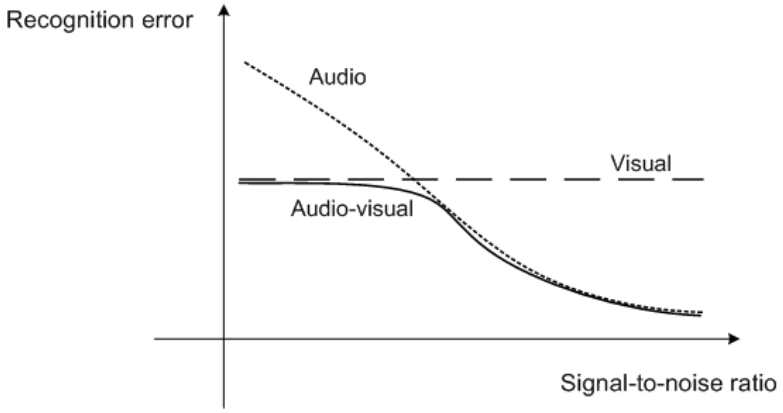
2.3 Approaches for information fusion in AVSR

The primary challenge in AVSR is to obtain the performance which is equal to or better than the performance of any modality for various noise conditions. When the noise level is low, the acoustic modality performs better than the visual one and, thus, the audio-visual recognition performance should be at least as good as that of the acoustic speech recognition. When the noise level is high and the visual recognition performance is better than the acoustic one, the integrated recognition performance should be at least the same to or better than the performance of the visual-only recognition.

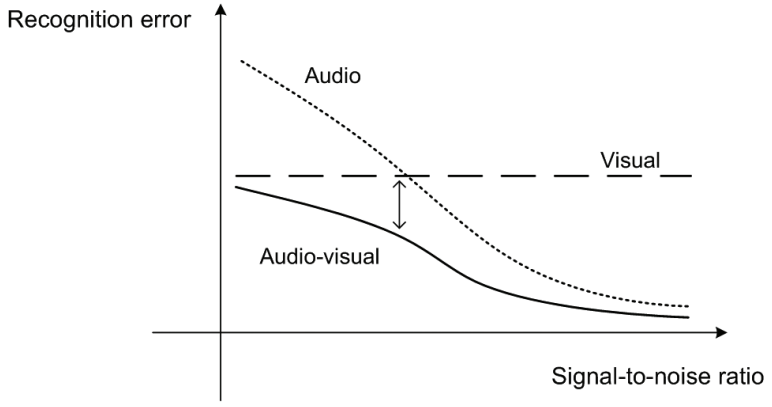
Besides, we expect the synergy effect of the two modalities by using AVSR systems. Thus, the goal of the second challenge in the use of audio-visual information for speech recognition is to improve the recognition performance with as a high synergy of the modalities as possible.

These two challenges are illustrated in Figure 3. The audio-visual information fusion process is an important issue causing the gap of the audio-visual recognition performance of the two cases in the figure. Combining the two modalities should take full advantage of the modalities so that the integrated system shows a high synergy effect for a wide range of noise conditions. On the contrary, when the fusion is not performed appropriately, we cannot expect complementarity and synergy of the two information sources and, moreover,

the integrated recognition performance may be even inferior to that of any of the unimodal systems, which is called "attenuating fusion" or "catastrophic fusion" (Chibelushi et al., 2002).



(a)



(b)

Fig. 3. Two challenges of AVSR. (a) The integrated performance is at least that of the modality showing better performance for each noise level. (b) The integrated recognition system shows the synergy effect.

In general, we can categorize methods of audio-visual information fusion into two broad categories: feature fusion (or early integration) and decision fusion (or late integration), which are shown in Figure 4. In the former approach, the features of the two modalities are concatenated to form a composite feature vector, which is inputted to the classifier for recognition. In the latter approach, the features of each modality are used for recognition separately and, then, the outputs of the two classifiers are combined for the final recognition result. Note that the decision fusion approach shares a similarity with the FLMP explained in the previous subsection in that both are based on the assumption of class-conditional independence, i.e., the two information sources are evaluated (or recognized) independently.

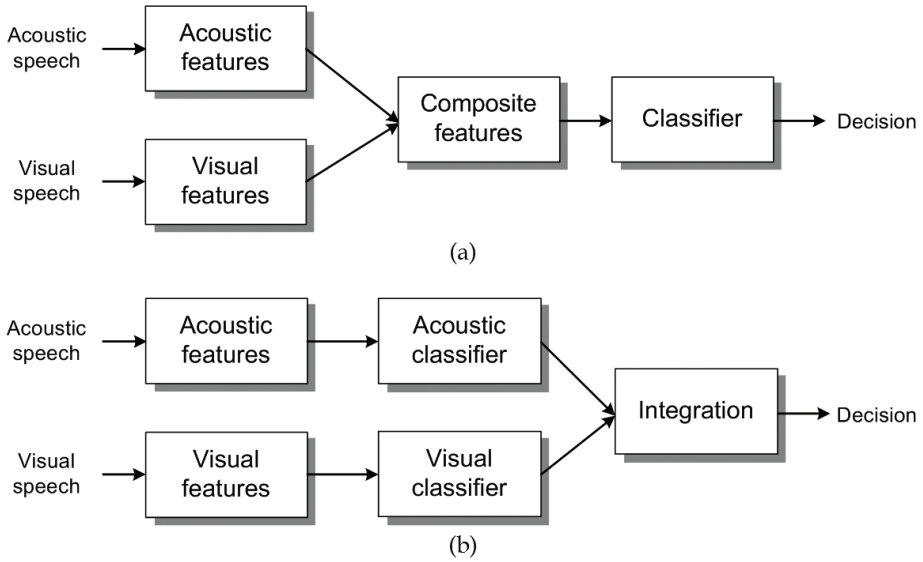


Fig. 4. Models for integrating acoustic and visual information. (a) Feature fusion. (b) Decision fusion.

Although which approach is more preferable is still arguable, there are some advantages of the decision fusion approach in implementing a noise-robust AVSR system. First, in the decision fusion approach it is relatively easy to employ an adaptive weighting scheme for controlling the amounts of the contributions of the two modalities to the final recognition according to the noise level of the speech, which is because the acoustic and the visual signals are processed independently. Such an adaptive scheme facilitates achieving the main goal of AVSR, i.e., noise-robustness of recognition over various noise conditions, by utilizing the complementary nature of the modalities effectively. Second, the decision fusion allows flexible modelling of the temporal coherence of the two information streams, whereas the feature fusion assumes a perfect synchrony between the acoustic and the visual feature sequences. It is known that there exists an asynchronous characteristic between the acoustic and the visual speech: The lips and the tongue sometimes start to move up to several hundred milliseconds before the acoustic speech signal (Benoît, 2000). In addition, there exists an "intersensory synchrony window" during which the human audio-visual speech perception performance is not degraded for desynchronized audio-visual speech (Conrey & Pisoni, 2006). Third, while it is required to train a whole new recognizer for constructing a feature fusion-based AVSR system, a decision fusion-based one can be organized by using existing unimodal systems. Fourth, in the feature fusion approach the combination of the acoustic and the visual features, which is a higher dimensional feature vector, is processed by a recognizer and, thus, the number of free parameters of the recognizer becomes large. Therefore, we need more training data to train the recognizer sufficiently in the feature fusion approach. To alleviate this, dimensionality reduction methods such as principal component analysis or linear discriminant analysis can be additionally used after the feature concatenation.

3. Decision fusion with adaptive weighting scheme

The dominant paradigm for acoustic and visual speech recognition is the hidden Markov model (HMM) (Rabiner, 1989). We train an HMM to construct a model for the acoustic or visual utterance of a speech class. And, the set of HMMs for all speech classes form a speech classifier.

As discussed in the previous section, the decision fusion approach is a good choice for designing a noise-robust AVSR system. Decision fusion in HMM-based AVSR systems is performed by utilizing the outputs of the acoustic and the visual HMMs for a given audio-visual speech datum. The important issue is how to implement adaptive decision fusion to obtain noise-robustness over various noise environments. To solve this, it is necessary to define the relative reliability measure of a modality (which is affected by the noise level) and determine an appropriate weight based on the measured reliabilities.

In this section, we present the principle of adaptive weighting, various definitions of the reliability measure, and a neural network-based method for obtaining proper integration weights according to the reliabilities.

3.1 Adaptive weighting

Adaptive weighting in decision fusion is performed in the following way: When the acoustic and the visual features (O_A and O_V) of a given audio-visual speech datum of unknown class are obtained, the recognized utterance class C^* is given by (Rogozan & Deléglise, 1998)

$$C^* = \arg\max_i \left\{ \gamma \log P(O_A \mid \lambda_A^i) + (1-\gamma) \log P(O_V \mid \lambda_V^i) \right\}, \tag{1}$$

where λ_A^i and λ_V^i are the acoustic and the visual HMMs for the *i*-th class, respectively, and $\log P(O_A \mid \lambda_A^i)$ and $\log P(O_V \mid \lambda_V^i)$ are their outputs (log-likelihoods). The integration weight γ determines how much the final decision relatively depends on each modality. It has a value between 0 and 1, and varies according to the amounts of noise contained in the acoustic speech. When the acoustic speech is clean, the weight should be large because recognition with the clean acoustic speech usually outperforms that with the visual speech; on the other hand, when the acoustic speech contains much noise, the weight should be sufficiently small. Therefore, for noise-robust recognition performance over various noise conditions, it is important to automatically determine an appropriate value of the weight according to the noise condition of the given speech signal.

3.2 Reliability measures

The reliability of each modality can be measured from the outputs of the corresponding HMMs. When the acoustic speech does not contain any noise, there are large differences between the acoustic HMMs' outputs. The differences become small when the acoustic speech contains noise, which reflects increased ambiguity in recognition due to the noise. This phenomenon is illustrated in Figure 5 which shows the outputs (log-likelihoods) of the HMMs for all utterance classes when a speech datum of clean or noisy condition is presented. (An utterance of the fourth class in the DIGIT database described in Section 4.1 is used for obtaining the result in the figure. For the acoustic features and the recognizer, refer to Sections 4.2 and 4.4, respectively.)

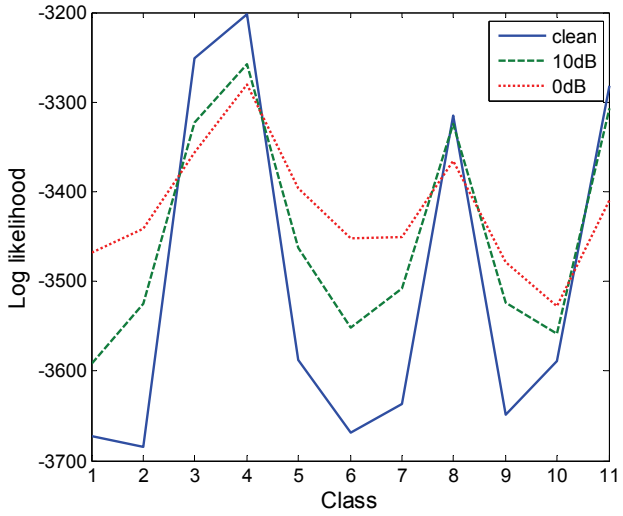


Fig. 5. Outputs of HMMs for different noise levels.

Considering this observation, we can define the reliability of a modality in various ways:

- Average absolute difference of log-likelihoods (**AbsDiff**) (Adjoudani & Benoît, 1996):

$$S = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} | L^i - L^j | , \tag{2}$$

  where $L^i = \log P(O | \lambda^i)$ is the output of the HMM for the *i*-th class and N the number of classes being considered.

- Variance of log-likelihoods (**Var**) (Lewis & Powers, 2004):

$$S = \frac{1}{N-1} \sum_{i=1}^{N} (L^i - \bar{L})^2 , \tag{3}$$

  where $\bar{L} = \frac{1}{N} \sum_{i=1}^{N} L^i$ is the average of the outputs of the N HMMs.

- Average difference of log-likelihoods from the maximum (**DiffMax**) (Potamianos & Neti, 2000):

$$S = \frac{1}{N-1} \sum_{i=1}^{N} \left| \max_j L^j - L^i \right| , \tag{4}$$

  which means the average difference between the maximum log-likelihood and the other ones.

- Inverse entropy of posterior probabilities (**InvEnt**) (Matthews et al., 1996):

$$S = \left[-\frac{1}{N} \sum_{i=1}^{N} P(C_i | O) \log P(C_i | O) \right]^{-1} , \tag{5}$$

where $P(C_i \mid O)$ is the posterior probability which is calculated by

$$P(C_i \mid O) = \frac{P(O \mid \lambda^i)}{\sum_{j=1}^{N} P(O \mid \lambda^j)}. \tag{6}$$

As the signal-to-noise ratio (SNR) value decreases, the differences of the posterior probabilities become small and the entropy increases. Thus, the inverse of the entropy is used as a measure of the reliability.

Performance of the above measures in AVSR will be compared in Section 4.

3.3 Neural network-based fusion

A neural network models the input-output mapping between the two reliabilities and the integrating weight so that it estimates the optimal integrating weights as shown in Figure 6 (Lee & Park, 2008), i.e.,

$$\hat{\gamma} = f(S_A, S_V), \tag{7}$$

where f is the function modelled by the neural network and $\hat{\gamma}$ the estimated integrating weight for the given acoustic and visual reliabilities (S_A and S_V, respectively). The universal approximation theorem of neural networks states that a feedforward neural network can model any arbitrary function with a desired error bound if the number of its hidden neurons is not limited (Hornik et al., 1989).

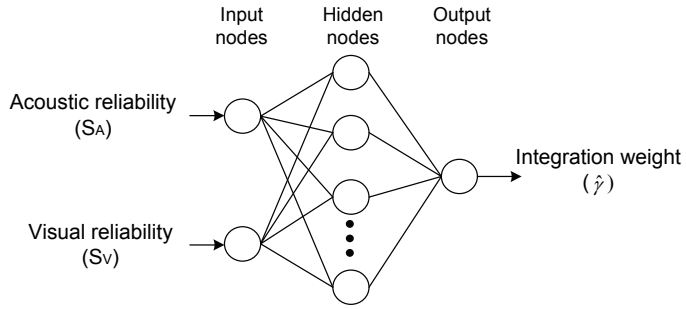


Fig. 6. Neural network for estimating integration weights.

The neural network should be trained before it is used as an estimator of the integrating weight. To ensure that we obtain appropriate weights for various noise conditions by using the neural network, both clean and noisy speech data are used for training. Since it is practically impossible to use the data of all possible noise conditions, we use only speech data for a few sampled conditions. Specifically, the clean, 20 dB, 10 dB and 0 dB noisy speech data corrupted by white noise are used for training. Then, the neural network produces appropriate weights for the noise conditions which are not considered during training by its generalization capability.

Training is conducted as follows: First, we calculate the reliability of each modality for each training datum by using one of the reliability measures described in Section 3.2. Then, we obtain the integrating weights for correct recognition of the datum exhaustively; while increasing the weight from 0 to 1 by 0.01, we test whether the recognition result using the

weight value is correct. Finally, the neural network is trained by using the reliabilities of the two modalities and the found weights as the training input and target pairs.
The integrating weight for correct recognition appears as an interval instead of a specific value. Figure 7 shows an example of this. It is observed that for a large SNR a large interval of the weight produces correct recognition and, as the SNR becomes small, the interval becomes small.

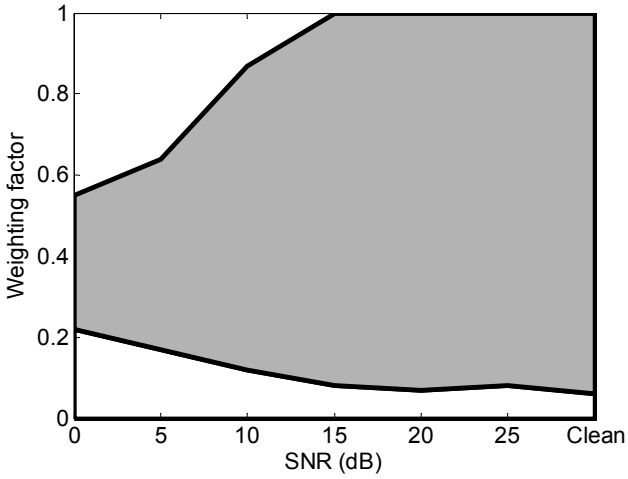


Fig. 7. Intervals of the integration weight producing correct recognition.

Therefore, the desired target for a training input vector of the neural network is given by an interval. To deal with this in training, the original error function used in the training algorithm of the neural network,

$$e(y) = t - y \,, \tag{8}$$

where t and y are the target and the output of the network, respectively, is modified as

$$e(y) = \begin{cases} \gamma_l - y & \text{for } y < \gamma_l \\ 0 & \text{for } \gamma_l \le y \le \gamma_u \\ \gamma_u - y & \text{for } \gamma_u < y \end{cases}, \tag{9}$$

where γ_l and γ_u are the lower and the upper bounds of the interval of the target weight value, respectively, which correspond to the boundaries of the shaded region in Figure 7.

4. Experiments

4.1 Databases

We use the two isolated word databases for experiments: the DIGIT database and the CITY database (Lee & Park, 2006). The DIGIT database contains eleven digits in Korean (including two versions of zero) and the CITY database sixteen famous Korean city names. Fifty six speakers pronounced each word three times for both databases. While a speaker was pronouncing a word, a video camera and a microphone simultaneously recorded the face

region around the speaker's mouth and the acoustic speech signal, respectively. The acoustic speech was recorded at the rate of 32 kHz and downsampled to 16 kHz for feature extraction. The speaker's lip movements were recorded as a moving picture of size 720x480 pixels at the rate of 30 Hz.
The recognition experiments were conducted in a speaker-independent manner. To increase reliability of the experiments, we use the jackknife method; the data of 56 speakers are divided into four groups and we repeat the experiment with the data of the three groups (42 speakers) for training and those of the remaining group (14 speakers) for test.
For simulating various noisy conditions, we use four noise sources of the NOISEX-92 database (Varga & Steeneken, 1993): the white noise (WHT), the F-16 cockpit noise (F16), the factory noise (FAC), and the operation room noise (OPS). We add each noise to the clean acoustic speech to obtain noisy speech of various SNRs.

4.2 Acoustic feature extraction

The popular Mel-frequency cepstral coefficients (MFCCs) are extracted from the acoustic speech signal (Davis & Mermelstein, 1980). The frequency analysis of the signal is performed for each frame segmented by the Hamming window having the length of 25 ms and moving by 10 ms at a time. For each frame we perform the Fourier analysis, computation of the logarithm of the Mel-scale filterbank energy, and the discrete cosine transformation. The cepstral mean subtraction (CMS) method is applied to remove channel distortions existing in the speech data (Huang et al., 2001). As a result, we obtain 12-dimensional MFCCs, the normalized frame energy, and their temporal derivatives (i.e., delta terms).

4.3 Visual feature extraction

The visual features must contain crucial information which can discriminate between the utterance classes and, at the same time, is common across speakers having different colors of skins and lips and invariant to environmental changes such as illuminations.
In general, there are two broad categories of visual speech feature extraction: the contour-based method and the pixel-based method. The contour-based approach concentrates on identifying the lip contours. After the lip contours are tracked in the image sequences, certain measures such as the height or width of the mouth opening are used as features (Kaynak et al., 2004), or a model of the contours is built and a set of parameters describing the model configuration is used as a feature vector (Dupont & Luettin, 2000; Gurbuz et al., 2001). In the pixel-based approach, the image containing the mouth is either used directly or after some image transformations (Bregler & Konig, 1994; Lucey, 2003). Image transformation methods such as principal component analysis (PCA), discrete cosine transform and discrete wavelet transform are frequently used.
We carefully design the method of extracting the lip area and define an effective representation of the visual features derived from the extracted images of the mouth region. Our method is based on the pixel-based approach because it has advantages over the contour-based one: It does not need a complicated algorithm for accurate tracking of the lip contours and does not lose important information describing the characteristics of the oral cavity and the protrusion of lips (Matthews et al, 2001). Figure 8 summarizes the overall procedure of extracting visual features.

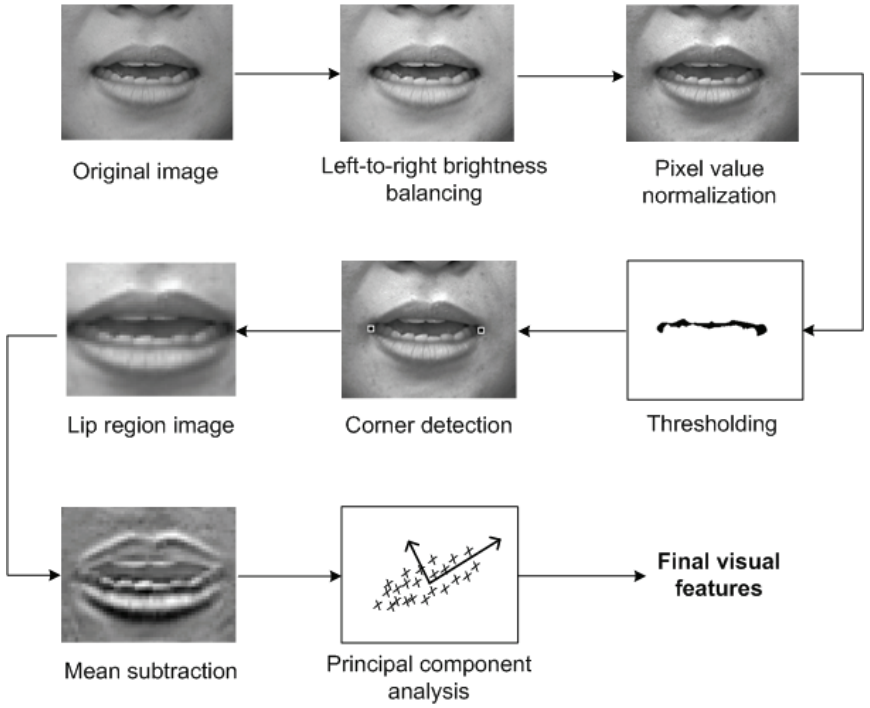


Fig. 8. Procedure of visual feature extraction.

1. We remove the brightness variation across the left and the right parts of an image so that the two mouth corners are accurately detected. We model the gradual horizontal brightness variation as the linear interpolation of the average pixel values of the left and the right small regions in the image. Then, this brightness variation is subtracted from the image in the logarithmic domain.
2. Normalization of the pixel values in the image is performed so that the pixel values of all incoming images have the same distribution characteristic. This reduces the variations of illumination conditions across recording sessions and the skin color difference across speakers. We found that the distribution of whole pixel values of the images in the database can be approximated by a Gaussian distribution. Thus, we set this Gaussian distribution as a target distribution and each image is transformed into a new image which follows the target distribution by the histogram specification technique (Gonzalez & Woods, 2001).
3. To find the mouth corners, we apply the bi-level thresholding method on the images. The thresholding is applicable for detecting mouth corners because there are always dark regions between upper and lower lips; when the mouth is open, the oral cavity appears dark, and when the mouth is closed, the boundary line between the lips appears dark. After thresholding, the left and the right end points of the dark region are the mouth corners. The mouth region is cropped based on the found corner points, so that we obtain scale- and rotation-invariant lip region images of 44x50 pixels.

4. For each pixel point, the mean value over an utterance is subtracted. Let *I(m,n,t)* be the *(m,n)*-th pixel value of the lip region image at the *t*-th frame. Then, the pixel value after mean subtraction is given by

$$J(m,n,t) = I(m,n,t) - \frac{1}{T}\sum_{t=1}^{T} I(m,n,t)\,, \tag{10}$$

   where T is the total length of the utterance. This is similar to the CMS technique in acoustic feature extraction and removes unwanted variations across image sequences due to the speakers' appearances and the different illumination conditions.
5. Finally, we apply PCA to find the main linear modes of variations and reduce the feature dimension. If we let **x** be the n_0-dimensional column vector for the pixel values of the mean-subtracted image, the n-dimensional visual feature vector **s** is given by

$$\mathbf{s} = P^T(\mathbf{x} - \bar{\mathbf{x}})\,, \tag{11}$$

   where $\bar{\mathbf{x}}$ is the mean of **x** for all training data, P is the n_0-by-n matrix whose columns are the eigenvectors for the n largest eigenvalues of the covariance matrix for all **x**'s. Here, n is much smaller than n_0(=44x50=2200) so that we obtain a compact visual feature vector. We set n to 12 in our experiment so that we obtain 12 static features for each frame. We also use the temporal derivatives of the static features as in the acoustic feature extraction.

Figure 9 shows the mean image of the extracted lip region images and the four most significant principal modes of intensity variations by ±2 standard deviations (std.) for the training data of the DIGIT database. We can see that each mode explains distinct variations

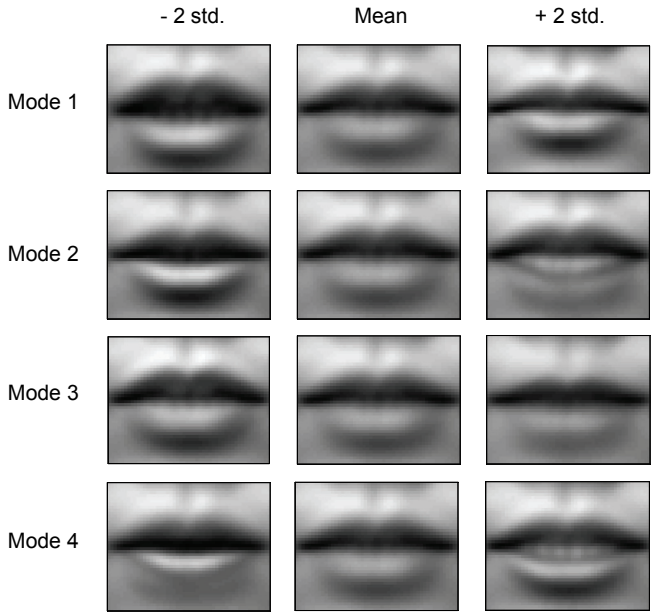


Fig. 9. First four principal modes of variations in the lip region images.

occurring in the mouth images. The first mode mainly accounts for the mouth opening. The second mode shows the protrusion of the lower lip and the visibility of the teeth. In the third mode, the protrusion of the upper lip and the changes of the shadow under the lower lip are shown. The fourth mode largely describes the visibility of the teeth.

4.4 Recognizer

The recognizer is composed of typical left-to-right continuous HMMs having Gaussian mixture models (GMMs) in each state. We use the whole-word model which is a standard approach for small vocabulary speech recognition tasks. The number of states in each HMM is set to be proportional to the number of the phonetic units of the corresponding word. The number of Gaussian functions in each GMM is set to three, which is determined experimentally. The HMMs are initialized by uniform segmentation of the training data onto the HMMs' states and iterative application of the segmental k-means algorithm. For training the HMMs, the popular Baum-Welch algorithm is used (Rabiner, 1989).

4.5 Results

First, we compare the reliability measures presented in Section 3.2. The audio-visual fusion is performed using the neural networks having five sigmoidal hidden neurons because use of more neurons did not show performance improvement. The Levenberg-Marquardt algorithm (Hagan & Menhaj, 1994), which is one of the fastest training algorithms of neural networks, is used to train the networks.

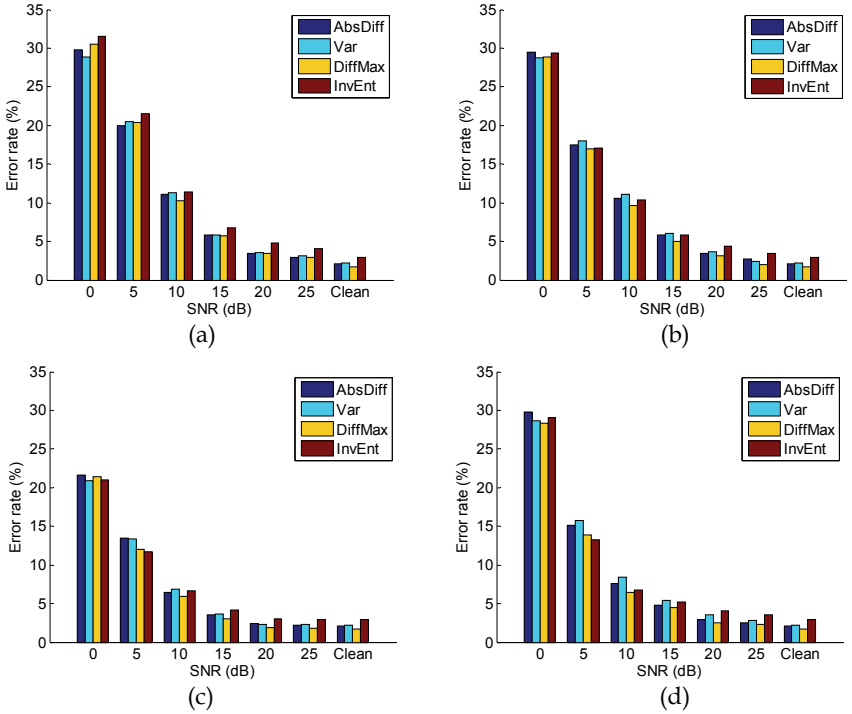


Fig. 10. Comparison of the reliability measures for the DIGIT database. (a) WHT. (b) F16. (c) FAC. (d) OPR.

Figures 10 and 11 compare the reliability measures for each database, respectively. It is observed that **DiffMax** shows the best recognition performance in an overall sense. The inferiority of **AbsDiff**, **Var** and **InvEnt** to **DiffMax** is due to their intrinsic errors in measuring reliabilities from the HMM's outputs (Lewis & Powers, 2004): Suppose that we have four classes for recognition and the HMMs' outputs are given as probabilities (e.g., [0.2, 0.4, 0.1, 0.5]). We want to get the maximum reliability when the set of the HMMs' outputs is [1, 0, 0, 0] after sorting. However, **AbsDiff** and **Var** have the maximum values when the set of the HMMs' outputs is [1, 1, 0, 0]. Also, they have the same values for [1, 0, 0, 0] and [1, 1, 1, 0], which are actually completely different cases. As for **InvEnt**, when we compare the cases of [0.1, 0.1, 0.4, 0.4] and [0.1, 0.2, 0.2, 0.5], the former has a higher value than the latter, which is the opposite of what we want.

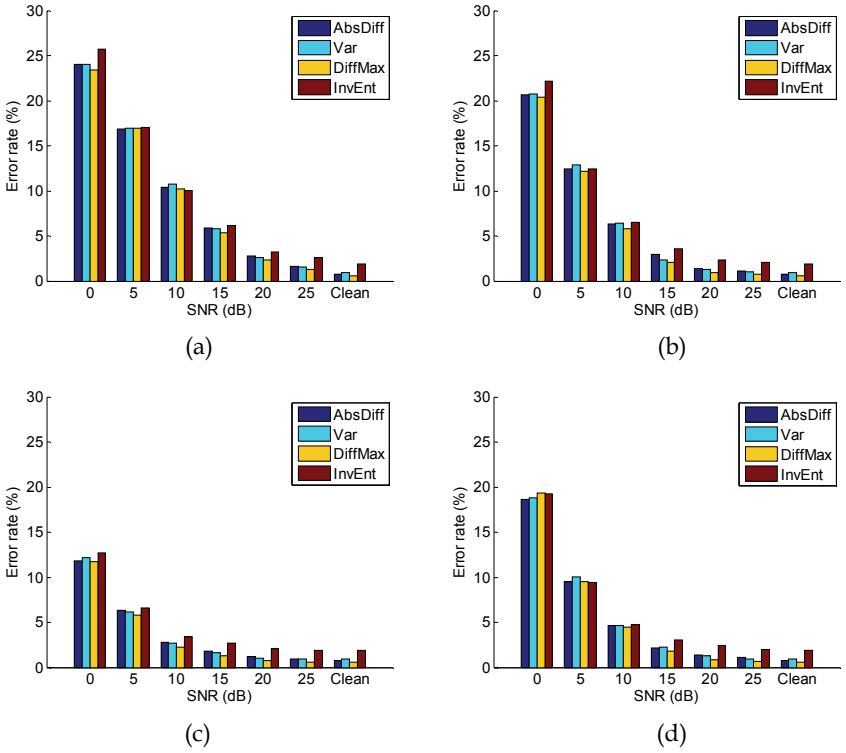


Fig. 11. Comparison of the reliability measures for the CITY database. (a) WHT. (b) F16. (c) FAC. (d) OPR.

Next, we examine the unimdal and the bimodal recognition performance. Figures 12 and 13 compare the acoustic-only, the visual-only and the integrated recognition performance in error rates for the two databases, respectively. From the results, we can observe the followings:

1. The acoustic-only recognition shows nearly 100% for clean speech but, as the speech contains more noise, its performance is significantly degraded; for some noise the error rate is even higher than 70% at 0dB.

2. The error rate of the visual-only recognition is 36.1% and 22.0% for each database, respectively, which appears constant regardless of noise conditions. These values are larger than the acoustic-only recognition performance for clean speech but smaller than that for noisy speech.
3. The performance of the integrated system is at least similar to or better than that of the unimodal system. Especially, the synergy effect is prominent for 5dB~15dB. Compared to the acoustic-only recognition, relative reduction of error rates by the bimodal recognition is 39.4% and 60.4% on average for each database, respectively. For the high-noise conditions (i.e., 0dB~10dB), relative reduction of error rates is 48.4% and 66.9% for each database, respectively, which demonstrates that the noise-robustness of recognition is achieved.
4. The neural network successfully works for untrained noise conditions. For training the neural network, we used only clean speech and 20dB, 10dB and 0dB noisy speech corrupted by white noise. However, the integration is successful for the other noise levels of the same noise source and the noise conditions of the other three noise sources.

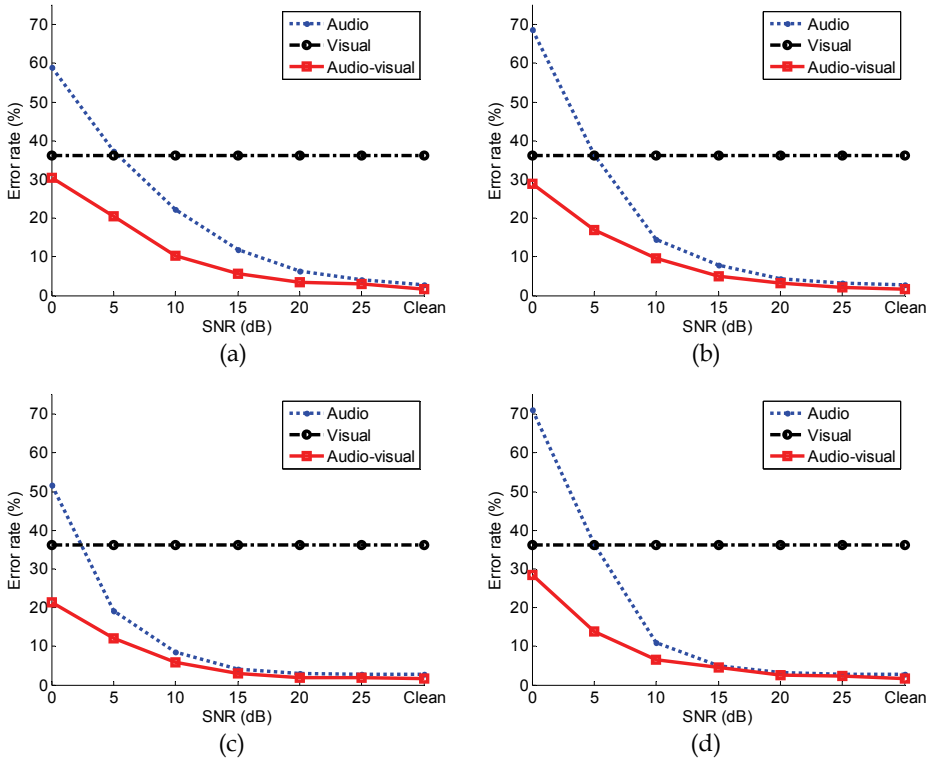


Fig. 12. Recognition performance of the unimodal and the bimodal systems in error rates (%) for the DIGIT database. (a) WHT. (b) F16. (c) FAC. (d) OPR.

Figure 14 shows the integration weight values (the means and the standard deviations) determined by the neural network with respect to SNRs for the DIGIT database. It is

observed that the automatically determined weight value is large for high SNRs and small for low SNRs, as expected.

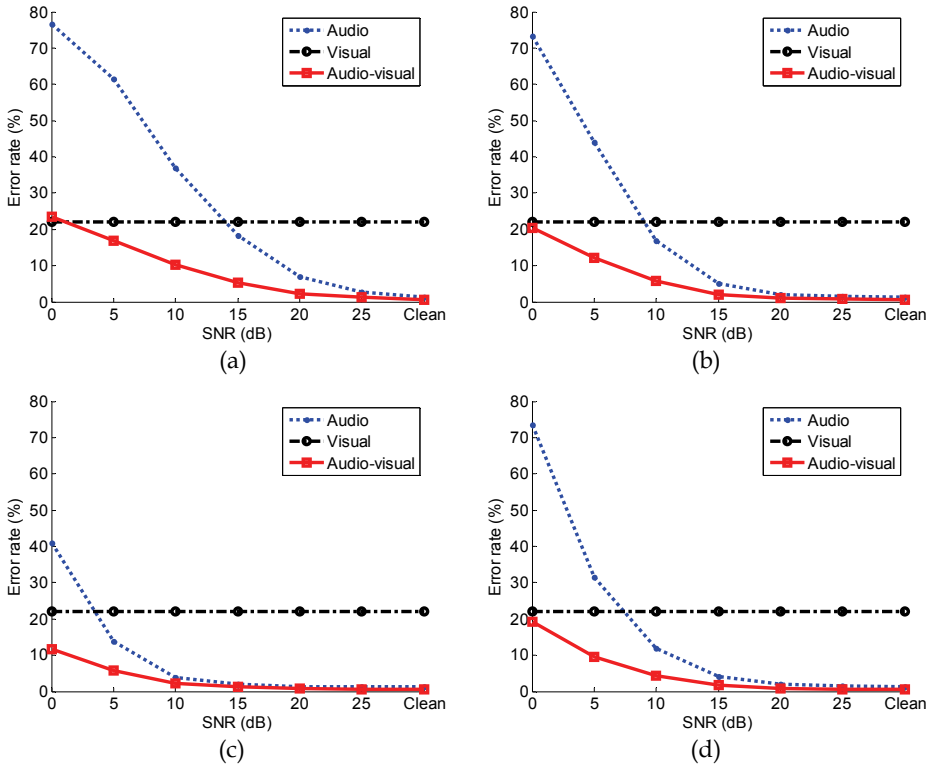


Fig. 13. Recognition performance of the unimodal and the bimodal systems in error rates (%) for the CITY database. (a) WHT. (b) F16. (c) FAC. (d) OPR.

5. Conclusion

This chapter addressed the problem of information fusion for AVSR. We introduced the bimodal nature of speech production and perception by humans and defined the goal of audio-visual integration. We reviewed two existing approaches for implementing audio-visual fusion in AVSR systems and explained the preference of decision fusion to feature fusion for constructing noise-robust AVSR systems. For implementing a noise-robust AVSR system, different definitions of the reliability of a modality were discussed and compared. A neural network-based fusion method was described for effectively utilizing the reliability measures of the two modalities and producing noise-robust recognition performance over various noise conditions. It has been shown that we could successfully obtain the synergy of the two modalities.

The audio-visual information fusion method shown in this chapter mainly aims at obtaining robust speech recognition performance, which may lack modelling of complicated humans' audio-visual speech perception processes. If we consider that the humans' speech

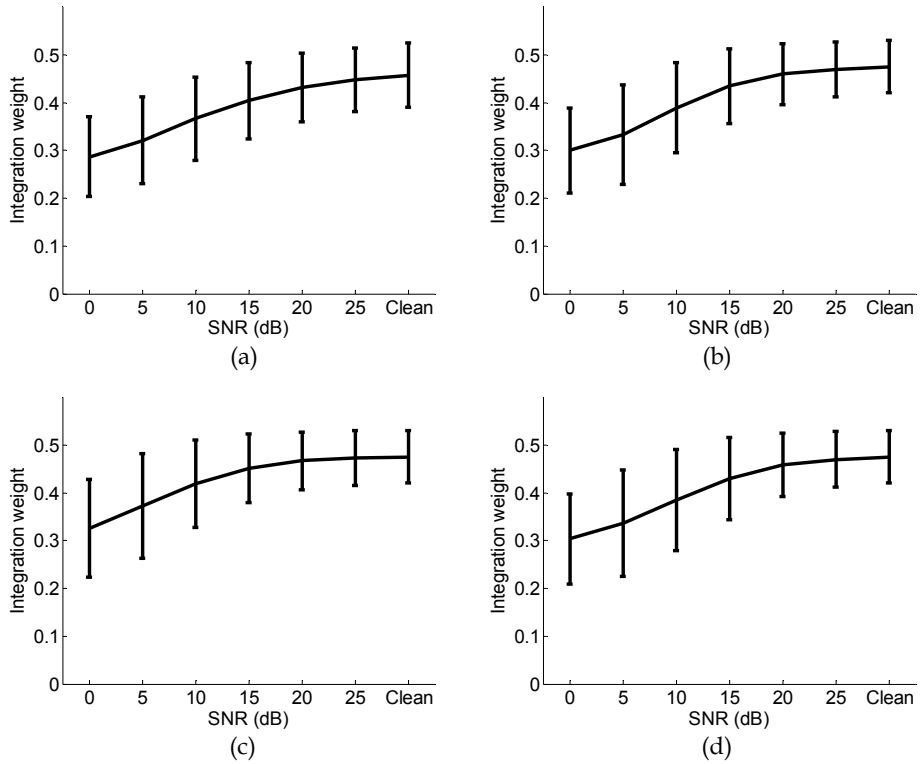


Fig. 14. Generated integration weights with respect to the SNR value for the DIGIT database.

perception performance is surprisingly good, it is worth investigating such perception processes carefully and incorporating knowledge about them into implementing AVSR systems. Although it is still not clearly understood about such processes, it is believed that the two perceived signals complicatedly interact at multiple stages in humans' sensory systems and brains. As discussed in Section 2.1, visual information helps to detect acoustic speech under the presence of noise, which suggests that the two modalities can be used at the early stage of AVSR for speech enhancement and selective attention. Also, it has been suggested that there exist an early activation of auditory areas by visual cues and a later speech-specific activation of the left hemisphere possibly mediated by backward-projections from multisensory areas, which indicates that audio-visual interaction takes place in multiple stages sequentially (Hertrich et al., 2007). Further investigation of biological multimodal information processing mechanisms and modelling them for AVSR would be a valuable step toward mimicking humans' excellent AVSR performance.

6. References

Adjoudani, A. & Benoît, C. (1996). On the integration of auditory and visual parameters in an HMM-based ASR, In: *Speechreading by Humans and Machines: Models, Systems, and*

Applications, Stork, D. G. & Hennecke, M. E., (Eds.), pp. 461-472, Springer, Berlin, Germany.

Arnold, P. & Hill, F. (2001). Bisensory augmentation: a speechreading advantage when speech is clearly audible and intact. *British Journal of Psychology*, Vol. 92, (2001) pp. 339-355.

Benoît, C.; Mohamadi, T. & Kandel, S. D. (1994). Effects of phonetic context on audio-visual intelligibility of French. *Journal of Speech and Hearing Research*, Vol. 37, (October 1994) pp. 1195-1203.

Benoît, C. (2000). The intrinsic bimodality of speech communication and the synthesis of talking faces, In: *The Structure of Multimodal Dialogue II*, Taylor, M. M.; Nel, F. & Bouwhis, D. (Eds.), John Benjamins, Amsterdam, The Netherlands.

Braida, L. (1991). Crossmodal integration in the identification of consonant segments. *The Quarterly Journal of Experimental Psychology Section A*, Vol. 43, No. 3, (August 1991) pp. 647-677.

Bregler, C. & Konig, Y. (1994). Eigenlips for robust speech recognition, *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 669-672, Adelaide, Autralia, 1994.

Calvert, G. A.; Bullmore, E. T.; Brammer, M. J.; Campbell, R.; Williams, S. C. R.; McGuire, P. K.; Woodruff, P. W. R.; Iversen, S. D. & David, A. S. (1997). Activation of auditory cortex during silent lipreading. *Science*, Vol. 276, (April 1997) pp. 593-596.

Campbell, R. (1988). Tracing lip movements: making speech visible. *Visible Language*, Vol. 22, No. 1, (1988) pp. 32-57.

Chibelushi, C. C.; Deravi, F. & Mason, J. S. D. (2002). A review of speech-based bimodal recognition. *IEEE Transactions on Multimedia*, Vol. 4, No. 1, (March 2002) pp. 23-37.

Conrey, B. & Pisoni, D. B. (2006). Auditory-visual speech perception and synchrony detection for speech and nonspeech signals. *Journal of Acoustical Society of America*, Vol. 119, No. 6, (June 2006) pp. 4065-4073.

Davis, S. B. & Mermelstein. (1980). Comparison of parametric representations for monosyllable word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 28, No. 4, (1980) pp. 357-366.

Dupont, S. & Luettin, J. (2000). Audio-visual speech modeling for continuous speech recognition. *IEEE Transactions on Multimedia*, Vol. 2, No. 3, (September 2000) pp. 141-151.

Ernest, M. O. & Banks, M. S. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, Vol. 415, No. 6870, (January 2002) pp. 429-433.

Fowler, C. A. (1986). An event approach to the study of speech perception from a direct-realist perspective. *Journal of Phonetics*, Vol. 14, (1986) pp. 3-28.

Gonzalez, R. C. & Woods, R. E. (2001). *Digital Image Processing*, Addison-Wesley Publishing Company.

Grant, K. W. & Seitz, P.-F. (2000). The use of visible speech cues for improving auditory detection of spoken sentences. *Journal of Acoustical Society of America*, Vol. 103, No. 3, (September 2000) pp. 1197-1208.

Gurbuz, S.; Tufekci, Z.; Patterson, E. & Gowdy, J. (2001). Application of affine-invariant Fourier descriptors to lipreading for audio-visual speech recognition, *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 177-180, Salt Lake City, UT, USA, May 2001.

Hagan, M. T. & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, (1994) pp. 989-993.

Hertrich, I.; Mathiak, K.; Lutzenberger, W.; Menning, H. & Ackermann, H. (2007). Sequential audiovisual interactions during speech perception: a whole-head MEG study. *Neuropsychologia*, Vol. 45, (2007) pp. 1342-1354.

Hornik, K.; Stinchcombe, M. & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, Vol. 2, No. 5, (1989) pp. 359-366.

Huang, X.; Acero, A. & Hon, H.-W. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Prentice-Hall, Upper Saddle River, NJ, USA.

Kaynak, M. N.; Zhi, Q.; Cheok, A. D.; Sengupta, K.; Jian, Z. & Chung, K. C. (2004). Lip geometric features for human-computer interaction using bimodal speech recognition: comparison and analysis. *Speech Communication*, Vol. 43, No. 1-2, (January 2004) pp. 1-16.

Kim, J. & Davis, C. (2004). Investigating the audio-visual speech detection advantage. *Speech Communication*, Vol. 44, (2004) pp 19-30.

Klatt, D. H. (1979). Speech perception: a model of acoustic-phonetic analysis and lexical access. *Journal of Phonetics*, Vol. 7, (1979) pp. 279-312.

Lee, J.-S. & Park, C. H. (2006). Training hidden Markov models by hybrid simulated annealing for visual speech recognition, *Proceedings of the International Conference on Systems, Man and Cybernetics*, pp. 198-202, Taipei, Taiwan, October 2006.

Lee, J.-S. & Park, C. H. (2008). Robust audio-visual speech recognition based on late integration. *IEEE Transactions on Multimedia*, Vol. 10, No. 5, (August 2008) pp. 767-779.

Lewis, T. W. & Powers, D. M. W. (2004). Sensor fusion weighting measures in audio-visual speech recognition, *Proceedings of the Conference on Australasian Computer Science*, pp. 305-314, Dunedine, New Zealand, 2004.

Liberman, A. & Mattingly, I. (1985). The motor theory of speech perception revised. *Cognition*, Vol. 21, (1985) pp. 1-33.

Lucey, S. (2003). An evaluation of visual speech features for the tasks of speech and speaker recognition, *Proceedings of International Conference on Audio- and Video-based Biometric Person Authentication*, pp. 260-267, Guilford, UK, June 2003.

Macaluso, E.; George, N.; Dolan, R; Spence, C. & Driver, J. (2004). Spatial and temporal factors during processing of audiovisual speech: a PET study. *NeuroImage*, Vol. 21, (2004) pp. 725-732.

Massaro, D. W. (1987). *Speech Perception by Ear and Eye: A Paradigm for Psychological Inquiry*, Erlbaum.

Massaro, D. W. (1998). *Perceiving Talking Faces: From Speech Perception to a Behavioral Principle*, MIT Press.

Massaro, D. W. (1999). Speechreading: illusion or window into pattern recognition. *Trends in Cognitive Sciences*, Vol. 3, No. 8, (August 1999) pp. 310-317.

Massaro, D. W. & Stork, D. G. (1998). Speech recognition and sensory integration: a 240-year-old theorem helps explain how people and machines can integrate auditory and visual information to understand speech. *American Scientist*, Vol. 86, No. 3, (May-June 1998) pp. 236-242.

Matthews, I.; Bangham, J. A. & Cox. S. (1996). Audio-visual speech recognition using multiscale nonlinear image decomposition, *Proceedings of the International Conference on Speech and Language Processing*, pp. 38-41, Philadelphia, USA, 1996.

Matthews, I.; Potamianos, G.; Neti, C. & Luettin, J. (2001). A comparison of model and transform-based visual features for audio-visual LVCSR, *Proceedings of the International Conference on Multimedia and Expo*, pp. 22-25, Tokyo, Japan, April 2001.

McClelland, J. L. & Elman, J. L. (1986). The TRACE model of speech perception. *Cognitive Psychology*, Vol. 18, (1986) pp. 1-86.

McGurk, H. & MacDonald, J. (1976). Hearing lips and seeing voices. *Nature*, Vol. 264, (December 1976) pp. 746-748.

Pekkola, J.; Ojanen, V.; Autti, T.; Jääskeläinen. I. P.; Möttönen, R.; Tarkiainen, A. & Sams, M. (2005). Primary auditory cortex activation by visual speech: an fMRI study at 3T. *NeuroReport*, Vol. 16, No. 2, (February 2005) pp. 125-128.

Potamianos, G. & Neti, C. (2000). Stream confidence estimation for audio-visual speech recognition, *Proceedings of the International Conference on Spoken Language Processing*, pp. 746-749, Beijing, China, 2000.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, Vol. 77, No. 2, (Febrary 1989) pp. 257-286.

Robert-Ribes, J.; Schwartz, J.-L.; Lallouache, T. & Escudier, P. (1998). Complementarity and synergy in bimodal speech: auditory, visual, and audio-visual identification of French oral vowels in noise. *Journal of Acoustical Society of America*, Vol. 103, No. 6, (June 1998) pp. 3677-3689.

Rogozan, A & Deléglise, P. (1998). Adaptive fusion of acoustic and visual sources for automatic speech recognition. *Speech Communication*, Vol. 26, No. 1-2, (October 1998) pp. 149-161.

Ross, L. A.; Saint-Amour, D.; Leavitt, V. M.; Javitt, D. C. & Foxe, J. J. (2007). Do you see what I am saying? Exploring visual enhancement of speech comprehension in noisy environments. *Cerebral Cortex*, Vol. 17, No. 5, (May 2007) pp. 1147-1153.

Ruytjens, L.; Albers, F.; van Dijk, P.; Wit, H. & Willemsen, A. (2007). Activation in primary auditory cortex during silent lipreading is determined by sex. *Audiology and Neurotology*, Vol. 12, (2007) pp. 371-377.

Schwartz, J.-L.; Berthommier, F. & Savariaux, C. (2004). Seeing to hear better : evidence for early audio-visual interactions in speech identification. *Cognition*, Vol. 93, (2004) pp. B69-B78.

Sekiyama, K. & Tohkura, Y. (1993). Inter-language differences in the influence of visual cues in speech perception. *Journal of Phonetics*, Vol. 21, (1993) pp. 427-444.

Sharma, R.; Pavlović, V. I. & Huang, T. S. (1998). Toward multimodal human-computer interface. *Proceedings of the IEEE*, Vol. 86, No. 5, (May 1998) pp. 853-869.

Stein, B. & Meredith, M. A. (1993). *The Merging of Senses*, MIT Press, MA, USA.

Summerfield, A. Q. (1987). Some preliminaries to a comprehensive account of audio-visual speech perception, In: *Hearing by Eye: The Psychology of Lip-reading*, Dodd, B. & Campbell, R. (Eds.), pp. 3-51, Lawrence Erlbarum, London, UK.

Varga, A. & Steeneken, H. J. M. (1993). Assessment for automatic speech recognition: II NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication*, Vol. 12, No. 3, (1993) pp. 247-251.

16

Multi-Stream Asynchrony Modeling for Audio Visual Speech Recognition

Guoyun Lv, Yangyu Fan, Dongmei Jiang and Rongchun Zhao
NorthWestern Polytechnical University (NWPU)
127 Youyi Xilu, Xi'an 710072,
P.R.China

1. Introduction

The success of currently available speech recognition systems was restricted to relative controlled laboratory conditions or application fields. The performance of these systems rapidly degraded in more realistic application environments (Lippmann, 1997). Since the vast majority of background noise were introduced by transmission channel, microphone distance or environment noise, some new audio feature extraction methods (Perceptual Linear Predictive (PLP), RelAtive SpecTrAl (RASTA) (Hermansky, 1990; Hermansky,1994), and other ways such as vocal tract length normalization and parallel model combination for speech and noise, were used to describe the complex speech variations. Though these methods improved the system robustness to noisy environment to some extent, but their advantage was limited.

Since both human speech production and perception are bimodal in nature (Potamianos et al, 2003), visual speech information from the speaker's mouth has been successfully shown to improve noisy robustness of automatic speech recognizers (Dupont & Luettin 2000; Gravier et al, 2002). There are two main challenging problems in the reported Audio-Visual Speech Recognition (AVSR) systems (Nefian et al, 2002; Gravier et al, 2002): First, the design of the visual front end, i.e. how to obtain the more static visual speech feature; second, how to build a audio-visual fusion model that describes the inherent correlation and asynchrony of audio and visual speech. In this paper, we concentrate on the latter issue.

Previous works on combining multiple features can be divided into three categories: feature fusion, decision fusion and model fusion. Model fusion seems to be the best technique to integrate information from two or more streams. However, the experiments results of many AVSR systems show that although the visual activity and audio signal are correlative, but they are not synchronous, the visual activity often precedes the audio signal about 120ms (Gravier et al, 2002; Potamianos et al, 2003) . Each AVSR system should take the asynchrony into account.

Since hidden Markov model (HMM) based models achieve promising performance in speech recognition, many literatures have adopted Multi-Stream HMM (MSHMM) to integrate audio and visual speech feature (Gravier et al, 2002; Potamianos et al, 2003; Nefian et al, 2002), such as State Synchrony Multi-Stream HMM (SS-MSHMM), State Asynchrony Multi-Stream HMM (SA-MSHMM) (Potamianos et al, 2003), Product HMM (PHMM) (Dupont, 2000), Couple HMM (CHMM) and Factorial HMM (FHMM) (Nefian et al, 2002)

and so on. In these models, audio and visual features are imported to two or more parallel HMMs with different topology structures respectively, but on some nodes, such as phone, syllable et al; some constraints are imposed to limit the asynchrony of audio and visual streams to state (phone or syllable) level. These MSHMMs describe the correlation and asynchrony of audio and visual speech to some extent. Compared with the single stream HMM, system performance is improved especially in noisy speech environment, but these MSHMMs can only use phone as recognition unit for speech recognition task on a middle or large vocabulary audio-visual database. It constrains the audio and visual stream to be synchronous in the phonemic boundary. However, the asynchrony of audio and visual stream exceeds phonemic boundary in many conditions. The better recognition rate should be obtained if loosing the asynchrony limitation of audio and visual stream.

In recent years, it was an active research topic to adopt Dynamic Bayesian Network (DBN) for speech recognition (Bilmes, 2002; Murphy, 2002; Zweig, 1998). DBN model is a statistic model that can represent multiple collections of random variables as they evolve over time. It is appropriate to describe complex variables and conditional relationship among the variables, since it can automatically learn the conditional probability distribution among the variables, with better extensible performance. Bilmes, Zweig et al, used single stream DBN model for isolated words and small vocabulary speech recognition (Bilmes et al, 2001; Lv et al, 2007). Zhang YM proposed a multi-stream DBN model for speech recognition by combining different audio features (MFCC, PLP, RASTA) (Zhang et al, 2003), although the model described the asynchrony of audio and visual streams by sharing the same word node, while in fact, there are not asynchrony for different audio features from the same voice. N. Gowdy expanded this model for audio-visual speech recognition (Gowdy et al, 2003), an improvement was obtained in word accuracy, while between the word nodes, and each stream is not complete independence, which affected the asynchrony of both streams to some extent. Bimes proposed a general multi-stream asynchrony DBN model structure (Bilmes & Bartels, 2005), in this model, the word transition probability is determined by the state transitions and the state positions both in the audio stream and in the visual stream. Between the word nodes, two streams have their own nodes and the dependent relationship between the nodes. But no more experimental results were given.

In this work, we use the general multi-stream DBN model structure given in (Bilmes & Bartels, 2005) as our baseline model. In (Bilmes & Bartels, 2005), both in audio stream and in visual stream, each word is composed of the fixed number of states, and each state is associated with observation vectors. The training parameters are very tremendous, especially for the task of large vocabulary speech recognition. In order to reduce the training parameters, in our model, both in audio stream and in visual stream, each word is composed of its corresponding phones sequence, and each phone is associated with observation vector. Since phones are shared by all the words, the training parameter will be enormously reduced, and we name it Multi-Stream Asynchrony DBN (MS-ADBN) model. But MS-ADBN model is word model whose recognition basic units are words. It is not appropriate for the task of large vocabulary AVSR. Base on MS-ADBN model, an extra hidden node level—state is added between phone node level and observation variable level in both stream, resulting in a novel Multi-stream Multi-states Asynchrony DBN (MM-ADBN) model. In MM-ADBN model, each phone is composed of fixed number of states, and each state is associated with observation vector, besides word, dynamic pronunciation process of phone is also described. Its recognition basic units are phones, and can be used for large vocabulary audio-visual speech recognition.

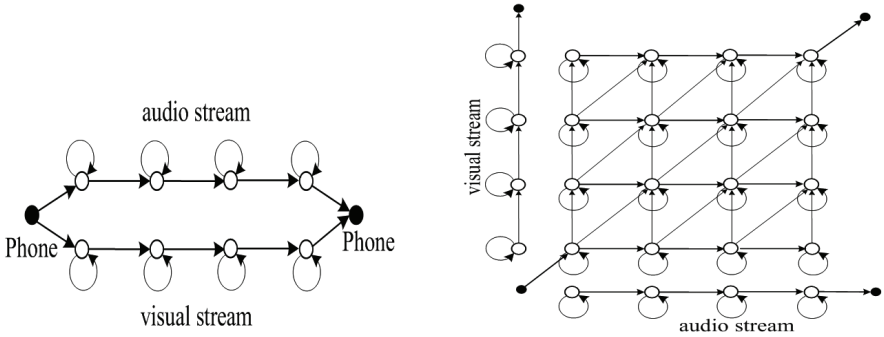
The paper is organized as follows. Section 2 describes the structures and conditional probability distributions of the proposed MS-ADBN model and MM-ADBN model. In section 3, experiments and evaluations are given, followed by our conclusions in section 4.

2. Multi-stream asynchrony model

In this section, at first, we briefly review the previous MSHMM, and then we describe the multi-stream asynchrony DBN model proposed in our work. Finally, we make simple comparisons for these audio-visual speech recognition models.

2.1 State asynchrony multi-stream HMM

Multi-Stream hidden Markov model (MSHMM) was a popular method within audio-visual model fusion framework. The MSHMM linearly combines the class log-likelihoods based on the audio-only and video-only observations at a number of possible stages (such as state, phone et al). In early most cases, the synchronous point of audio stream and visual stream is at the HMM state level, and we name it State Synchronous MSHMM (SS-MSHMM). To take asynchrony of audio and visual stream into account, the synchronous point should be taken to a coarser level, such as phone, syllable, or word level. However, on one hand, for middle and large vocabulary speech recognition, the phone recognition unit must be used; on the other hand, to implement easily, previous popular works often use the state asynchrony multi-stream HMM (SA-MSHMM) (Gravier, 2002; Nefian et al, 2002), and the synchronous points are taken to the phonemic boundaries. Because of the limitation of HMM expression ability, such a model can be implemented as a product HMM (PHMM), as illustrated in Fig. 1. Typically, SA-MSHMM with four audio and four video HMM states is given in Fig. 1 (a), and its corresponding product HMM is given in Fig. 1(b).



(a) SA-MSHMM with four audio and video states (b) Corresponding product HMM

Fig. 1. Illustration of SA-MSHMM and its corresponding product HMM

The observation probability can be described as:

$$b_j(o_t) = \prod_{s\in\{a,v\}} [\sum_{m=1}^{M_s} \omega_{jsm} N(o_t^s; \mu_{jsm}; \sigma_{jsm})]^{\lambda_s} \tag{1}$$

Where, $s = a, v$, respectively for audio and visual stream. λ_s is stream exponent, $\lambda_a + \lambda_v = 1$; j is state node, m means mth Gaussian mixture unit, M_s is number of Gaussian mixture unit

in s^{th} stream, ω_{jsm} denotes weight value, μ and σ is the mean and covariance of Gaussian distribution $N(.)$.

Although SA-MSHMM can describe the asynchrony of audio and visual stream to some extent, But problems remain due to the inherent limitation of the HMM structure. On one hand, for large vocabulary speech recognition tasks, phones are the basic modeling units, the model will force the audio stream and the visual stream to be synchronized at the timing boundaries of phones, which is not coherent with the fact that the visual activity often precedes the audio signal even by 120 ms. On the other hand, Once a little slight varieties are done on MSHMM, a large amount of human effort must be placed into making significant modifications on top of already complex software without having any guarantees about their performance. So a new and unified multi-stream model framework is expected to loose the limitation of asynchrony of audio stream and visual stream to the coarser level.

2.2 MS-ADBN model

A Dynamic Bayesian Network (DBN) is a statistical model that can represent collections of random variables and their dependency relationships as they evolve over time. HMM is just special case of much more general DBN model. Comparing with HMM, DBN model has a more flexible and extensible structure, and explicitly describes the hierarchical relationship of main components (e. g word, phone, state and observation) of speech recognition. In general, the DBN model meets two conditions: 1) except the initial frame, the topology structure is same in each frame; 2) the condition probability relationship between the frames follows the one-order Markov model. Additionally, Uniform training and decoding algorithm make the implement of DBN model become easier.

Since DBN model has some preponderant on describing the complex model structure, multi-stream DBN model is expected to model the audio-visual speech recognition structure by loosing the asynchrony of the audio stream and visual stream.

Fig. 2 illustrates the recognition structure of a multi-stream asynchrony DBN (MS-ADBN) model. It is composed of a Prologue part (initialization), a Chunk part that is repeated every time frame (t), and a closure of sentence with an Epilogue part. Abbreviation of every node is denoted in the parentheses: (W) is the word unit in a sentence; (WT) is the occurrence of a transition from one word to another word; (PP1) and (PP2) are the position of the current phone in the current word; (PT1) and (PT2) are the occurrence of a transition from a phone to another phone; (P1) and (P2) is the phone node; O1 is acoustic observation; O2 is visual observation vector. The nodes with shade are the observation variables, and the nodes without shades are the hidden state variables.

In MS-ADBN model, the word variable and word transition variable are at the top of the structure, when a word transition occurs, it will reset (PP1) and (PP2) to their initial value, hence audio stream and visual stream are forced to be synchronous in the same word node.

While between the word nodes, each stream has its own independence nodes and conditional probability distributions between the nodes, each word is composed of its corresponding composed phones, and each phone is associated with observation features. Namely, it allows two independence representations for dynamic pronunciation process of a word in this model. Additionally, word transition is determined by audio and visual steam together, to make word transition occur, we must have that both PP1 and PP2 are the last phone of current word, as well as both PT1 and PT2 occurs. Comparing with MSHMM, the asynchrony of audio and visual stream is really loosed to word level.

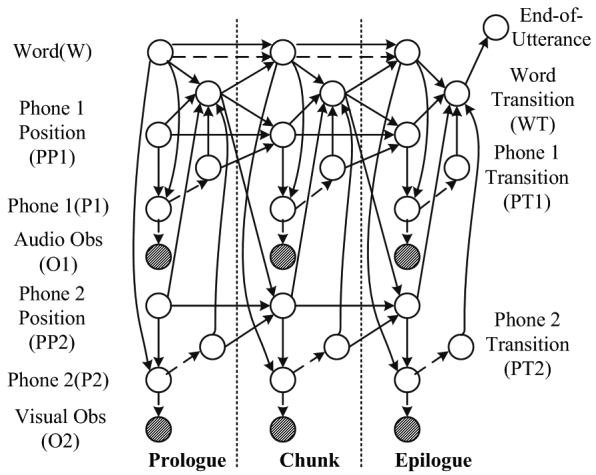


Fig. 2. MS-ADBN speech recognition model

While MS-ADBN model only describes the dynamic pronunciation process of word, it is a word model whose basic recognition units are words. It is only appropriate for small vocabulary speech recognition. For large vocabulary audio visual speech recognition, the less recognition sub-word units—phone, should be modeled.

2.3 MM-ADBN model

To describe the dynamic process of phone, we proposed a novel multi-stream multi-states asynchrony DBN (MM-ADBN) model given in Fig. 3.

It can be seen that MM-ADBN model is an augmentation of the MS-ADBN model, to which we add some extra hidden nodes (state, state position and state transition) and corresponding conditional probability relationships between phone variable level and observation variable level in both audio stream and visual stream, and new additive parts are labelled with red color in Fig.3. In MM-ADBN model, both in audio stream and in visual stream, each phone is composed of fixed number of states, and each state is associated with observation vector. Hence, it is a phone model whose basic recognition units are phones, and can be used for large vocabulary audio-visual speech recognition. In Fig. 3, the definitions of the word and phone related nodes are the same as those in the MS-ADBN model. The new notations: (SP1) and (SP2) are the position of the current state in the current phone; (ST1) and (ST2) are the occurrence of a transition from a state to another state, defined similarly as (PT1) or (PT2); (S1) and (S2) are the state node. Suppose the input speech contains T frames of features, and the set of all the hidden nodes is denoted as $H_{1:T}$.

$$\begin{aligned} H_{1:T} = (&W_{1:T}, WT_{1:T}, PP1_{1:T}, PP2_{1:T}, PT1_{1:T}, PT2_{1:T}, SP1_{1:T}, \\ &SP2_{1:T}, ST1_{1:T}, ST2_{1:T}, P1_{1:T}, P2_{1:T}, S1_{1:T}, S2_{1:T}) \end{aligned} \tag{2}$$

For the model given in Fig. 3, the probability of observation can be computed as

$$p(O1_{1:T}, O2_{1:T}) = \sum_{H_{1:T}} p(H_{1:T}, O1_{1:T}, O2_{1:T}) \tag{3}$$

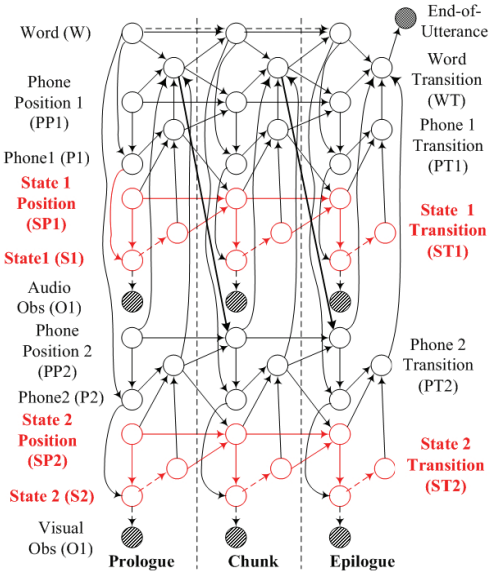


Fig. 3. MM-ADBN speech recognition model

For a better understanding of MM-ADBN model, we describe each node's Conditional Probability Distributions (CPD) as follows.

a. Observation Feature (O1 and O2). The observation feature Ox_t is a random function of the state Sx_t with the CPD $P(Ox_t|Sx_t)$, which is described by a normal Gaussian Model $N(O_t, \mu_{Sx_t}, \sigma_{Sx_t})$ with mean μ_{Sx_t} and covariance σ_{Sx_t}, where symbol x denotes audio stream or visual stream, x=1 means audio stream and x=2 means visual stream, which also be used in the following expression.

b. State transition probability (ST1 and ST2), which describe the probability of the transition from the current state to the next state in the audio stream and visual stream respectively. The CPD $P(STx_t|Sx_t)$ is random since each state has a nonzero probability for staying at the current state or moving to the next state, in the initial frame, the CPD is assumed as 0.5.

c. State node (S1 and S2), since each phone is composed of fixed number of states, giving the current phone and the position of the current state in the phone, the state Sx_t is known with certainty.

$$p(Sx_t = j | Px_t = i, SPx_t = m) = \begin{cases} 1 & if\ j\ \text{is the}\ m-th\ \text{state of the phone}\ i \\ 0 & otherwise \end{cases} \quad (4)$$

d. State position (SP1 and SP2), in the initial frame, the initial value is zero. In the other time slices, its CPD has three behaviors, (i) It might not change from one frame to the next frame when there is no state transition and phone transition; (ii) It might increment

by 1 when there is a state transition and the model is not in the last state of the phone; (iii) It might be reset to 0 when a phone transition occurs.

$$p(SPx_t = j \mid SPx_{t-1} = i, PTx_{t-1} = m, STx_{t-1} = n) = \begin{cases} 1 & m = 1, j = 0 \\ 1 & \text{m} = 0,\text{n} = 0,\text{j} = \text{i} \\ 1 & m = 0, n = 1, j = i + 1 \\ 0 & otherwise \end{cases} \tag{5}$$

e. Phone node (P1 and P2). Each word is composed of its corresponding phones, giving the current word and the position of the current phone in the word, the phone Px_t is known with certainty.

$$p(Px_t = j \mid W_t = i, PPx_t = m) = \begin{cases} 1 & j \text{ is the } m-th \text{ phone of the word } i \\ 0 & otherwise \end{cases} \tag{6}$$

f. Phone position (PP1 and PP2). In the initial frames, the initial value is zero. In the other time frame, the CPD is as follows.

$$p(PPx_t = j \mid PPx_{t-1} = i, WT_{t-1} = m, PTx_{t-1} = n) = \begin{cases} 1 & m = 1, j = 0 \text{ or } \text{m} = 0,\text{n} = 0,\text{j} = \text{i} \\ 1 & m = 0, n = 1, j = i + 1 \\ 0 & otherwise \end{cases} \tag{7}$$

g. Phone transition (PT1 and PT2). In this model, each phone is composed of fixed number of states. The CPD $P(PTx_t \mid Px_t, SPx_t, STx_t)$ is given by:

$$P(PTx_t = j \mid Px_t = a, SPx_t = b, STx_t = m) = \begin{cases} 1 & j = 1, m = 1, b = laststate(a) \\ 1 & j = 0, m = 1, b = \sim laststate(a) \\ 0 & otherwise \end{cases} \tag{8}$$

Where $laststate(a)$ denotes the last state of phone 'a'. Only when the last state of a phone is reached, and a state transition is allowed, the current phone can transit to a new phone unit.

h. Word transition (WT), which is determined by audio stream and visual stream together.

$$p(WT_t = j \mid W_t = a, PP1_t = b, PP2_t = c, PT1_t = m, PT2_t = n) = \begin{cases} 1 & j = 1, m = 1, n = 1, b = lastphone1(a), c = lastphone2(a) \\ 1 & j = 0 \ (m \neq 1 \text{ or } n \neq 1 \text{ or } b \neq lastphone1(a) \text{ or } c \neq lastphone2(a) \\ 0 & otherwise \end{cases} \tag{9}$$

The condition $b = lastphone1(a)$ means that b corresponds to the last phone of the word 'a' in the audio stream. Similarly, the condition $c = lastphone2(a)$ means that c corresponds to

the last phone of the word 'a' in the visual stream. Equation (9) means that when the phone units reach the last phone of the current word for both in audio stream and in visual stream respectively, and phone transitions for both two streams are allowed, the word transition occurs. Otherwise, word transition is not changed.

i. Word node (W), in initial frame, the word variable W starts out using a unigram distribution over words in the vocabulary. In the other frames, the word variable W_t depends on W_{t-1} and WT_t with CPD $P(W_t = j \mid W_{t-1} = i, WT_t = m)s$.

$$P(W_t = j \mid W_{t-1} = i, WT_t = m) = \begin{cases} bigram(i, j) & if \quad m = 1 \\ 1 & if \quad m = 0, \quad i = j \\ 0 & otherwise \end{cases} \tag{10}$$

where $bigram(i, j)$ means the transition probability from word i to word j.

2.4 Comparison of model

It can be known from the above models, main differences of several models are as follows.

1. The asynchrony of audio stream and visual stream: SS-MSHMM describes the asynchrony of the two streams at the HMM state level, and SA-MSHMM (be implemented as PHMM) looses the asynchrony to the phone boundary level, while MS-ADBN model and MM-ADBN model describe the asynchrony of both streams at word level.
2. The different recognition unit: MS-ADBN model is a word model whose recognition basic units are words. While MM-ADBN model is a phone model whose recognition basic units are phones, and can be used for large vocabulary audio visual speech recognition.

In the following experiments section, for the models proposed in this section, we will make some comparisons on the sake of different asynchrony description and different recognition basic unit.

3. Experiments and evaluation

In our work, The Graphical Models Toolkit (GMTK) (Bilmes & Zweig, 2002) has been used for the inference, learning and recognition of all the DBN models and Hidden Markov Model Toolkit (HTK) (Young, 1994) has been used for all HMMs. Speech recognition experiments have been done on a continuously spoken audio-visual digit database and an audio-visual large vocabulary continuous speech database respectively.

3.1 Audio visual database description

The digit continuous audio-visual database has been recorded with the scripts from the AURORA database 2.0 (Hirsch & Pearce 2000) which contains digit sequence from telephone dialing. Each sequence contains several digits from the digit set {zero, one, …, nine, oh}. 22 phone units are obtained by transcribing the digit set with the TIMIT dictionary. 100 clean audio-visual sentences have been selected as training set, and another 50 audio-visual sentences as testing set. White noise with signal to noise ratio (SNR) ranging from 0dB to 30dB has been added to obtain noisy speech.

The continuous audio-visual experiments database has been recorded with the scripts from the TIMIT database. 6 people's 600 sentences containing 1693 word units have been used in our experiments. Totally 76 phone units (including "silence" and short pause "sp") are obtained by transcribing the sentence scripts into phone sequences using the TIMIT dictionary. Since the database is relatively small for large vocabulary audio-visual speech recognition. To test performance of MM-ADBN model, we use the jackknife procedure, 600 sentences were split up in six equal parts, and six recognition experiments were carried out. In each recognition experiment, 500 sentences are used as training set, the remaining 100 sentences as testing set. Report test results are the average of the results of six experiments. While for MS-ADBN model, since it is word model, to avoid the case that some words in the testing sentence may not appear in the training set, all 600 sentences are used as training set and testing set. Noisy environments are also considered by adding white noise with SNRs ranging from 0dB to 30dB as testing set.

The above two databases are recorded with the same condition: with high-quality camera, clean speech environment, uniform background and lighting. The face of the speaker in the video sequence is high-quality frontal upright, and video is MPEG2-encoded at a resolution of 704×480, and at 25Hz.

3.2 Audio and visual feature extraction

Mel Filterbank Cepstrum Coefficients (MFCC) features are extracted by HTK with the frame rate of 100 frames/s. 13 MFCC features, energy, together with their delta and acceleration coefficient, resulting in a feature vector of 42 acoustic features (MFCC_E_D_A) has been extracted.

Visual feature extraction is given in Fig. 4, which starts with the detection and tracking of the speaker's face (Ravyse et al, 2006), Since the mouth is the most important speech organs, the contour of the lips is obtained through the Bayesian Tangent Shape Model (BTSM) (Zhou et al, 2003), for each image, 20 profile points include outer contour and inner contour of the mouth are obtained, which is given in Fig. 5. Based on these profile feature points, we extract a 20 dimensional geometrical feature vector: 5 vertical distance features and 5 horizontal distance features between outer contour feature points, 3 vertical distance features and 3 horizontal distance features between inner contour feature points, 4 angle features (α, β, θ and φ). Sketch map of the features are given in Fig. 5.

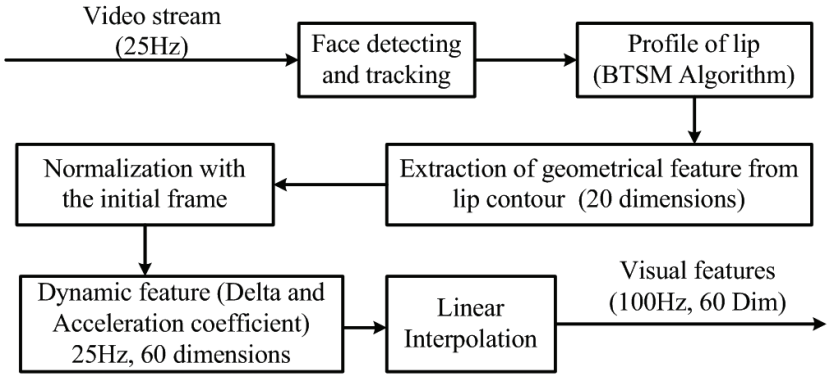
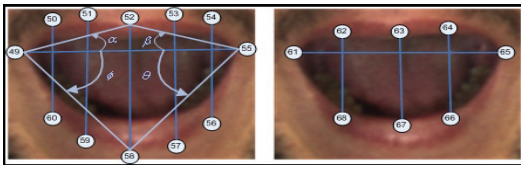


Fig. 4. Visual feature extraction



(a) Outer contour (b) Inner contour

Fig. 5. Profiles and geometrical features of lip

To eliminate the different speaker's affection, all visual features are normalized by subtracting the corresponding initial frame geometrical feature. In order to describe the dynamic visual feature, we extract the delta and acceleration coefficient of the basic visual feature. At the same time, the visual features are extracted at 25Hz, since the audio features are processed at 100Hz, and the visual features are linearly interpolated to make them occur at the same frame rate as the audio features. Totally, the 60 dimensions lip geometrical features are obtained.

3.3 Experiment setup and results

To evaluate the performance of model proposed in the paper, for the sake of comparison, experiments are also done under the same conditions on the conventional triphone HMM, two single stream DBN model (Lv et al. 2007): WP-DBN models with word-phone structure and WPS-DBN model with word-phone-state structure, At the same time, state synchrony multi-stream HMM (SS-MSHMM) and state asynchrony multi-stream HMM (SA-MSHMM) are used, and SA-MSHMM is implemented as product HMM, with four audio and four video HMM states, totally 16 composite states. For PHMM, at various SNRs (ranging from 0dB to 30dB), stream exponent of audio stream is varied from 0 to 1 in step of 0.05, and the value of the stream exponent that maximized the word accuracy is chosen.

In the experiments on the digit audio-visual database, for MS-ADBN model, both in audio stream and in visual stream, each of the 22 phones is associated with the observation feature, with the probability is modeled by 1 Gaussian. Together with the three-phone "silence" model and the one phone "short pause" model which actually ties its phone with the middle phone of the silence model, totally there are 50 Gaussians in the model for two streams. While for MM-ADBN model, in each stream, each of the 22 phones is composed of 4 states modeled by 1 Gaussian, together with the silence and short pause model, totally, there are 182 Gaussian. Since in the training sentences, each digit has about 60 samples, and each phone has about 100 occurrences. It can be seen that every model can be properly trained.

In the experiments on the continuous audio-visual database, for MS-ADBN model, in each stream, each of the 74 phones is associated with the observation feature, with the probability is modeled by 1 Gaussian. Together with the silence and short pause model, totally parameters of 154 Gaussians need be trained for both streams. While for MM-ADBN model, in each stream, each of the 74 phones is composed of 4 states modeled by 1 Gaussian, together with silence model and short pause model, totally, there are 598 Gaussians. As a consequence, comparing with MM-ADBN model, MS-ADBN model has relatively small training parameters. In the training set, since each word has about 4 samples, MS-ADBN can not be trained sufficiently, while each phone has about 600 occurrences. MM-ADBN can be properly trained.

Setup	0dB	5dB	10dB	15dB	20dB	30dB	Clean	0-30dB
WP-DBN (audio only)	42.94	66.10	71.75	77.97	81.36	96.61	97.74	72.79
HMM (audio only)	30.21	41.0	62.67	74.62	85.67	98.04	98.79	65.36
WPS-DBN (audio only)	19.6	28.7	46.41	64.71	81.7	96.08	97.04	56.2
WP-DBN, video only	66.67	66.67	66.67	66.67	66.67	66.67	66.67	66.67
HMM (video only)	64.2	64.2	64.2	64.2	64.2	64.2	64.2	64.20
WPS-DBN (video only)	66.06	66.06	66.06	66.06	66.06	66.06	66.06	66.06
SS-MSHMM (audio and visual feature)	42.73	54.24	67.88	77.27	85.15	91.21	92.73	69.75
SA-MSHMM (audio and visual feature)	44.63	55.31	69.23	77.89	86.92	94.36	95.72	71.39
MS-ADBN (audio and visual feature)	53.94	70.61	86.06	89.39	93.03	95.76	97.27	81.46
MM-ADBN (audio and visual feature)	33.64	43.03	60.61	73.03	81.52	89.39	94.55	63.54

Table 1. Word recognition rate for the digit audio-visual database (in %)

Setup	0dB	5dB	10dB	15dB	20dB	30dB	Clean
WP-DBN (audio only)	2.39	5.61	9.07	14.80	17.06	22.79	27.57
HMM (audio only)	0.72	1.07	3.46	14.32	27.21	44.87	49.76
WPS-DBN (audio only)	2.51	5.13	9.11	16.47	29.24	50.48	62.77
WP-DBN, video only	6.56	6.56	6.56	6.56	6.56	6.56	6.56
HMM (video only)	10.86	10.86	10.86	10.86	10.86	10.86	10.86
WPS-DBN (video only)	16.11	16.11	16.11	16.11	16.11	16.11	16.11
SA-MSHMM (audio and visual feature)	11.69	18.38	25.89	36.99	44.15	52.15	55.37
MS-ADBN (audio and visual feature)	11.32	12.79	13.18	15.64	17.89	24.10	29.43
MM-ADBN (audio and visual feature)	16.21	21.16	32.72	40.24	49.38	55.98	65.34

Table 2. Word recognition rate for the continuous audio-visual database (in %)

Word recognition rates for digit audio-visual database and continuous audio-visual database, using MS-ADBN model and MM-ADBN model, respectively, are given in Table 1 and Table 2. For the sake of comparison, word recognition rates obtained from HMM, SS-MSHMM, SA-MSHMM, WP-DBN model and WPS-DBN model are also given.

It can be notice from Table 1 and Table 2 that:

a. For audio-only speech recognition on digit audio-visual database, under clean or relatively clean conditions with SNRs as 20dB and 30dB, the speech recognition rates of WP-DBN model are lower than those of triphone HMM. But the recognition rates under 20dB show that WP-DBN is more robust to noisy environments. Additionally, for speech recognition with visual features on digit audio-visual database, WP-DBN model performs slightly better than triphone HMM. A possible reason is that the DBN model describes better the dynamic temporal evolution of the speech process. While WPS-DBN model has the worse performance than triphone HMM, a possible reason is that

WPS-DBN model uses single Gaussian model, triphone HMM uses Multi-Gaussian mixture model. For audio only or video only speech recognition on continuous audio-visual database, WPS-DBN model outperform than triphone HMM at various SNRs.

b. Because of integrating the visual features and audio features, multi-stream models have the better performance than corresponding single stream models. For digit audio-visual database, in the noisy environment with signal to noise ratios ranging from 0dB to 30dB, comparing with HMM, WP-DBN and WPS-DBN model, the average improvements of 6.03%, 8.67% and 7.34% are obtained in speech recognition rate from SA-MSHMM, MS-ADBN and MM-ADBN model respectively. As well as for continuous audio-visual database, in clean speech, the improvements of 5.61%, 7.81% and 0.42% respectively.
c. For digit audio-visual database, MS-ADBN model has the better performance than SS-MSHMM and SA-MSHMM. This trend becomes even more obvious with the increasing of noise. Since the SA-MSHMM forces audio stream and visual stream to be synchronized at the timing boundaries of phones, while the MS-ADBN model looses the asynchrony of both streams to word level, the recognition results show the evidence that the MS-ADBN model describes more reasonable audio visual asynchrony in speech. As well as for continuous audio-visual database, MM-ADBN model has the better performance than SA-MSHMM. At clean speech environment, MM-ADBN model has the improvement of 9.97% than SA-MSHMM in speech recognition rate.
d. It should be noticed that under all noise conditions for digit audio-visual database, the MM-ADBN model gets worse but acceptable recognition rates than the MS-ADBN model, while for continuous audio-visual database, MM-ADBN model outperform than MS-ADBN model at various SNRs. At clean speech environment, the speech recognition rate of MS-ADBN model is 35.91% higher than that of the MS-ADBN in speech recognition rate. These are in coincidence with the speech recognition results of the single stream WP-DBN model and WPS-DBN model in (Lv et al. 2007). Since MM-ADBN model and WPS-DBN model are all phone models and are appropriate for large vocabulary speech recognition. MS-ADBN model and WP-DBN model are all word models, which cannot be properly trained for large vocabulary database, and they are appropriate for small vocabulary speech recognition, since they can be properly trained.

4. Conclusions and future work

In this paper, two multi-stream asynchrony Dynamic Bayesian Network (DBN) model: MS-ADBN model and MM-ADBN model, are proposed for small vocabulary and large vocabulary audio-visual speech recognition, which loose the limitation of asynchrony of the audio stream and visual stream to word level. Essentially, MS-ADBN model is a word model with word-phone-observation topology structure, whose recognition basic units are word, while MM-ADBN model is phone model with word-phone-state-observation topology structure, whose recognition basic units are phones. Speech recognition experiments are done on digit audio-vidio database and continuous audio-vidio database, results show that: MS-ADBN model has the highest recognition rate on digit audio-visual database; while for continuous audio-visual database, in clean speech environment, comparing with SA-MSHMM and MS-ADBN model, the improvements of 35.91% and 9.97% are obtained for MM-ADBN model in speech recognition rate. In the future work, we

will continue to improve the MM-ADBN model, and build up the MM-ADBN model based word-triphone-state topology for large vocabulary audio-visual speech recognition.

5. Acknowledgment

This research has been conducted within the "Audio Visual Speech Recognition and Synthesis: Bimodal Approach" project funded in the framework of the Bilateral Scientific and Technological Collaboration between Flanders, Belgium (BILO4/CN/02A) and the Ministry of Science and Technology (MOST), China ([2004]487), and the fund of the National High Technology Research and Development Program of China (Grant No. 2007AA01Z324). We would like to thank Prof. H. Sahli and W. Verhelst (Vrije Universiteit Brussel, Electronics & Informatics Dept., Belgium) for their help and for providing some guidance. We would also like to thank Dr. Ilse Ravyse, Dr. Jiang Xiaoyue, Dr. Hou Yunshu and Sun Ali for providing some help for the audio-visual database and visual feature data.

6. References

Lippmann, R. P. (1997). *Speech recognition by machines and humans*. speech communication, vol. 22, pp. 1-15, 1997.

Hermansky, H. (1990). *Perceptual Linear Predictive (PLP) Analysis of speech*. Journal of Acoustical Society of America, vol. 87, No. 4, pp. 1738-1752, 1990.

Hermansky, H. & Morgan, N. (1994). *RASTA processing of speech*. IEEE transaction on speech and audio processing, vol. 2, no.4, pp. 587-589, 1994.

Potamianos, G. & Neti, C. et al (2003). *Recent advances in the automatic recognition of audiovisual speech*. Proc. IEEE, vol.91, no 9, pp.1306-1326, 2003.

Dupont, S. & Luettin, J. (2000). *Audio-visual speech modeling for continuous speech recognition*. IEEE Trans. on Multimedia, vol. 2, pp.141–151, 2000.

Gravier, G.; Potamianos, G. & Neti, C. (2002). *Asynchrony modeling for audio-visual speech recognition*. in Proc. Human Language Technology Conf., San Diego, CA, pp. 1–6, 2002.

Nefian, A.; Liang, L. & Pi, L. et al (2002). *Dynamic Bayesian Networks for audio-visual speech recognition*. in EURASIP Journal on Applied Signal Processing, vol. 11, pp.1-15, 2002.

Bilmes, J. & Zweig, G. (2002). *The Graphical Modelds Toolkit:An Open Source Software System For Speech And Time-Series Processing*. Proceedings of the IEEE International Conf. on Acoustic Speech and Signal Processing (ICASSP), vol. 4, pp.3916-3919, 2002.

Murphy, K. (2002). *Dynamic Bayesian networks: Representation, inference and learning*. Ph.D. dissertation, Dept. EECS, CS Division, Univ. California, Berkeley, 2002.

Zweig, G. (1998). *Speech recognition with dynamic Bayesian networks*. Ph.D. dissertation, Univ. California, Berkeley, 1998.

Bilmes, J & Zweig, G. et al (2001). *Discriminatively structured graphical models for speech recognition: JHU-WS-2001 final workshop report*. Johns Hopkins Univ., Baltimore, MD, Tech. Rep. CLSP, 2001.

Lv, G.Y.; Jiang, D.M. & H, Sahli. et al (2007). *a Novel DBN Model for Large Vocabulary Continuous Speech Recognition and Phone Segmentation*. International Conference on Artificial Intelligence and Pattern Recognition (AIPR-07), Orlando, USA, pp. 437-440, 2007.

Zhang, Y.M.; Diao, Q. & Huang, S. et al (2003). *DBN based multi-stream models for speech*. in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Hong Kong, China, vol. 1, pp. 836–839, 2003.

Gowdy, J.N.; Subramanya, A. & Bartels, C. et al (2004). *DBN based multi-stream models for audio-visual speech recognition*. In Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, vol. 1, pp. 993–996, 2004.

Bilmes, J. & Bartels, C. (2005). *Graphical Model Architectures for Speech Recognition*. IEEE Signal Processing Magazine, Vol. 22, no.5, pp.89–100, 2005.

Young, S.J.; Kershaw, D. & Odell, J. et al (1998). *The HTK Book*. http://htk.eng.cam.ac.uk/docs/docs.shtml.

Hirsch, H. G. & Pearce, D. (2000). *The aurora experimental framework for the performance evaluations of speech recognition systems under noisy conditions*. ICSA ITRW ASR2000, September, 2000.

Zhou, Y.; Gu, L. & Zhang, H.J. (2003). *Bayesian Tangent Shape Model: Estimating Shape and Pose Parameters via Bayesian Inference*. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003), Wisconsin, USA, vol. 1, pp. 109-116, 2003.

Ravyse, I. ; Jiang, D.M. & Jiang, X.Y. et al (2006). *DBN based Models for Audio-Visual Speech Analysis and Recognition*. PCM 2006, vol. 1, pp.19-30, 2006.

Speaker recognition/verification

17

Normalization and Transformation Techniques for Robust Speaker Recognition

Dalei Wu, Baojie Li and Hui Jiang
Department of Computer Science and Engineering, York University, Toronto, Ont., Canada

1. Introduction

Recognizing a person's identity by voice is one of intrinsic capabilities for human beings. Automatic speaker recognition (SR) is a computational task for computers to perform a similar task, i.e., to recognize human identity based on voice characteristics. By taking a voice signal as input, automatic speaker recognition systems extract distinctive information from the input, usually using signal processing techniques, and then recognize a speaker's identity based on the extracted information by comparing it with the knowledge previously learned at a training stage. The extracted distinctive information is encoded in a sequence of feature vectors, which is referred to as frame sequence. In terms of purposes of applications, SR tasks can be classified into two categories: speaker identification and speaker verification.

Speaker identification (SI) is an application to recognize a speaker's identity from a given group of enrolled speakers. If a speaker is assumed to be always in the enrolled speaker group, it is referred to as the closed set speaker identification; Otherwise, it is referred to as the open set speaker identification. On the other hand, speaker verification (SV) is an application to verify a speaker identity by simply making a binary decision, i.e., answering an identity question by either yes or no. SV is one of biometric authentication techniques, along with others, such as fingerprint (Jain et al., 2000) or iris authentication (Daugman, 2004).

In the past decades, a variety of techniques for modeling and decision-making have been proposed to speaker recognition and proved to work effectively to some extent. In this chapter, we shall not delve too much into the survey for these techniques, but rather focus on normalization and transformation techniques for robust speaker recognition. For a tutorial of the conventional modeling and recognizing techniques, the reader can refer to (Campbell, 1999; Reynolds, 2002; Bimbot et al., 2004). Here, we just make it explicit that among many techniques the most successful ones are Gaussian mixture model (GMM) and hidden Markov model (HMM). With GMM/HMM, high performance can be achieved in sound working conditions, such as in a quiet environment, and for broadband speech. However, these techniques run into problems in realistic applications, since many realistic applications can not always satisfy the requirements of clean and quiet environments. Instead, the working environments are more adverse, noisy and sometimes in narrow-band width, for instance, telephony speech. Most SR systems degrade their performance substantially in adverse conditions. To deal with the difficulties, robust speaker recognition is such a topic for study.

As robust speech recognition does, robust speaker recognition is concerned with improving performance of speaker recognition systems in adverse or noisy (additive and convolutional noise) conditions and making systems more robust to a variety of mismatch conditions. The essential problem for robust speaker recognition is the existence of mismatch between training and test stages. As the most prominent SR systems adopt statistical methods as their main modeling technique, such as GMM and HMM, these systems confront the common issues held by all the statistical modeling methods, i.e., vulnerable to any mismatch between the training and test stages. In noisy environments, the mismatch inevitably becomes larger than in clean conditions, due to a larger range of data variance caused by the interference of ambient noise. Hence, how to deal with a variety of types of mismatch becomes a crucial issue for robust speaker recognition.

Much research has been devoted for solving the problem of mismatch in last decades. To summarize these techniques for robust speaker recognition is the main purpose of this chapter. To the authors' best knowledge, so far there is no any article in the literature to survey this subject, although some more general tutorials for speaker recognition exist (Campbell, 1999; Reynolds, 2002; Bimbot et al., 2004). Different from these tutorials, we shall only focus on reviewing the techniques that aim to reduce or at least alleviate the mismatch for robust speaker recognition in terms of normalization and transformation at two levels, i.e., normalization/transformation at the score level and normalization/transformation at the feature level. In order to avoid confusion and also be easier to discuss directions for future work in later sections, we shall explicitly explain the terms of normalization and transformation we used above. Consistent to its general meaning, normalization, we mean here, is a sort of mapping functions, which map from one domain to another. The mapped images in the new domain often hold a property of zero mean and unit variance in a general sense. By transformation, we refer to more general mapping functions which do not possess the property of zero mean and unit variance. Although these two terms are distinctive in subtle meanings, they are sometimes used by different authors, depending on their preferences. In this chapter, we may use them exchangeably without confusion. Just by using these techniques, speaker recognition systems become more robust to realistic environments.

Normalization at the score level is one of noise reduction methods, which normalizes log-likelihood scores at the decision stage. A log-likelihood score, for short, score, is a logarithmic probability for a given input frame sequence generated based on a statistical model. Since the calculated log-likelihood scores depend on test environments, the purpose of normalization aims at reducing this mismatch between a training and test set by adapting the distribution of scores to test environments, for instance, by shifting the means and changing the range of variance of the score distribution. The normalization techniques at the score level are mostly often used in speaker verification, though they can be also applied to speaker identification, because they are extremely powerful to reduce the mismatch between the claimant speaker and its impostors. Thus, in our introduction to normalization techniques at the score levels, we shall use some terminologies from speaker verification, such as claimant speaker/model, or impostor (world) speaker/model, without explicitly emphasizing these techniques being applied to speaker identification as well. The reader who is not familiar with these terminologies can refer to (Wu, 2008) for more details.

The techniques for score normalization basically includes Z-norm (Li et al., 1988; Rosenberg et al., 1996), WMAP (Fredouille et al., 1999), T-norm (Auckenthaler et al., 2000), and D-norm

(Ben et al. 2002). In retrospect, the first normalization method, i.e. Z-norm, dates back to Li et al. (1988) who used it for speaker verification. With ***Z-norm***, a.k.a. zero normalization, Li removed most of the variability across segments by making the log-likelihood scores relative to the mean and standard deviation of the distribution of impostor scores. In (Rosenberg et al. 1996), a score was normalized by directly subtracting from it the score from the impostor model, which incorporated the mean and variance of the impostor model. Strictly speaking, the method adopted by Rosenberg et al. is different from that used by Li et al., in the sense that the normalization did not directly act on the mean and variance of the impostor model, but instead, on the score calculated based on the mean and variance of the impostor model. Therefore, to some extent, Resenberg's method can be regarded as a variant of Z-norm. ***WMAP*** is a score normalization method based on world model and a posterior probability (WMAP), which is in fact a two-step method. At the first step, the posterior score is normalized using a world model (see Eq. (4)), representing the population in general (see Wu, 2008). At the second step, the score is converted into posterior probability by using Bayesian rule (see Eq. (5)). ***T-norm***, test normalization, is a method based on mean and variance of the score distribution estimated from a test set. It is similar to the Z-norm, except that the mean and variance of the impostor model are estimated on a test set. ***D-Norm*** is one of the score normalization techniques based on the use of Kullback-Leibler (KL) distance. In this method, KL distance between a claimed model and a world model is first estimated by Monte Carlo simulation, which has been experimentally found to have a strong correspondence with impostor scores. Hence, the final scores are normalized by the estimated KL distance.

The second class of normalization and/or transformation techniques is applied at the feature level. In contrast to normalization/transformation at the score level, which uses normalization techniques at a later stage, normalization/transformation in this class is applied at a very early stage, i.e., at the feature level. The typical methods are composed of cepstral mean subtraction (CMS), spectrum subtraction, RASTA (Hermansky et al., 1991), H-norm (Reynolds, 1996), C-norm (Reynolds, 2003), linear discriminant analysis (LDA) and nonlinear discriminant analysis (NLDA) (Wu, 2008). ***Cepstral mean subtraction*** and ***spectrum subtraction*** are very similar, as they perform normalization with a similar method, i.e., subtracting from each single feature frame a global mean vector, which is estimated across an overall sentence. However, these two normalizations methods are differently applied to the cepstral (logarithm spectral) or spectral domain, which their naming is owing to. ***RASTA*** processing transforms the logarithmic spectra by applying a particular set of band-pass filters with a sharp spectral zero at the zero frequency to each frequency channel, with a purpose of suppressing the constant or slowly-varying components, which reflect the effect of convolutional noises in communication channels. ***H-norm*** is also called handset normalization method, which was firstly applied by Reynolds et al (1996) to alleviate the negative effect on speech signals due to using different handset microphones. The idea of H-norm is that it uses frame energy dependent CMS for each frame normalization, so it is in fact a piece-wise linear filtering method. ***C-norm***, is a technique designed for cellular telephones, which transforms a channel dependent frame vector into a channel independent frame vector (see Eq. (20)). Thus, the final recognition is conducted in a channel independent feature space. These two methods are using linear or nonlinear transformations to project an original feature space to another feature space, in order to suppress the effects of noisy channels. ***Linear discriminant analysis (LDA)*** and ***nonlinear discriminant analysis (NLDA)***

are applied to this case. LDA seeks the directions to maximize the ratio of between-class covariance to within-class covariance by linear algebra, whereas NLDA seeks the directions in a nonlinear manner implemented by neural network. The details for these normalization and transformations methods will be presented in Section 2 and 3.
The remainder of this chapter is organized as follows: in Section 2, the score normalization techniques are firstly summarized in details, following the order presented in the overview above. In Section 3, the description to the normalization and transformation at the feature level is given. In Section 4, some recent efforts are presented. The discussions and limitations are commented in Section 5. In Section 6, final remarks concerning possible extensions and future works are given. Finally, this chapter is concluded with our conclusions in Section 7.

2. Normalization techniques at the score level

2.1 Z-Norm

Zero normalization, Z-norm in short, is one of score normalization methods applied for speaker verification at the score level, which was firstly proposed by Li et al. (1988). In Li's proposal, variations in a given utterance can be removed by making the log-likelihood scores relative to the mean and variance of the distribution of the impostor scores. Concretely speaking, let $L(\mathbf{x}_i \mid S)$ be a log-likelihood score for a given speaker model S and a given feature frame $\mathbf{x}_i$, where an overall utterance is denoted by $\mathbf{X}=\{\mathbf{x}_i\}$, $i \in [1,N]$. Here $L(\mathbf{x}_i \mid S)$ is also called raw score. We shall then refer to $L_{norm}(\mathbf{x}_i \mid S)$ as the normalized log-likelihood score. Based on the notations, we have the following equation,

$$L(\mathbf{x} \mid S) = \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{x}_i \mid S), \tag{1}$$

and the normalized score,

$$L_{norm}(\mathbf{x} \mid S) = \frac{L(\mathbf{x} \mid S) - \mu_I}{\sigma_I}, \tag{2}$$

where μ_I and σ_I are the mean and standard deviation of the distribution of the impostor scores, which are calculated based on the impostor model S_I.
The original Z-norm was later improved by a variant method, which was proposed by Rosenberg et al. (1996) to normalize a raw log-likelihood score relative to the score obtained from an impostor model, i.e.,

$$L_{norm}(\mathbf{x} \mid S) = L(\mathbf{x} \mid S) - L(\mathbf{x} \mid S_I) \tag{3}$$

where S is the claimed speaker and S_I represents the impostors of speaker S. In fact, this variant version has become more widely used than the first version of Z-norm. For instance, the next presented normalization method – WMAP adopts it as the first step of normalization.

2.2 WMAP

WMAP is referred to as score normalization based on world model and a posterior probability. WMAP consists of two stages. It uses a posterior probability at the second stage

to substitute for the score normalized by a world model at the first stage. This procedure can be described as follows:
Step1: normalization using the world model. With the log-likelihood score $L(\mathbf{X}|S)$ for the speaker S and the utterance **X**, and the log-likelihood score $L(\mathbf{X}|\overline{S})$ for the world model of speaker S and utterance **X**, the normalized score is then given by

$$R_s = L(\mathbf{X}|S) - L(\mathbf{X}|\overline{S}). \tag{4}$$

This step is called score normalization by world model.
Step 2: normalization as a posterior probability. At the second step, the score R_s is further normalized as a posterior probability using the Bayes' rule, i.e.,

$$Score_{norm} = \frac{P(R_s|S)P(S)}{P(R_s|S)P(S) + P(R_s|\overline{S})P(\overline{S})}, \tag{5}$$

where $P(S)$ and $P(\overline{S})$ are prior probabilities for target speaker and impostor, $P(R_s|S)$ and $P(R_s|\overline{S})$ are the probability for the ratio R_s generated by the speaker model S and impostor model $\overline{S}$ respectively, which can be estimated based on a development set.
From these formulae, we can see the most advantage of WMAP, compared with Z-norm, is its two stage scheme for score normalization. The first step for normalization focuses on the difference between target and impostor scores. This difference may vary in a certain range. Thus, in the second normalization, the score difference is converted into the range of [0, 1], a posterior probability, which renders a more stable score.

2.3 T-norm

T-norm is also called as test-norm because this method is based on the estimation on the test set. Essentially, T-norm can be regarded as a further improved version of Z-norm, as the normalization formula is very similar to that of Z-norm, at least in formality. That is, a normalized score is obtained by

$$L_{norm}(\mathbf{x}|S) = \frac{L(\mathbf{x}|S) - \mu_{I\_test}}{\sigma_{I\_test}}, \tag{6}$$

where μ_{I_test} and σ_{I_test} are the mean and standard deviation of the distribution of the impostor scores estimated on a test set. In contrast, for Z-norm, the corresponding μ_I and σ_I are estimated on the training set (see Eq. (2)).
As there is always mismatch between a training and test set, the mean and standard deviation estimated on a test set should be more accurate than those estimated on a training set and therefore it naturally results in that performance of T-norm is superior to that of Z-norm. This is one biggest advantage of T-norm. However, one of the major drawbacks for T-norm is that it may require more test data in order to attain sufficiently good estimation, which is sometime impossible and impractical.

2.3 D-norm

D-norm is a score normalization based on Kullback-Leibler (KL) distance. In Ben et al. (2002), D-norm was proposed to use KL distance between a claimed speaker's and the

impostor's models as a normalization factor, because it was experimentally found that the KL distance has a strong correspondence with the impostor scores. In more details, let us firstly define the KL distance. For a probability density function $p(\mathbf{X}|S)$ of speaker S and an utterance $\mathbf{X}$, and a probability density function $p(\mathbf{X}|W)$ of the speaker S's world model W and an utterance $\mathbf{X}$, the KL distance of S to W, KL_w is denoted by

$$KL_S = E_p\left[\log(\frac{p_S}{p_W})\right] = \int p_S \log(\frac{p_S}{p_W})dx\,, \tag{7}$$

where $E_p[\bullet]$ is an expectation under the law p.
Similarly, the KL distance of W to S, KL_w is defined as an symmetric distance $KL2$.

$$KL_w = E_w\left[\log(\frac{p_W}{p_S})\right] = \int p_W \log(\frac{p_W}{p_S})dx\,. \tag{8}$$

Hence, the KL distance between S and W is defined by

$$KL2 = KL_S + KL_W\,. \tag{9}$$

Direct computation of KL distance according to Eqs.(7)-(9) is not possible for most complex statistical distributions of p_S and p_W, such as GMM or HMM. Instead, the Monte Carlo simulation method is normally employed.
The essential idea of the Monte-Carlo simulation is to randomly generate some synthetic data for both claimed and impostor models. Let us denote a synthetic data from a speaker S by $\tilde{\mathbf{y}}_n^S$ and a synthetic data from an impostor model W by $\tilde{\mathbf{y}}_n^W$. And also suppose a Gaussian mixture model (GMM) is used to model speaker S and the world model W, i.e.,

$$p(\mathbf{y}) = \sum_{i=1}^{m} w_i \mathbb{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\,, \tag{10}$$

where $\mathbb{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is a normal distribution with mean $\boldsymbol{\mu}_i$ and covariance Σ_i, and m is the total number of mixtures.
Then according to the Monte-Carlo method, a synthetic data $\tilde{\mathbf{y}}_n$ is generated with transforming a random vector, $\tilde{\mathbf{y}}'_n$, which is generated from a standard normal distribution with zero-mean and unit-variance. The transformation is done with the specific mean and variance, which are parameters related to one Gaussian, randomly selected among all the mixtures in a given GMM

$$\tilde{\mathbf{y}}_n = \boldsymbol{\Sigma}_k^{-1}(\mathbf{y}'_n + \boldsymbol{\mu}_k)\,. \tag{11}$$

As the most important assumption of D-norm, the KL distance is assumed to have a strong correspondence with the impostor score, which was experimentally supported in Ben et al. (2002), i.e.,

$$KL2 = -\alpha \cdot L(\mathbf{X}|\bar{S})\,, \tag{12}$$

where α=2.5, used by Ben et al. (2002).

Finally, at the last step, the normalized score is obtained by the equation,

$$L_{Norm}(\mathbf{X} \mid S) = \frac{L(\mathbf{X} \mid S)}{KL2} . \tag{13}$$

This is the overall procedure for D-norm.

3. Transformation techniques at the feature level

3.1 Cepstral (spectral) mean subtraction

Cepstral mean subtraction (CMS) is one of the most widely used normalization methods at the feature level and also a basis for other normalization methods (Reynolds, 2002). Given an utterance $\mathbf{X}=\{\mathbf{x}_i\}$, $i \in [1,N]$with a feature frame $\mathbf{x}_i$, the mean vector $\mathbf{m}$ of all the frames, for the given utterance, is calculated as follows:

$$\mathbf{m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i . \tag{14}$$

The normalized feature $\hat{\mathbf{x}}_t$ with CMS is then expressed by

$$\hat{\mathbf{x}}_t = \mathbf{x}_i - \mathbf{m} . \tag{15}$$

The idea of CMS is simply based on the assumption that noise level is consistently stable across a given sentence, so that by subtracting the mean vector from each feature frame, the background and channel noise could be possibly removed. However, it should be noted that speaking characteristics are most likely to be removed as well by this subtraction, as they are characterized by an individual speaker's speaking manner and should therefore also be consistently stable across a sentence at least.

Another normalization method at the feature level, which is very similar to CMS, is spectral mean subtraction (SMS). While CMS does mean subtraction in the cepstral domain, SMS instead conducts subtraction in the spectral domain. Due to their extremely similarity in methods, their normalization effects share the same pros and cons.

3.2 RASTA processing

RASTA processing transforms the logarithmic spectra by applying a particular set of band-pass filters with a sharp spectral zero at the zero frequency to each frequency channel, with a purpose of suppressing the constant or slowly-varying components, which reflect the effect of convolutional factors in communication channels. As is known (Hermansky et al. 1991), linear distortions, as caused by telecommunication channels or different microphones, appear as an additive constant in the log spectrum. So with band-pass filters in the log domain, the effects of additive or channel noise could be substantially alleviated. This is the essential idea for RASTA processing. Concretely speaking, the RASTA processing is carried out on the logarithmic bark-scale spectral domain by Hermansky et al (1991). So it can be considered as an additional step, inserted between the first (logarithm spectrum conversion) and the second step (equal loudness transform) in the common steps of the conventional perceptual linear prediction features (PLP) (Hermansky, 1990). After this additional step is taken, the other conventional steps from PLP, e.g. equal loudness transform, inverse logarithmic spectra, etc. are accordingly conducted. For this additional step, in a certain

frequency channel (a set of frequency channels is divided in order to extract features by a set of filters at the stage of feature extraction in speech processing, more details can refer to Wu, 2008; Young et al. 2002), a band-pass filtering is used as RASTA processing, through an IIR filter with the transfer function

$$H(z) = 0.1 \times \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{z^{-4} \times (1 - 0.98z^{-1})}. \tag{16}$$

The low cut-off frequency of the filter determines the fastest spectral changes which are ignored by RASTA processing, whereas the high cut-off frequency then determines the fastest spectral changes which are preserved in a channel.

3.3 H-norm

H-norm is referred to as handset normalization, which is a technique especially designed for speaker normalization over various handsets. Essentially, it can be considered as a variant of CMS, because H-norm does energy dependent CMS for different energy "channels". Concretely speaking, for an input frame stream $\{\mathbf{x}_i\}$ and its corresponding frame energies $\{e_i\}$, the energies are divided into L levels $\{E_l\}$, $l \in [1,L]$, on each of which a mean vector $\mathbf{m}_l$ is calculated according to the equation,

$$\mathbf{m}_l = \frac{1}{T_l} \sum_{E_l \le \mathbf{x}_t < E_{l+1}} \mathbf{x}_t \,, \tag{17}$$

whereis the number of frames whose energy levels are in $[E_l, E_{l+1}]$. Then for H-norm, a frame $\mathbf{x}_i$ is normalized by energy dependent CMS, i.e.

$$\mathbf{x}_i^{norm} = \mathbf{x}_i - \mathbf{m}_l, \; iff \; e_i \in [E_l, E_{l+1}]. \tag{18}$$

So the H-norm is to some extent more like a piecewise CMS, using different CMS at different energy levels. This renders H-norm to be probably more subtle and therefore more accurate than the uniform CMS scheme.

3.4 C-norm

C-norm is referred to as cellular normalization which was proposed by Reynolds (2003) for compensation of channel effects of cellular phones. However, C-norm is also called a method of feature mapping, because C-norm is based on a mapping function from a channel dependent feature space into a channel independent feature space. The final recognition procedure is done on the mapped, channel independent feature space. Following the symbols, which we used above, $\mathbf{x}_t$ is denoted as a frame at time t in a channel dependent (CD) feature space, anda frame at time t in a channel independent (CI) feature space. The GMM modeling for the channel dependent feature space is denoted G^{CD} as and the GMM for the channel independent feature space is denoted as G^{CI}. The Gaussian mixture to which a frame $\mathbf{x}_t$ belongs is chosen according to the maximum likelihood criterion, i.e.

$$i = \arg\max_j \left\{ \omega_j^{CD} \cdot p_j^{CD}(\mathbf{x}_t \mid \boldsymbol{\mu}_j^{CD}, \boldsymbol{\sigma}_j^{CD}) \right\}, \tag{19}$$

where a Gaussian mixture is defined by its weight, mean and standard deviation $\{\omega_j^{CD}, \boldsymbol{\mu}_j^{CD}, \boldsymbol{\sigma}_j^{CD}\}$. Thus, by a transformation $f(\bullet)$, a CI frame feature $\mathbf{y}_t$ is mapped from $\mathbf{x}_t$ according to

$$\mathbf{y}_t = f(\mathbf{x}_t) = (\mathbf{x}_t - \boldsymbol{\mu}_i^{CD}) \frac{\boldsymbol{\sigma}_i^{CI}}{\boldsymbol{\sigma}_i^{CD}} + \boldsymbol{\mu}_i^{CI} \ , \tag{20}$$

where i is a Gaussian mixture to which $\mathbf{x}_t$ belongs and is determined in terms of Eq. (19). After the transformation, the final recognition is conducted on the CI feature space, which is expected with the advantages of channel compensation.

3.5 Principal component analysis

Principal component analysis (PCA) is a canonical method to find some of the largest variance directions of a given set of data samples. If the vectors pointed by these directions are used as a set of bases for a new feature space, then an original feature space can be transformed into the new feature space. That is, for any d-dimensional feature frame $\mathbf{x}_i$ in the original feature space X, we have a transformation obtained by PCA, such that the original feature space X is transformed into a new one Y,

$$W : X \to Y \ ,$$

and $\mathbf{x}_t$ is transformed into $\mathbf{y}_t$, i.e.

$$W : \mathbf{x}_t \to \mathbf{y}_t \ . \tag{21}$$

The transformation matrix W can be sought by diagonalization of the covariance matrix $\boldsymbol{C}$ of the given data set X. If we set $\mathbf{X}=[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, where n is the size of the given data set, then the covariance matrix C is defined as

$$\mathbf{C} = \frac{1}{n} \mathbf{X}'\mathbf{X} \ . \tag{22}$$

By diagonalizing the covariance matrix C, we have

$$\mathbf{C}\mathbf{u} = \lambda \mathbf{u} \ , \tag{23}$$

where $\mathbf{u's}$ are principal vectors and λ's are the variances (or principal values) on the basis of $\mathbf{u's}$. Thus, by sorting the principal vectors according their principal values, we have the transformation $\boldsymbol{W}$ in such a form,

$$\mathbf{W} = [\mathbf{u}_1'; \ldots; \mathbf{u}_d'] \ .$$

So the transformed feature $\mathbf{y}_t$ is

$$\mathbf{y}_t = \mathbf{W}\mathbf{x}_t \ . \tag{24}$$

This is the traditional PCA, which is implemented with linear algebra. However, there is another variant of PCA, which is implemented with one type of neural networks – multi-layer perceptron (MLP). MLP is well known to be able to approximate any continuous linear and nonlinear function. Therefore, it has a wide range of applications in feature transformation. Let us first present how an MLP network is used to realize the functionality

of PCA in this section. And in following sections, we shall go back to discuss how an MLP is used as a nonlinear discriminant projector.

To remind the reader the fundamental principles of MLP, we summarize the most basic aspects of MLP. MLP is one of neural networks, which is composed of an input layer, where inputs are fed into the neural network, an output layer, where the transformed sequences are outputted, as well as several hidden layers between the input and output layer. A layer is called as a hidden layer, because it is between the input and output layer, so that it is invisible to the outside of the neural network. A typical example of MLPs, has an input layer, a hidden layer with a pretty large number of units, which are always referred to as hidden units, and an output layer, as illustrated as in Fig. 1. (a). The training of MLP, using back propagation (BP) algorithm, is well known as discriminative training, where the target (or reference) classes are fed into the output layer as the supervisors for training. Therefore, the training of MLP is definitely a supervised learning process. The standard target classes are identities for the given training sample $\mathbf{x}_t$. The overall training process resembles a recognition procedure with class identity labeling. It is beyond this chapter to describe further details regarding the theory of MLP. Readers can refer to Duda et al. (2001) and Wu (2008) for more details.

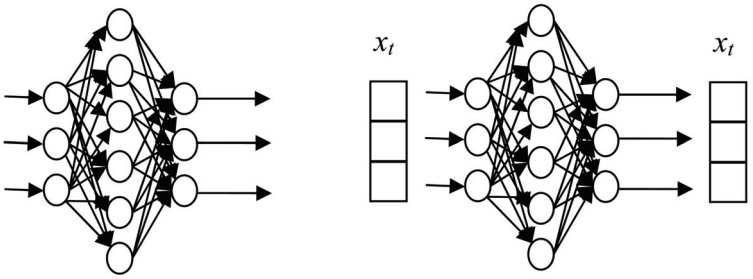


Fig. 1. (a) A typical fully-connected MLP with a hidden layer. (b) An MLP used for implementation of PCA.

In the standard MLP training, class identity tags are employed as supervisors. However, for an MLP to implement PCA, for a given data sample (frame) $\mathbf{x}_t$, instead of using $\mathbf{x}_t$'s class identity, $\mathbf{x}_t$, by itself, is used as supervisor in the procedure. This process is therefore named as self projection (see Fig. 1 (b)). If the number of the hidden layer of the MLP is less than the dimension of the features, then this method has an effect of dimension reduction, which is very similar to PCA.

3.6 Linear discriminant analysis

Linear discriminant analysis (LDA) can also be used as a method to transform a raw feature to another one in a more discriminative feature space. LDA is an optimal linear transformation which maximizes the ratio of the between-class covariance to the within-class covariance. Through this projection, some of the variation due to nonessential characteristics for class identities may be reduced, while class specific properties remain. This therefore enhances class discrimination.

The most important points for LDA are how to define within-class, between-class covariance and the optimization of their ratio. Suppose there are K classes, $\{C_i: i \in [1,K]\}$, in the d-dimensional space

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n],$$

the m-dimensional projected space

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n],$$

and there is an optimal linear transformation $\mathbf{W}$, such that $\mathbf{Y}=\mathbf{W}^T\mathbf{X}$.
Denote by $\tilde{\mathbf{m}}_k$ the mean vector in Space $\mathbf{Y}$ for class C_k and $\mathbf{m}_k$ the mean vector in Space $\mathbf{X}$ for class C_k. As such, the within-class covariance $\tilde{\mathbf{S}}_w$ of Space Y can be rewritten as

$$\tilde{\mathbf{S}}_w = \frac{1}{n}\sum_{i=1}^{n}\sum_{\mathbf{y}_i \in C_k} (\mathbf{y}_i - \tilde{\mathbf{m}}_k)(\mathbf{y}_i - \tilde{\mathbf{m}}_k)^T = \mathbf{W}^T\mathbf{S}_w\mathbf{W}, \tag{25}$$

where $\mathbf{S}_w$ is the within-class covariance of Space $\mathbf{X}$.
The between-class covariance $\tilde{\mathbf{S}}_b$ of Space $\mathbf{Y}$ can be expressed as

$$\tilde{\mathbf{S}}_b = \sum_{k\in[1,K]} (\tilde{\mathbf{m}}_k - \tilde{\mathbf{m}})(\tilde{\mathbf{m}}_k - \tilde{\mathbf{m}})^T = \mathbf{W}^T\mathbf{S}_b\mathbf{W}, \tag{26}$$

where $\mathbf{S}_b$ is the between-class covariance of Space $\mathbf{X}$.
So the optimization objective function is the ratio between the between-class and within-class covariance, i.e.

$$\lambda = \frac{\left|\mathbf{W}^T\mathbf{S}_b\mathbf{W}\right|}{\left|\mathbf{W}^T\mathbf{S}_w\mathbf{W}\right|}. \tag{27}$$

Solving this optimization problem (see Wu, 2008 for more details), we can have

$$\mathbf{S}_b\mathbf{w}_i = \lambda_i\mathbf{S}_w\mathbf{w}_i, \tag{28}$$

where $\mathbf{w}_i$ and λ_i are the i-th eigenvector and eigenvalue, respectively. If the number of eigenvectors selected for the transformation is less than m, then LDA in fact reduces the dimensionality of a mapped space. This is the most often case for the application of LDA.

3.7 Nonlinear discriminant analysis

Besides linear transforms such as PCA and LDA, nonlinear transforms can also be employed as a method of transformations at the feature level. This type of methods is named as nonlinear discriminant analysis (NLDA). There are two folds of meanings for the essence of the NLDA. First, it is one of discriminant algorithms. The goal of the NLDA is quite similar to that held by the LDA. They are both aiming at maximizing the between-class covariance and simultaneously minimizing the within-class covariance. Second, the maximization procedure for the NLDA is not carried out in a linear way, but in a nonlinear way. These two properties reflect the most important aspects of the NLDA.
On the other hand, the NLDA is sort of an extension to the LDA. A nonlinear function degenerates to a linear function, when a certain condition is specified. A set of nonlinear functions can be regarded as a super set that contains a set of linear functions. Thus, the NLDA is an extension to the LDA. Normally, NLDA is implemented by neural networks

such as MLP, whereas LDA is done by manipulation of linear algebra. Their implementation methods are substantially different. However, the essence of these two methods is similar, as described above. In fact, LDA can also be done by a linear MLP, viz. an MLP with only an input and output layer, but without any hidden layer. A simple reason to deduce this is that there is no any nonlinear operation in the linear MLP, therefore the transform solution of the linear MLP has a global optimum, which is similar to that obtained by linear algebra. The details for comparison of the LDA and the linear MLP can refer to Wu (2008).

A nonlinear function has a stronger capacity than a linear one to change the behaviors of a raw feature space. Therefore, NLDA transformations at the feature level are more powerful to enhance robust features by reducing the noisy parts in raw features. This is the essential idea for the application of NLDA, also including LDA, to robust speaker recognition.

MLP is one of the prevalent tools to realize NLDA. MLP is widely known as universal approximator. A hidden layer MLP with linear outputs can uniformly approximate any continuous function on a compact input domain to arbitrary accuracy provided the network has a sufficiently large number of hidden units (Bishop, 2006). Thus, we shall use MLP as the main implementer for the NLDA.

In contrast to being applied for a function approximator or a discriminative classifier, an MLP has to be adapted to a feature transformer, when it serves as an NLDA for feature transformation. In this case, the projected features can be output from any layer of an MLP. A hidden unit often employs a sigmoid function, which nonlinearly warps high values to units, and low values to zeros, or other similar shaped nonlinear functions as "gate" functions. If the features are output after these "gate" functions, the obtained features would possess a sharp peak in their distribution, which results in non-Gaussianlization in the features newly transformed. This could correspondingly give rise to poor performance when statistical models are employed at the modeling stage, such as GMM for speaker recognition. Therefore, we particularly generate the transformed features by outputting them from the positions before the "gate" functions of the hidden units in a given network (see Fig. 2.a, b).

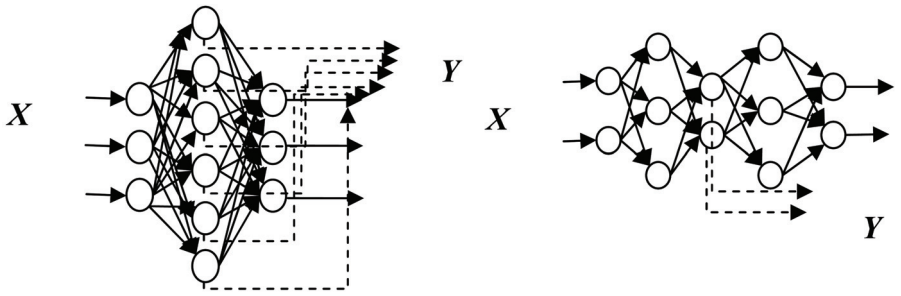


Fig. 2. Depiction of MLP used as feature transformer. (a) A hidden layer MLP, features output before the hidden units. (b) A three hidden layer MLP, feature output from a compact layer, before the hidden units.

Since it is better to keep the projected features with the same dimensions as those of the raw features for both efficiency and easy evaluation, instead of using the structured MLP in Fig. 2.a, we often adopt a specially-structured MLP with a small number of hidden units in one of the hidden layers, which we shall refer to it as a "compact" layer, as depicted in Fig. 2.b. With the special MLP with a compact layer, dimension reduction can be easily carried out,

as often do LDA and PCA. It thus provides the full capability to evaluate transformed and raw features with the same dimensionality. This is the basic scheme for an MLP to be employed as feature transformer at the feature level.

We have known that new features are output from a certain hidden layer. However, we do not know how to train a MLP yet. The MLP is trained on a class set. How to construct such a set for training is very crucial for an MLP to work efficiently. The classes selected in the set are the sources of knowledge for an MLP to learn discriminatively. So they are directly related to the recognition purposes of applications. For instance, in speech recognition, the monophone class is often used for MLP training. For speaker recognition, the training class set is naturally composed of speaker identities.

However, NLDA is not straightforward to be applied to robust speaker recognition, although we have known that speaker identities are used as target classes for MLP training. Because, compared with speech recognition, the number of speakers is substantially larger in speaker recognition, while such size for speech recognition is roughly about 30 in terms of phonemes. Therefore, it is impractical or even impossible to use thousands of speakers as the training classes for an MLP. Instead, we use a subset of speakers, which are most representative of the overall population, and therefore referred to as speaker basis, for MLP training. It is said to be "speaker basis selection" concerning how to select the optimal basis speakers (Wu et al., 2005a, 2005b; Morris et al., 2005).

The basic idea for speaker basis selection is based on an approximated Kullback-Leibler (KL) distance between any two speakers, say, S_i and S_k. By the definition of KL distance (Eq.(7)), the distance between S_i and S_i is written as the sum of two KL distance $KL(S_i \| S_k)$ and its reverse $KL(S_k \| S_i)$.

$$D(S_i, S_k) = KL(S_i \| S_k) + KL(S_k \| S_i) = \int \left(p(\mathbf{x} \mid S_i) - p(\mathbf{x} \mid S_k)\right) \log \frac{p(\mathbf{x} \mid S_i)}{p(\mathbf{x} \mid S_k)} d\mathbf{x}, \tag{29}$$

where represents an utterance. This cannot be evaluated in closed form when $p(\mathbf{x} \mid S_i)$ is modeled by a GMM. However, provided $P(S_i) = P(S_k)$, Eq.(29) can be simplified as the expectation of $K(S_i, S_k, \mathbf{k})$.

$$\begin{aligned} D(S_i, S_k) &\propto \int_{\mathbf{x}} p(\mathbf{x}) \left(p(S_i \mid \mathbf{x}) - P(S_k \mid \mathbf{x})\right) \log \frac{p(S_i \mid \mathbf{x})}{p(S_k \mid \mathbf{x})} d\mathbf{x} \\ &= \int_{\mathbf{x}} p(\mathbf{x}) K(S_i, S_k, \mathbf{x}) d\mathbf{x} = E\left[K(S_i, S_k, \mathbf{x})\right] \end{aligned}, \tag{30}$$

where $K(S_i, S_k, \mathbf{k})$ equals to

$$K(S_i, S_k, \mathbf{x}) = \left(p(S_i \mid \mathbf{x}) - P(S_k \mid \mathbf{x})\right) \log \frac{p(S_i \mid \mathbf{x})}{p(S_k \mid \mathbf{x})}. \tag{31}$$

Based on a development set *Dev*, the expectation of $K(S_i, S_k, \mathbf{k})$ can be approximated by the data samples on the set *Dev*, i.e.,

$$D(S_i, S_k) \cong \sum_{\mathbf{x} \in Dev} K(S_i, S_k, \mathbf{x}). \tag{32}$$

With the approximated KL distance between any two speakers, we can further define the average distance from one speaker to the overall speakers, the population.

$$SK(S_i) = \frac{1}{\|S\|} \sum_{S_k \in S} D(S_i, S_k) \,, \tag{33}$$

where S is the set of the speaker population and $\|S\|$ is the total number of speakers in S. Then speaker basis are selected as the speakers with the top N maximum average distance (MaxAD). In fact, the essential point behind the MaxAD is to select the points that are close to the boundary of the point clustering, since their average distances to all the other points tend to be larger than those of the internal points. This can be proved as follows. Suppose there exists an internal point p that is right to one boundary point p', left to all the other N-2 points, and it has a MaxAD to all the other points. We can prove that

$$MaxAD(p') > MaxAD(p) \,.$$

This is because: for any other p_i that is right to p, ppi<p′pi, due to p′ is a boundary point and is right to p, otherwise p′ is not a boundary point (a boundary point p′ has to be outside the circle with the centroid p_i and the diameter pp_i; otherwise p is also a point on the boundary). So we have

$$\sum_{i}^{N-2} p_i p + pp' < \sum_{i}^{N-2} p_i p' + pp' \text{, that is } MaxAD(p') > MaxAD(p) \,.$$

But this is contradictory to the fact that the point p has a MaxAD distance. Therefore, the point with the MaxAD must be closer to on the boundary of the point clustering. □

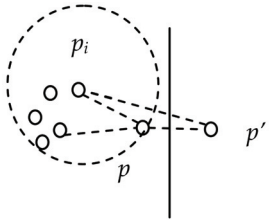


Fig. 3. Schematic demonstration for proof of the point with MaxAD must be on the boundary of the point cluster.

Thus, with an NLDA implemented by an MLP that is trained on selected basis speakers, raw features are transformed into discriminative features, which are more robust to speaker recognition.

4. Some recent efforts

Besides the fundamental normalizations and transformations at the both feature and score levels that we have mentioned in above sections, there have been some recent efforts on this domain, especially on the application of NLDA for transformation.

As introduced in Section 4.7, an NLDA is often implemented by an MLP trained on speaker identities. This is a straightforward idea to use an MLP for the generation of discriminative features for robust speaker recognition. However, it does not stop the possibility of using other class identities for NLDA training. For instance, Stoll et al. (2007) has been recently investigated using phone classes as recognition targets for speaker recognition. They found

that the discriminative features generated with such a trained MLP also contain a lot of information for speaker discrimination. This method is named as Tandem/HATS-MLP, because such a structure for MLP training was firstly proposed for speech recognition (Chen et al., 2004; Zhu et al. 2004).

The investigation of the complementary property of discriminative features to other types of features and modeling approaches, is an alternative direction for extending the method of NLDA for robust speaker recognition. Konig and Heck first found that discriminative features are complementary to the mel-scaled cepstral features (MFCCs) and suggested these two different types of features can be linearly combined as composite features (Konig et al. 1998; Heck et al. 2000). Stoll et al. (Stoll et al. 2007) used the discriminative features as inputs for support vector machines (SVM) for robust speaker recognition and found it outperforms the conventional GMM method. All these efforts confirm that the discriminative features are harmonic to other feature types as well as other modeling techniques.

Using multiple MLPs (MMLPs) as feature transformer is also a possible extension to the NLDA framework. The MMLP scheme was proposed to exploit the local clustering information in the distribution of the raw feature space (Wu, 2008b). According to the idea of MMLP, the raw feature space is firstly partitioned into a group of clusters by using either phonetic labeling information or an unsupervised recognition method of GMM. Then a sub-MLP is employed as a feature transformer within each cluster. The obtained discriminative features are then "softly"(using affine combination) or "hardly" (using switch selection) combined into the final discriminative features for recognition. This method has been found to outperform the conventional GMM method and to be marginally better than SMLP. More work is needed in future study.

5. Discussions: capacities and limitations

In this section, we shall comment, mainly from two different perspectives, capacities and limitations of normalization and transformation methods at both score and feature levels. In one way, we discuss how these techniques help to reduce mismatch between training and test conditions is considered. In the second one, we discuss how these methods are combined with other parts of SR systems, as either early or late processing module.

5.1 Mismatch reduction

How to deal with noises is a universal topic for robust speaker recognition. All the methods reviewed in this chapter are absolutely concerned with this topic. However, noises can normally be categorized into two groups, i.e., additive noise and convolutional noise. The additive noise further consists of environmental noise and cross-talking from other speakers, while convolutional noise is mainly caused by communication channels. For most noise, it occurs in a sudden and unpredictable way. Its interference results in a huge mismatch from the models trained before-hand, which in turn often severely degrades system performance. Thus, for robust speaker recognition, one of the key problems is how to reduce mismatch between different training and test scenarios. Of course, mismatch always exits, no matter how hard a system is carefully prepared. So a smart system knows how to compensate a variety of mismatches, whereas a poor system does not. We shall see, in such a context, how well each of the normalization methods works in terms of mismatch reduction.

Let us first check score level normalizations. Z-norm validates its method on the basis of an assumption that scores of an impostor model should hold the same tendency as the hypothesized one when they are under the same circumstances, obviously including mismatched conditions, such as noises. Z-norm should have general capabilities to reduce mismatch caused by both additive and convolutiional noises. However, this method may fail when scores of the impostor are not tightly linked in change tendency with scores of the claimant, which is often highly possible in realistic applications. T-norm is an extension to Z-norm, and still based on the same assumption. So T-norm has the similar capacities and limitations for mismatch reduction. However, T-norm is better than Z-norm, since distribution statistics of impostors are directly estimated from a separate test set. Hence, it can further reduce the mismatch between training and test conditions. WMAP is a two-step method. Its first step is quite similar to Z-norm, but WMAP enhances its normalization power by converting its decision scores into posterior probabilities so as to be much easier to be compared using a single global threshold. However, WMAP doesn't use any test data because WMAP uses scores calculated from the world model, not from the distribution of test data. It may prevent from enhancing the capacities of WMAP to reduce mismatch. Therefore, this may imply a possible extension to WMAP, in which a score is calculated from a world model estimated on the test set, and then used for the normalization in the first step, the posterior probability is computed in the second step. This scheme can be referred to as T-WMAP, which can extend WMAP to possibly better capacities for mismatch reduction. D-norm is a quite different approach, but it is also based on the same assumption as Z-norm, that is, normalization based on impostor scores may eliminate mismatch between training and test set. The only distinguished aspect is that D-norm uses Monte-Carlo simulation to estimate KL-distance to replace the impostor scores, conditioned on a strong correspondence experimentally found between them. In addition, D-norm does not use the test data either. Hence, it suggests a possible extension for D-norm, which employs test data, and therefore should be referred to as TD-norm.

We further check feature level transformations. CMS and spectrum subtraction are obviously based on an assumption that noise is stationary across a whole utterance, e.g. convolutional noises. Therefore, CMS is useful to reduce the mismatch caused by stationary noise in an utterance. However, it does not excel at dealing with some non-stationary noise, such as dynamic ambient noise and cross-talking. RASTA uses a set of band-pass filters to suppress constant or slowly-varying noises. At this point, RASTA is quite similar to CMS on the effect to eliminate convolutional noise, whereas it has limited capacities to handle the mismatch resulted in by other dynamic noises. H-norm is an extension of CMS to deal with handset mismatch in multiple energy channels. Based on the same assumption, H-norm is also mostly effective to handle constant channel noises, especially for reduction of handset mismatch. C-norm is a technique to design for alleviating the mismatch due to different cellular phones. It further extends CMS by transforming a channel dependent feature to a channel independent feature with the consideration of not only the mean vector, but also covariance matrix of a feature space. From this perspective, C-norm is an advanced version of CMS with more powerful capacities to normalize data to a universal scale with zero mean and unit variance. However, C-norm is also based on a similar assumption and therefore especially effective for the mismatch due to stationary noise. From here, we can see that all the above techniques can in fact be put into a framework based on the same assumption that

noise is stationary across an utterance. This obviously renders these normalization methods particular efficient for convolution noises as in telephony speech.
The other subgroup of normalization methods at the feature level doesn't base on the assumption of stationary noise. But, instead, the transformation is learned from an overall training set and thus the methods in this group take advantage of more data. In this group, PCA is a non-discriminant method. It seeks several largest variance directions to project a raw feature space to a new feature space. During this process, PCA often makes the projected feature space more compact and therefore is employed as a popular method of dimension reduction. An implement assumption is that noise components may not have the same directions as principal variances. So it may reduce the noise components that are vertical to the principal variances by applying PCA to the raw features. However, the noise components that are horizontal to the principal variances still remain. According to this property, PCA has a moderate capacity to deal with the mismatch resulted from a wide range of noises. LDA extends PCA with the same feature of dimension reduction, to linear discriminative training. The discriminative learning is efficient to alleviate the mismatch caused by all the noise types, whatever is additive or convolutional noise. Only the essential characteristics related to class identities are supposed to remain, while all other negative noisy factors are supposed to be removed. NLDA further extends LDA to a case of nonlinear discriminative training. As nonlinear discriminative training can degenerate to the linear discriminative training by manipulating a special topology of MLP (see Section 3.7), NLDA has more powerful capacities to enhance "useful" features and compensates harmful ones. In terms of mismatch reduction, NLDA and LDA are applicable to a wide range of noise, so as to be considered as broad-spectrum "antibiotic" for robust speaker recognition, whereas the others as narrow-spectrum "antibiotic".
Based on the above analysis, we can summarize that some of normalization techniques are particularly calibrated for specific types of mismatch, where others are generally effective. Roughly speaking, score normalizations are often more "narrow", but the feature level transformations are more or less "broad" in mismatch reduction. This distinction may imply the possibility of integrating these techniques into a single framework.

5.2 Processing at early and late stage

Another distinction between feature and score transformation is the time at which transformation is applied. Transformation is applied at the early stage for feature transformation, while it is used at the late stage for score level normalization. Generally speaking, the earlier processing is applied; the less chance noise components may negatively affect system performance. Judging from this pointview, feature transformation should attain more powerful capacities to deal with noise components, especially in a wider range, because the noise and useful parts are not blended by the modeling procedure yet. This definitely leaves more chances for robust normalization algorithms to work on noise reduction.
However, there are also advantages to apply normalization and transformation at the score level. The normalization methods at the score level often have lower computational complexity for implementation and application because at the score level, the high dimensional features have already been converted into a score domain that is often a scalar. Therefore, any further processing on the 1-d scores is pretty simple and efficient. This advantage thus renders the score normalizations are more efficient for realistic applications,

at least in terms of computation requirements. For instance, they can be efficiently applied to some real-time applications, such as applications running on PDAs, or mobile phones, whereas the feature normalizations have a bit higher complexity for these application scenarios.

Due to the pros and cons of feature and score level normalizations, we may raise such a question if it is possible to combine them into a universal framework and fully utilize their advantages. The answer is "Yes". One simple combination of feature and score level normalizations is definitely possible. However, these methods can also be integrated. For instance, it may be possible to apply NLDA at the score level, to project raw scores to better discriminative scores, as does NLDA for feature projection. This pre-processing at the score level can be referred to NLDA score transformation. By such an NLDA transformation, some confused scores may be corrected. This is just as an example, given for elucidating the idea of combining feature and score level transformation. Bearing in mind this idea, many other similar methods may be proposed to further improve robustness of speaker recognition systems in noisy conditions.

6. Final remarks: possible extensions

With summarization of work of normalization and transformation at the feature and score levels, we almost have gone through the most popular techniques for robust speaker recognition. However, another possible direction of work should not be ignored, that is transformation at the model level. This has seldom been investigated in the literature, therefore we did not summarize this direction as a separate section. Instead, we put it here in a section to discuss possible extensions. If transformation is applied at the model level, this type of techniques is often referred to as model adaptation or compensation in speech recognition (Gales, 1995). So if a similar method is applied to robust speaker recognition, we may refer to it as speaker model compensation (SMC). A direct application of SMC is to adapt a well trained model to a particular mismatched condition only using test data. If the test data is adequate, multiple transforms can be applied to different groups of Gaussian mixtures. Otherwise, a global transform can be applied to the overall Gaussian components. In future work, this idea is worth to be investigated.

Another example of the application of model level transformation goes to recent work of using speaker model synthesis based on cohort speakers (Wei et al., 2007), although it embodies the idea of synthesis indeed, but not the idea of adaptation. In this method, any new speaker model is synthesized by a set of cohort speaker models. The mismatch can be reduced in this manner by cohort speaker models being trained before hand based on channel dependent data.

With the model transformation being added, a general framework for robust speaker recognition is complete with all three directions. The future work will substantially rely on them for further development of robust speaker recognition. However, compared with feature and score level transformation, there are relatively fewer efforts dedicated at the model level. More work could be done under this direction with the possibility of successful extensions, in terms of the authors' point of view. It should be promising for future work.

In the chapter, we have mentioned some extensions as possible future work for robust speaker recognition. In order to make it clearer to readers, who may be of interest, we shall summarize them in the following as the final remarks:

- T-WMAP: extending WMAP to test set (Section 5.1).

- TD-nom: extending D-norm to test set (Section 5.1).
- NLDA transformation at the score level (Section 5.2).
- SMC adaptation for speaker recognition (Section 6).
- SVM score transformation: this is another possibility to apply NLDA at the score level. Instead of using MLP, support vector machine (SVM) is also possible to be used for score normalization.
- Combination of the feature, model and score transformations (Section 5.2).

The possible extensions can never be exhausted. We listed them here just as examples to demonstrate the possible direction for future work.

7. Conclusion

Robust speaker recognition faces a variety of challenges for identifying or verifying speaker identities in noisy environments, which cause a large range of variance in feature space and therefore are extremely difficult for statistical modeling. To compensate this effect, much research efforts have been dedicated in this area. In this chapter, we have mainly summarized these efforts into two categories, namely normalization techniques at the score level and transformation techniques at the feature level. The capacities and limitations of these methods are discussed. Moreover, we also introduce some recent advances in this area. Based on these discussions, we have concluded this paper with possible extensions for future work.

8. References

Auckenthaler, R.; Carey, M., et al. (2000). Score normalization for text-independent speaker verification systems. *Digital Signal Processing* Vol. 10, pp. 42-54.

Ben, M.; Blouet, R. & Bimbot, F. (2002). A Monte-Carlo method for score normalization in Automatic speaker verification using Kullback-Leibler distances. *Proceedings of IEEE ICASSP '02*, vol. 1, pp. 689–692.

Bimbot, F., et al. (2004). A tutorial on text-independent speaker verification. EURASIP. Vol. 4, pp. 430-451.

Bishop, M. (2006). Pattern Recognition and Machine Learning, Springer Press.

Campbell, J. P. (1997). Speaker recognition: a tutorial, *Proceedings of IEEE* Vol. 85, No. 9, pp. 1437-1462.

Chen, B.; Zhu, Q. & Morgan, N. (2004). Learning long-term temporal features in LVCSR using neural network. *Proceedings of ICSLP'04*.

Daugman, J. G. (2004). How iris recognition works. *IEEE Trans. Circuits and Syst. For Video Tech.* Vol. 14, No. 1, pp. 21-30.

Duda, R. & Hart., P. (2001). *Pattern classification*, Willey Press.

Fredouille, C.; Bonastre, J.-F. & Merlin, T. (1999) . Similarity normalization method based on world model and a posteriori probability for speaker verification. *Proceedings of Eurospeech'99*, pp. 983–986.

Gales, M. (1995). Model based techniques for robust speech recognition. *Ph.D. thesis*, Cambridge University.

Heck, L.; Konig, Y. et al. (2000). Robustness to telephone handset distortion in speaker recognition by discriminative feature design. *Speech Communication* Vol. 31, pp. 181-192.

Hermansky, H. (1990). Perceptual linear prediction (PLP) analysis for speech. *J. Acoustic. Soc. Am.*, pp.1738-1752.

Hermansky, H.; Morgan, N., et al. (1991). Rasta-Plp speech analysis. *ICSI Technical Report TR-91-069*, Berkeley, California.

Jain, A. K. & Pankanti, S. (2000). Fingerprint classification and recognition. *The image and video handbook*.

Jin, Q. & Waibel, A. (2000). Application of LDA to speaker recognition. *Proceedings of ICSLP'00*.

Konig, Y. & Heck, L. (1998). Nonlinear discriminant feature extraction for robust text-independent speaker recognition. *Proceedings of RLA2C, ESCA workshop on Speaker Recognition and its Commercail and Forensic Applications*, pp. 72-75.

Li, K. P. & Porter, J. E. (1988). Normalizations and selection of speech segments for speaker recognition scoring. *Proceedings of ICASSP '88*, Vol. 1, pp. 595–598.

Morris, A. C.; Wu, D. & Koreman, J. (2005). MLP trained to classify a small number of speakers generates discriminative features for improved speaker recognition. *Proceedings of IEEE ICCST 2005*.

Reynolds, D.A. (1996). The effect of handset variability on speaker recognition performance: experiments on the switchboard corpus. *Proceedings of IEEE ICASSP '96*, Vol. 1, pp. 113–116.

Reynolds, D. A. (2002). An Overview of Automatic Speaker Recognition Technology. *Proceedings of ICASSP'02*.

Reynolds, D A. (2003). Channel robust speaker verification via feature mapping. *Proceedings of ICASSP'03*, Vol. 2, 53-56.

Rosenberg, A.E. & Parthasarathy, S. (1996). Speaker background models for connected digit password speaker verification. *Proceedings of ICASSP '96*, Vol. 1, pp. 81–84.

Stoll, L.; Frankel, J. & Mirghafori, N. (2007). Speaker recognition via nonlinear discriminant features. *Proceedings of NOLISP'07*.

Sturim, D. E. & Reynolds, D.A. (2005). Speaker adaptive cohort selection for tnorm in text-independent speaker verification. *Proceedings of ICASSP'05,* Vol. 1, pp. 741–744.

Wei, W.; Zheng, T. F. & Xu, M. X. (2007). A cohort-based speaker model synthesis for mismatched channels in speaker verification. *IEEE Trans. ON Audio, Speech, and Language Processing*, Vol. 15, No. 6, August, 2007.

Wu, D.; Morris, A. & Koreman J. (2005a). MLP internal representation as discriminative features for improved speaker recognition. in *Nonlinear Analyses and Algorithms for Speech Processing Part II* (series: Lecture Notes in Computer Science), pp. 72-80.

Wu, D.; Morris, A. & Koreman, J. (2005b). Discriminative features by MLP preprocessing for robust speaker recognition in noise. *Proceedings of ESSV 2005*, 2005, pp 181-188.

Wu, D. (2008a). Discriminative Preprocessing of Speech, VDM Verlag Press, ISBN: 978-3-8364-3658-8.

Wu, D.; Li, J. & Wu, H. (2008b). Improving text-independent speaker recognition with locally nonlinear transformation. *Technical report*, Computer Science and Engineering Department, York University, Canada.

Young, S., et al. (2002). The HTK book V3.2. Cambridge University.

Zhu, Q.; Chen, B & Morgan, N. (2004). On using MLP features in LVCSR. *Proceedings of ICSLP'04*.

18

Speaker Vector-Based Speaker Recognition with Phonetic Modeling

Tetsuo Kosaka, Tatsuya Akatsu, Masaharu Kato and Masaki Kohda
Yamagata University
Japan

1. Introduction

This chapter describes anchor model-based speaker recognition with phonetic modeling. Gaussian Mixture Models (GMMs) have been successfully applied to characterize speakers in speaker identification and verification when a large amount of enrolment data to build acoustic models of target speakers is available. However, a small amount of enrolment data of as short as 5 sec. might be preferred for some tasks. A conventional GMM-based system does not perform well if the amount of enrolment data is limited. In general, 1-minute or more of enrolment data are required in the conventional system.

In order to solve this problem, a speaker characterization method based on anchor models has been proposed. The first application of the method was proposed for speaker indexing (Sturim et al., 2001). And the method has been also used for speaker identification (Mami & Charlet, 2003) and speaker verification (Collet et al., 2005).

In the anchor model-based system, the location of each speaker is represented by a speaker vector. The speaker vector consists of the set of the likelihood between a target utterance and the anchor models. It can be considered as a projection of the target utterance in a speaker space. One of the merits of this approach is that it is not necessary to train a model for a new target speaker, because the set of anchor models does not include the model of target speaker. It can save users time to utter iteratively for model training.

However, there is a significant disadvantage in the system because the recognition performance is insufficient. It has been reported that an identification rate of 76.6% was obtained on a 50-speaker identification task with 16-mixture GMMs as anchor models (Mami & Charlet, 2003). Also, an equal error rate (EER) of 11.3% has been reported on speaker verification task with 256-mixture GMMs (Collet et al., 2005). Compared with the conventional GMM approach, the performance of anchor model-based system is remarkably insufficient.

The aim of this work is to improve the performance of the method by using phonetic modeling instead of the GMM scheme as anchor models and to develop text-independent speaker recognition system that can perform accurately with very short reference speech. A GMM-based acoustic model covers all phonetic events for each speaker. It can represent an overall difference in acoustic features between speakers, however, it cannot represent a difference in pronunciation. Consequently, we propose the method to detect the detailed difference in phonetic features and try to use it as information for speaker recognition. In order to detect the phonetic features, a set of speaker-dependent phonetic HMMs is used as

the anchor models. The likelihood calculation between the target utterance and the anchor models is performed with an HMM-based phone recognizer with a phone-pair grammar.
In order to evaluate the proposed method, we compare the phonetic-based system with the GMM-based system on the framework of the anchor model. The number of dimensions in speaker space is also investigated. For this purpose, a large-size speech corpus is used for training the anchor models (Nakamura et al., 1996). Furthermore, another anchor model-based system in which phonetically structured GMMs (ps_GMMs) are used is compared to show the reason why phonetic modeling is effective in this method. Phonetically structured GMMs have been proposed by Faltlhauser (Faltlhauser & Ruske, 2001) to improve speaker recognition performance. In the method, 'phonetic' mixture components in a single state are weighted in order to improve speaker recognition performance.
The rest of this introduction reviews some related work. Recently, some phonetic based methods have been proposed. Hebert et al. have proposed the speaker verification method based on a tree structure of phonetic classes (Hebert & Heck, 2003). This paper reported that the proposed phonetic class-based system overcame a conventional GMM approach. Park et al. have proposed a speaker identification method in which phonetic class GMMs for each speaker were used (Park & Hazen, 2002). Both two methods differ from our approach in that a model for a target speaker is needed. Kohler et al. have developed a speaker-recognition system based only on phonetic sequences instead of the method based on acoustic feature vectors (Kohler et al., 2001). In this method, a test speaker model is not an acoustic model and it is generated by using n-phone frequency counts. The work which has a similar motivation of reducing enrolment data has been proposed by Thyes (Thyes et al., 2000). In this work, the concept of 'eigenvoice' was used for representing a speaker space. The method of 'eigenvoice' was proposed for speaker adaptation on earlier work (Kuhn et al., 1998). A phonetic information was not used for speaker discrimination in that work.
This chapter is organized as follows. Section 2 describes the method of speaker recognition. Section 3 shows the results of speaker identification experiments. Finally, we conclude the paper and suggest future research in Section 4.

2. Speaker recognition with phonetic-based modeling

2.1 Conventional speaker recognition

The technology of the speaker recognition can be categorized into two fields: one is speaker identification, and the other is speaker verification. Speaker identification is a technique for assigning the input utterance to one person of a known speaker set, while speaker verification is a technique for confirming the identity of the input speaker. Although the anchor model-based method can be applied to both techniques, a speaker identification method is described mainly in the following sections. Since it is difficult to separate the speaker information from the phonetic one, many speaker recognition systems perform in a text dependent way. In those systems, users must utter a predefined key sentence. However, sometimes that is not acceptable to users. In this work, we have developed the speaker recognition system which behaves in a text independent way. In that system, users can utter an arbitrary sentence.
In conventional speaker recognition systems, GMMs have been successfully used to characterize speakers. The characteristics of a reference speaker are modeled by GMM,

$$p(o_t \mid \lambda) = \sum_k w_k b_k(o_t) , \tag{1}$$

with mixture weights w_k and Gaussian densities $b_k(o_t)$. The average log-likelihood of a model given an utterance $\boldsymbol{o} = \{o_1, \cdots, o_T\}$ is calculated as

$$L(\boldsymbol{o} \mid \lambda) = \frac{1}{T}\sum_{t=1}^{T} \log p(o_t \mid \lambda) . \tag{2}$$

The average log-likelihood scores are compared to determine an input speaker in speaker identification system. For speaker verification system, those scores are normalized to reduce the variation of utterances,

$$\widetilde{L}(\boldsymbol{o} \mid \lambda) = L(\boldsymbol{o} \mid \lambda) - L(\boldsymbol{o} \mid \lambda_{UBM}) , \tag{3}$$

where λ_{UBM} is the Universal Background Model which is derived from training data of all or selected speakers to normalize the variation. For the tasks of speaker identification or verification, a reference speaker model λ must be trained by using 60 sec. or more of enrolment speech in advance. It causes users the loss of taking time to utter iteratively. Since acoustic models of reference speakers are not required in anchor model-based system, the user has only to utter just one sentence in advance.

2.2 Speaker space representation using anchor models

In the anchor model-based system, the speaker is characterized by a vector consisted of the set of the likelihood between the target utterance and the anchor models. The speech utterance is represented by the following vector:

$$\boldsymbol{v} = \begin{bmatrix} \dfrac{\log p(\boldsymbol{o} \mid M_1) - \mu}{\sigma} \\ \dfrac{\log p(\boldsymbol{o} \mid M_2) - \mu}{\sigma} \\ \vdots \\ \dfrac{\log p(\boldsymbol{o} \mid M_N) - \mu}{\sigma} \end{bmatrix} , \tag{4}$$

where

$$\mu = \frac{1}{N}\sum_{n=1}^{N} \log p(\boldsymbol{o} \mid M_n) , \tag{5}$$

and

$$\sigma = \sqrt{\frac{1}{N}\sum_{n=1}^{N} \left(\log p(\boldsymbol{o} \mid M_n) - \mu\right)^2} . \tag{6}$$

$\log P(\boldsymbol{o} \mid M_n)$ is the log-likelihood of the input utterance $\boldsymbol{o}$ for the anchor model M_n. The vector is normalized to have zero mean and unit variance to reduce the likelihood variation among utterances (Akita & Kawahara, 2003). N is the number of anchor models and denotes the number of dimensions of speaker space. In the identification step, a measure

between each reference speaker vector $\boldsymbol{r}_i$ and the input vector is calculated in speaker space. We used Euclidean metric for distance calculation. Input speaker is identified by:

$$\hat{i} = \arg\min_i D(\boldsymbol{v}, \boldsymbol{r}_i) \ , \tag{7}$$

where i is a reference speaker index and input vector $\boldsymbol{v}$ is determined to be uttered by speaker $\hat{i}$. Note that the target speaker is not included in N speakers of anchor models. Since the method doesn't require the model training for the target speaker, only about single utterance is needed for reference vector.

2.3 Phonetic representation of anchor models

In the previous works, GMMs were used as the anchor models (Sturim et al., 2001; Mami & Charlet, 2003). A GMM covers all phonetic events for each speaker, however it does not directly consider phonetic information. It can cause the degradation in the performance of speaker discrimination. For example, the vowel /a/ of speaker A is sometimes confused with the vowel /o/ of speaker B in a phoneme recognition task. The GMM-based system cannot represent such a difference. Consequently, we propose the method to detect the detailed difference in phonetic features and try to use it as information for speaker recognition.

In order to improve the performance of the method, phonetic HMMs or phonetically structured GMMs are used. In this section, phonetic HMM based system is described. This approach requires a phonetic speech recognizer in order to calculate the log-likelihood $\log P(\boldsymbol{o} \mid M_n)$ shown in Eq. (4). The log-likelihood for the speaker n is obtained by a phoneme recognizer with speaker dependent phonetic HMMs. Since the recognizer can decode an unknown utterance, the identification system can be performed in a text independent way.

Figure 1 indicates the examples of speaker vector composed of 1000 dimensions with both HMMs and GMMs. The horizontal axis represents the values of speaker vector for the utterance number 26, and the vertical axis represents the values for the utterance number 27. Both utterances are given by the same speaker (female), however, the contents of utterances are different. Each point represents the values of one of the 1000 anchor models. The left figure shows the vector values with HMMs as anchor models, and the right one shows the vector values with GMMs as anchor models. Even though contents are not same between horizontal axis and vertical axis, two sets of values indicate a similar tendency in these figures. This observation suggests that the speaker identification in text independent way can be performed with this method. In the left figure, the correlation between two is smaller than that of the right one. This suggests that speaker recognition performs well by using HMMs as anchor models.

Figure 2 also indicates the examples of speaker vector composed of 1000 dimensions with both HMMs and GMMs. In this figure, the horizontal axis and the vertical axis represent the different speakers (F.AIFU and F.HAHZ are speaker IDs. Both speakers are female). However, the contents of utterances are same.

In contrast to the results of Fig. 1, a correlation between two axes is small. This means that the values of speaker vector differ too much between different speakers even if the contents of utterances are same.

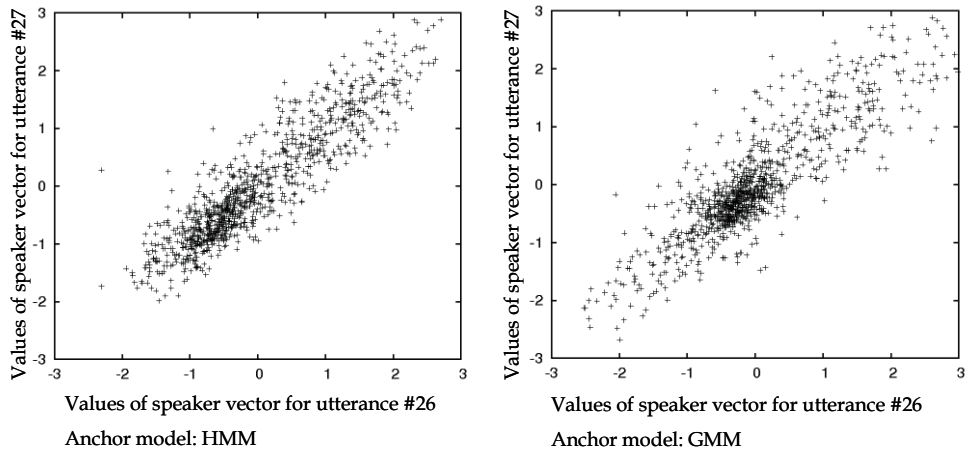


Fig. 1. Example of values of speaker vector. The horizontal and vertical axes represent the same speaker. (the left figure: HMMs are used as anchor models, the right figure: GMMs are used as anchor models)

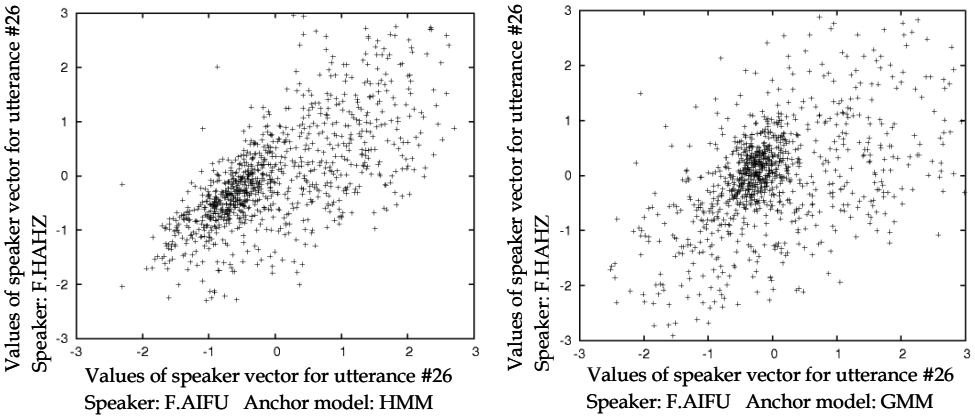


Fig. 2. Example of values of speaker vector. The horizontal and vertical axes represent the different speakers. (the left figure: HMMs are used as anchor models, the right figure: GMMs are used as anchor models)

2.4 Phonetically structured GMMs as anchor models

In our approach, phonetically structured GMMs (ps_GMMs) (Faltlhauser & Ruske, 2001) are also used as anchor models to find the reason why phonetic modeling is effective in the proposed method. In the method, 'phonetic' mixture components in a single state are weighted in order to create a GMM based on phonetic classes. In our work, PDFs obtained from monophone HMMs are used for 'phonetic' mixture components.

Assume that S-state and K-mixture monophone HMM for phoneme j is trained in advance. Total number of PDFs in monophone HMMs is $J \times S \times K$ when the number of phonemes is J. Those PDFs are gathered to make a single state model λ_{ps},

$$p(o_t \mid \lambda_{ps}) = \sum_{j=1}^{J} \sum_{s=1}^{S} \sum_{k=1}^{K} w_{jsk} N(o_t, \mu_{jsk}, \Sigma_{jsk}), \tag{8}$$

where $N(o_t, \mu_{jsk}, \Sigma_{jsk})$ is a Gaussian distribution of the k th density in the mixture at state s of phoneme j and w_{jsk} is a mixture weight.

After producing λ_{ps}, parameters of λ_{ps} are re-estimated. Three types of methods were compared in a speaker identification task in order to find the best re-estimation method.

No re-estimation Re-estimation is not conducted. All PDFs from monophone HMMs are used as they are. New mixture weight $\hat{w}_{jsk}$ is simply given as follows:

$$\hat{w}_{jsk} = \frac{w_{jsk}}{JS}. \tag{9}$$

Weight re-estimation Only mixture weights are re-estimated to obey the probabilistic constraint as follows:

$$\sum_{j=1}^{J} \sum_{s=1}^{S} \sum_{k=1}^{K} w_{jsk} = 1. \tag{10}$$

PDF and weight re-estimation Both all of PDFs and weights are re-estimated. In this case, phonetic information is only used as an initial model. Then the method is not exactly a phonetic modeling.

In the experiment of a 30-speaker identification task, 'weight re-estimation' and 'PDF and weight re-estimation' obtained similar identification rates. The identification rates were 89.88% and 89.99%, respectively. The performance of 'no re-estimation' was insufficient and the identification rate was 82.64%. The method of 'weight re-estimation' is used in the following experiments.

In terms of model topology, ps_GMM consists of a single state just like the conventional GMM, however, each group of PDFs in ps_GMM is trained with the data of each phoneme class. Then comparing conventional GMM with ps_GMM, the model topology is same but PDFs are different. Also comparing ps_GMM with phonetic HMM, PDFs are same but the model topology is different.

3. Speaker identification experiments

3.1 Speaker identification system

An experimental system of speaker identification has been developed and used for evaluation. In this section, we describe the overview of the system. Fig. 3 shows the block diagram of the proposed speaker identification system.

Since we are planning to develop a speaker identification method in noisy conditions, the analysis is processed by the ETSI advanced front-end (AFE-WI008) in which noise robust

algorithms are used (ETSI, 2002). In this front-end, noise reduction for additive noise and blind equalization for channel distortion are applied. The blind equalization process is omitted for our experiments, because we found that it had a bad influence on the performance of speaker identification from the results of comparative experiments. In this front-end, a speech signal is digitized at a sampling frequency of 16kHz and at a quantization size of 16bits. The length of the analysis frame is 25ms and the frame period is set to 10ms. The 13-dimensional feature (12-dimensional MFCC and log power) is derived from the digitized samples for each frame. Additionally, the delta and delta-delta features are calculated from MFCC feature and the log power. Then the total number of dimensions is 39. The delta and delta-delta coefficients are useful for the system of HMM-based anchor models. After speech analysis is carried out, an input features are transformed into a speaker vector by Eq. (4). A value of log-likelihood in Eq. (4) is obtained by a phoneme recognizer with a phone-pair grammar. In the recognizer, one-pass frame-synchronous search algorithm with beam searching has been adopted. The speaker vector derived from the input utterance is used for calculating distances from reference speaker vectors, which are computed in advance. The reference speaker of minimal distance is determined to be an identified speaker by Eq. (7).

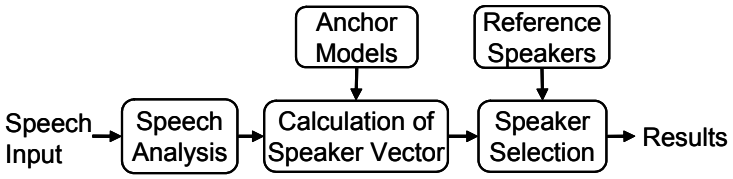


Fig. 3. Block diagram of speaker identification system

3.2 Experimental set-up

For evaluation of the proposed method, we used ATR SDB-I as a speech corpus (Nakamura et al., 2001). This corpus was designed to cover speaker variations with a large number of speakers' read speech and dialogue speech. For representing the anchor models, phonetically-balanced speech data uttered by 2032 speakers composed of 744 male and 1288 female were used. Then the maximum number of dimensions of speaker space was 2032.

Since a limited amount of speech data from each speaker was available, we used the maximum a *posteriori* (MAP) estimation instead of the ML (Maximum Likelihood) estimation for training of the anchor models. MAP estimation is successfully used for adaptation of CHMM parameters (Lee & Gauvain, 1993). It uses information from an initial model as *a priori* knowledge to complement the training data. This *a priori* knowledge is statistically combined with *a posteriori* knowledge derived from the training data. When the amount of training data is small, the estimates are tightly constrained by the *a priori* knowledge, and the estimation error is reduced.

The evaluation data sets consisted of 30 speakers, each of which contains 25 utterances. The average length of utterances in test set was 5.5 sec.

In order to avoid variations in identification performance with reference speech, the following evaluation method was adopted. 24 out of the 25 utterances by each evaluation speaker were tested, and the rest of one utterance was used as a reference speech. Then the 25 different references were used in the same way and the average identification rate of those results was calculated. Thus, the average length of test speech and that of reference

speech were same and were only 5.5 sec. In general, 1-minute or more of enrolment data are required in the conventional speaker recognition systems. Compared with those systems, the proposed system can be performed with a very short utterance. It can save users time to utter iteratively.

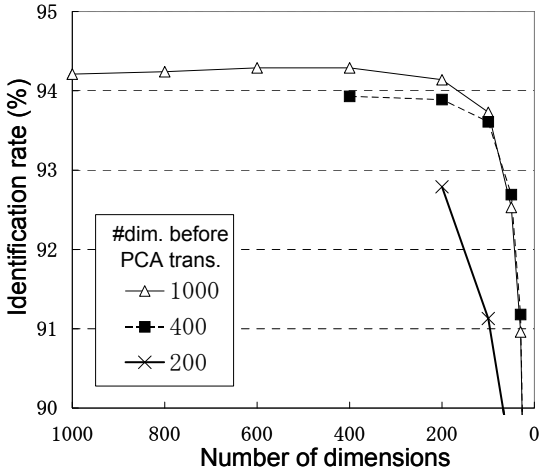


Fig. 4. Reduction of number of dimensions with PCA trans.

3.3 Influence of the number of dimensions

First, we evaluated the influence of the number of dimensions on a speaker space. In (Mami & Charlet, 2003), the speaker space was composed of 500 dimensions each of which was calculated with 16-mixture GMM. Since the detailed study is not carried out until now, an adequate number of parameters for representing speaker space is not clear. In the experiments, various numbers of dimensions from 200 to 2032 were compared. The result shows that the performance is saturated around 1000-dim. In order to study an influence of the number of dimensions further, we apply a technique of PCA (Principal Component Analysis) to reduce a redundancy of a speaker space. Since speakers for anchor models are selected randomly from speaker list, features of some speakers are similar and may be redundant. Fig. 4 shows the results of reduction of number of dimensions on a 30-speaker identification task. Three types of experiments were conducted. The number of dimensions before PCA was 200, 400 and 1000, respectively. The results show that the degradation of recognition performance was not observed at the range of 400 to 1000-dim. This means that there is some redundancy in representing the speaker space.

3.4 Comparison of anchor models

In this section, three types of anchor models are compared on 30-speaker identification task: 3-state 10-mixture phonetic HMMs (the number of phonemes is 34), 1020-mixture phonetically structured GMMs (ps_GMMs) and 1024-mixture conventional GMMs. For the phonetic HMMs, total number of PDFs is 3state × 10mixture × 34phonemes = 1020. Then the similar number of PDFs is used for three types of models, and they are comparable. The number of model parameters was determined experimentally. The details of the relation

between the number of the mixture components and the identification rate have been reported in (Kosaka et al., 2007). The ps_GMMs in this experiment are composed as follows. All PDFs except those in silence model are extracted from the phonetic HMMs to form a single state GMM. After producing the GMM, only mixture weights are re-estimated.
Table 1 shows the speaker identification result. The number of dimensions was 1000 and the number of test speakers was 30. The HMM-based system showed significant improvement over the GMM-based system, although the number of PDFs was nearly same in those systems. The performance of ps_GMMs is in between two. Comparing conventional GMM with ps_GMM, the model topology is same but PDFs are different. Also comparing ps_GMM with phonetic HMM, PDFs are same but the model topology is different. This means that both the model topology and the PDFs derived from phonetic models contributed to improve the performance of speaker identification. Finally, the identification rate of 94.21% could be obtained with 3-state 10-mixture HMM system in 30 speaker identification task. By comparison with the GMM-based system, the relative improvement of 62.9% was achieved.
We also investigated the comparison between an anchor model-based system and a conventional GMM-based system described in Sect. 2.1. In our experiments, the average length of reference speech was only 5.5 sec. and it is difficult to train GMMs accurately by using the ML (Maximum Likelihood). Thus, we used the maximum a *posteriori* (MAP) estimation instead of the ML estimation for training of conventional GMMs. The number of mixture components was varied to find the most appropriate one. The speaker identification rate of 77.14% was obtained with 8-mixture GMMs. This indicates that conventional GMM-base system does not work well with such a small amount of enrolment data.

Anchor model	**HMM**	**ps_GMM**	**GMM**
Identification rate (%)	94.21	89.88	84.41

Table 1. Performance comparison of three types of anchor models (#test speakers = 30)

4. Conclusions

This chapter proposed the method of anchor model-based speaker recognition in text-independent way with phonetic modeling. Since the method doesn't require model training for the target speaker, only about single utterance is needed for reference speech. In order to improve the recognition performance, phonetic modeling was used instead of Gaussian Mixture Model (GMM) scheme as anchor models. The proposed method was evaluated on Japanese speaker identification task. Compared with the performance of GMM-based system, significant improvement could be achieved. The identification rate of 94.21% could be obtained with 3-state 10-mixture HMMs in 30-speaker identification task. In the experiments, the average length of reference speech was only 5.5 sec. By comparison with the GMM-based system, the relative improvement of 62.9% was achieved. The results show that the phonetic modeling is effective for anchor model-based speaker recognition.
We are now conducting the evaluation of the method on speaker verification task. We are also conducting the evaluation of speaker identification in noisy conditions. Some results in noisy conditions have been reported in (Goto et al., 2008). The merit of this method is that the system can detect speaker characteristics with a very short utterance as short as 5 sec. Then the method can be used in the tasks of speaker indexing or tracking.

5. References

Akita, Y. & Kawahara, T. (2003), Unsupervised speaker indexing using anchor models and automatic transcription of discussions, *Proceedings of Eurospeech2003,* pp.2985-2988, Geneva, Switzerland, Sept. 2003

Collet, M.; Mami, Y.; Charlet, D. & Bimbot, F. (2005), Probabilistic anchor models approach for speaker verification, *Proceedings of INTERSPEECH2005,* pp.2005-2008, Lisbon, Portugal, Sept. 2005

ETSI, (2002), ETSI ES 202 050 V1.1.1, *STQ; Distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms,* European Telecommunications Standards Institute, France

Faltlhauser, R. & Ruske, G. (2001), Improving speaker recognition performance using phonetically structured Gaussian mixture models, *Proceedings of Eurospeech2001,* pp.751-754, Aalborg, Denmark, Sept. 2001

Goto, Y.; Akatsu, T.; Katoh, M.; Kosaka, T. & Kohda, M. (2008), An investigation on speaker vector-based speaker identification under noisy conditions, *Proceedings of ICALIP2008,* pp.1430-1435, Shanghai, China, Jul. 2008

Hebert, M. & Heck, L.P. (2003), Phonetic class-based speaker verification, *Proceedings of INTERSPEECH2003,* pp.1665-1668, Geneva, Switzerland, Sept. 2003

Kohler, M.A.; Andrews, W.D. & Campbell, J.P. (2001), Phonetic speaker recognition, *Proceedings of EUROSPEECH2001,* pp.149-153, Aalborg, Denmark, Sept. 2001

Kosaka, T.; Akatsu, T.; Katoh, M. & Kohda, M. (2007), Speaker Vector-Based Speaker Identification with Phonetic Modeling, *IEICE Transactions (Japanese)*, Vol. J90-D, No. 12, Dec. 2007, pp. 3201-3209

Kuhn, R,; Nguyen, P.; Junqua, J.-C.; Goldwasser, L.; Niedzielski, N.; Fincke, S.; Field, K. & Contolini, M. (1998), Eigenvoices for speaker adaptation, *Proceedings of ICSLP98,* pp. 1771-1774, Sydney, Australia, Dec. 1998

Lee, C.-H. & Gauvain, J.-L. (1993), Speaker adaptation based on MAP estimation of HMM parameters, *Proceedings of ICASSP93*, pp.558-561, Minneapolis, USA, Apr. 1993, IEEE

Mami, Y. & Charlet, D. (2003), Speaker identification by anchor models with PCA/LDA post-processing, *Proceedings of ICASSP2003,* pp.180-183, Hong Kong, China, Apr. 2003, IEEE

Nakamura, A.; Matsunaga, S.; Shimizu, T.; Tonomura, M. & Sagisaka, Y. (1996), Japanese speech databases for robust speech recognition, *Proceedings of ICSLP1996,* pp.2199-2202, Philadelphia, USA, Oct. 1996

Park, A & Hazen, T.J. (2002), ASR dependent techniques for speaker identification, *Proceedings of ICSLP2002,* pp.1337-1340, Denver, USA, Sept. 2002

Sturim, D.; Reynolds, D. ; Singer, E. & Campbell, J. (2001), Speaker indexing in large audio databases using anchor models, *Proceedings of ICASSP2001,* pp.429-432, Salt Lake City, USA, May. 2001, IEEE

Thyes, O.; Kuhn, R.; Nguyen, P. & Junqua, J.-C. (2000), Speaker identification and verification using eigenvoices, *Proceedings of ICSLP2000,* pp. 242-246, Beijing, China, Oct. 2000

19

Novel Approaches to Speaker Clustering for Speaker Diarization in Audio Broadcast News Data

Janez Žibert[1] and France Mihelič[2]
[1]*Primorska Institute for Natural Science and Technology, University of Primorska*
[2]*Faculty of Electrical Engineering, University of Ljubljana*
Slovenia

1. Introduction

The growing demand to shift content-based information retrieval from text to various multimedia sources means there is an increasing need to deal with large amounts of multimedia information. The data provided from television and radio broadcast news (BN) programs are just one example of such a source of information. In our research we focus on the processing and analysis of audio BN data, where the main information source is represented by speech data. The main issues in our work relate to the preparation and organization of BN audio data for further processing in information audio-retrieval systems based on speech technologies.

This chapter addresses the problem of structuring the audio data in terms of speakers, i.e., finding the regions in the audio streams that belong to a single speaker and then joining each region of the same speaker together. The task of organizing the audio data in this way is known as speaker diarization and was first introduced in the NIST project of *Rich Transcription* in the "Who spoke when" evaluations (Fiscus et al., 2004; Tranter & Reynolds, 2006). The speaker-diarization problem is composed of several stages, in which the three main tasks are performed: speech detection, speaker- and background-change detection, and speaker clustering. While the aim of the speech detection and the speaker- and acoustic-segmentation procedures is to provide the proper segmentation of the audio data streams, the purpose of the speaker clustering is to join or connect together segments that belong to the same speakers, and this is usually applied in the last stage of the speaker-diarization process. In this chapter we focus on speaker-clustering methods, concentrating on developing proper representations of the speaker segments for clustering, and research different similarity measures for joining the speaker segments and explore different stopping criteria for the clustering that result in a minimization of the overall diarization error of such systems.

The chapter is organized as follows: In Section 2, two baseline speaker-clustering approaches are presented. The first is a standard approach using a bottom-up agglomerative clustering principle with the Bayesian information criterion as the merging criterion. In the second system an alternative approach is applied, also using bottom-up clustering, but the representations of the speaker segments are modeled by Gaussian mixture models, and for

the merging criteria a cross log-likelihood ratio is used. Section 3 is devoted to the development of a novel fusion-based speaker-clustering system, where the speaker segments are modeled by acoustic and prosody representations. By adding prosodic information to the basic acoustic features we have extended the standard clustering procedure in such a way that it will work with a combination of both representations. All the presented clustering procedures were assessed on two different BN audio databases and the evaluation results are presented in Section 4. Finally, a discussion of the results and the conclusions are given in Sections 5 and 6.

2. Speaker clustering in speaker-diarization systems

Speaker diarization is the process of partitioning the input audio data into homogeneous segments according to the speaker's identity. The aim of speaker diarization is to improve the readability of an automatic transcription by structuring the audio stream into speaker turns, and in cases when used together with speaker-identification systems, by providing the speaker's true identity. Such information is of interest to several speech- and audio-processing applications. For example, in automatic speech-recognition systems the information can be used for unsupervised speaker adaptation (Anastasakos et al., 1996, Matsoukas et al., 1997), which can significantly improve the performance of speech recognition in large vocabulary, continuous speech-recognition systems (Gauvain et al., 2002; Woodland, 2002; Beyerlein et al., 2002). This information can also be applied for the indexing of multimedia documents, where homogeneous speaker or acoustic segments usually represent the basic units for indexing and searching in large archives of spoken audio documents, (Makhoul et al., 2000). The outputs of a speaker-diarization system can also be used in speaker-identification or speaker-tracking systems, (Delacourt et al., 2000; Nedic et al., 1999).

Most speaker-diarization systems have a similar general architecture to that shown in Figure 1. First, the audio data, which are usually derived from continuous audio streams, are segmented into speech and non-speech data. The non-speech segments are discarded and not used in subsequent processing, which is done in a speech-detection module. The speech data are then chopped into homogeneous segments in an audio-segmentation module (marked as acoustic change detection in Figure 1). The segment boundaries are located by finding the acoustic changes in the signal, and each segment is, as a result, expected to contain speech from only a single speaker. The resulting segments are then clustered so that each cluster corresponds to just a single speaker. This is done in a speaker-clustering module and usually represents the final stage in speaker-diarization systems. At this stage, each cluster is labeled with relative speaker-identification names. Additionally, speaker identification or gender detection can be performed. In the first case, each of the speaker clusters can be given a true speaker name, or it is left unlabelled if the speech data in the cluster do not correspond to any of the target speakers. In the case of gender detection, each cluster gets an additional label to indicate to which gender it belongs. As such the speaker diarization of continuous audio streams is a multistage process made up of four main components: speech detection, speaker audio segmentation, speaker clustering, and speaker identification. The latest overview of the approaches used in speaker-diarization tasks can be found in (Tranter & Reynolds, 2006).

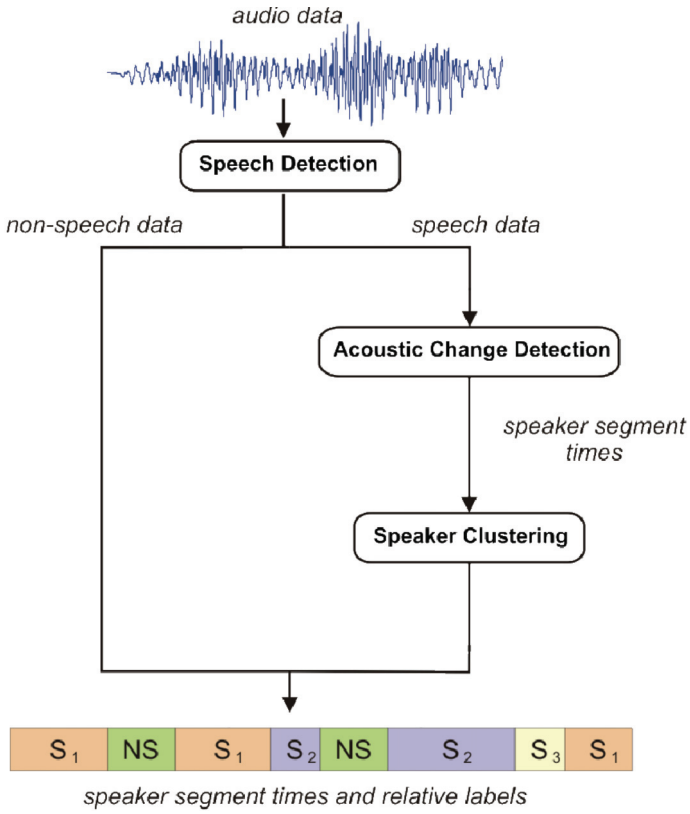


Fig. 1. The main building blocks of a typical speaker-diarization system. Most systems have components to perform speech detection, speaker- or acoustic-based segmentation and speaker clustering, which may include components for gender detection and speaker identification.

We built a speaker-diarization system that is used for speaker tracking in BN shows (Žibert, 2006b; Žibert et al., 2007). The system was designed in the standard way by including components for speech detection, audio segmentation and speaker clustering. Since we wanted to evaluate and measure the impact of speaker clustering on the overall speaker-diarization performance, we built a system where the components for speech detection and audio segmentation remained fixed during the evaluation process, while different procedures were implemented and tested in the speaker-clustering task.

The component for speech detection was derived from the speech/non-speech segmentation procedure, which was already presented in (Žibert et al., 2007). The procedure aimed to find the speech and non-speech regions in continuous audio streams represented by phoneme-recognition features (Žibert et al., 2006a). The features were derived directly from phoneme transcripts that were produced by a generic phone-recognition system. A speech-detection procedure based on these features was then implemented by performing a Viterbi decoding in the classification network of the hidden Markov models, which were previously trained on speech and non-speech data. This rather alternative approach to deriving speech-

detection features proved to be more robust and accurate for detecting speech segments (Žibert et al., 2006a; Žibert et al., 2007).

Further segmentation of the speech data was made by using the acoustic-change detection procedure based on the Bayesian information criterion (BIC), which was proposed in (Chen & Gopalakrishnan, 1999) and improved by (Tritchler & Gopinath, 1999). The applied procedure processed the audio data in a single pass, with the change-detection points found by comparing the probability models estimated from two neighboring segments with the BIC. If the estimated BIC score was under the given threshold, a change point was detected. The threshold, which was implicitly included in the penalty term of the BIC, has to be given in advance and was in our case estimated from the training data. This procedure is widely used in most current audio-segmentation systems (Tranter & Reynolds, 2006; Fiscus et al., 2004; Reynolds & Torres-Carrasquillo, 2004; Zhou & Hansen, 2000; Istrate et al., 2005; Žibert et al., 2005).

While the aim of an acoustic-change detection procedure is to provide the proper segmentation of the audio-data streams, the purpose of speaker clustering is to join together the segments that belong to the same speakers. In our system we realized three different speaker-clustering procedures, which are described in detail in the following sections.

The result of such a speaker-diarization system is segmented audio data, annotated according to the relative speaker labels (such as 'spk1', 'spk2', etc.). Each such speaker cluster can be additionally processed through the speaker-identification module to find the true identities of the speakers who are likely to be in the processing audio data (such as prominent politicians or the main news anchors and reporters in the BN data). This can be achieved by a variety of methods that can be performed during the speaker-clustering stage. However, finding the true identities of the speakers was not within the scope of this research.

2.1 Speaker clustering

The aim of speaker clustering in speaker-diarization systems is to associate or cluster together the segments from the same speaker. Ideally, this clustering produces one cluster for each speaker, with all the segments from a given speaker in a single cluster. The dominant approach used in diarization systems is called hierarchical agglomerative clustering (Theodoridis & Koutroumbas, 2003); it consists of the following steps (Tranter & Reynolds, 2006):

1. *Initialization:* each segment represents a single cluster;
2. *Similarity measure:* compute the pair-wise distances between each cluster;
3. *Merging step:*
   a. merge the closest clusters together;
   b. update the distances of the remaining clusters to the new cluster;
4. *Stopping criterion:* iterate step 3 until some stopping criterion is met.

The main issues concerning the above speaker-clustering approach include the choice of a proper similarity measure, the proper representations of the cluster data and finding a suitable stopping criterion. The audio data used for the speaker clustering is in general represented by acoustic features consisting of either mel-frequency cepstral coefficients (MFCCs) or perceptual linear-prediction coefficients (PLPCs), (Picone, 1993). The cluster data represented by these features are then usually modeled by Gaussian distributions, and the resulting similarity measures are computed as the likelihood functions from these

models (Moh et al., 2000; Reynolds & Torres-Carrasquillo, 2004; Sinha et al., 2005). The most common approach is to represent the clusters by single full-covariance Gaussian distributions, whereas for the similarity measure a Bayesian information criterion is used (Chen & Gopalakrishnan, 1999). For good performance of the clustering, the stopping criterion also needs to be properly chosen. A suitable stopping criterion should end the merging process at the point where the audio data from each speaker is concentrated mainly in one cluster, and in general it is set according to a similarity measure and cluster models that are used in the merging process of the speaker clustering.

In our research we implemented the same clustering approach, but we experimented with different similarity measures, different representations of the audio data and different cluster models. Three approaches were investigated. In the first one we followed the standard procedure of speaker clustering, based on the Bayesian information criterion. The alternative approach, which is also widely used in speaker-diarization systems and was also implemented in our study, aims to incorporate Gaussian mixture models into the speaker-clustering process. The audio data in both approaches are usually represented by a single stream of acoustic features (MFCCs, PLPs), which result in an acceptable performance of the speaker clustering in cases when the acoustic conditions do not change. But this is not the situation when dealing with BN data, since BN news is composed of audio data from various acoustic environments (different types of acoustic sources, different channel conditions, background noises, etc.). To improve speaker clustering in such cases we proposed an alternative representation of speech signals, where the acoustic parameterizations of the clusters were extended by prosodic measurements.

When speaker clustering is used as one stage in a speaker-diarization system, several improvements can be made to increase the performance of the speaker diarization, like joint segmentation and clustering (Meignier et al., 2000) and/or cluster re-combination (Zhu et al., 2005). Both methods aim to improve the base speaker-clustering results by integrating several speaker-diarization tasks together or re-running the clustering on under-clustered fragments of audio data. In our research we focused mainly on an evaluation and a development of the base speaker-clustering approaches, and did not implement any of these methods, even though they could be easily applied in the same manner as they are applied in other systems.

Also note that the presented agglomerative clustering approach is not the only possible solution for speaker clustering. This kind of approach is suitable in cases when all the audio data are available in advance. When the data need to be processed simultaneously, e.g., in the online processing of BN shows, other approaches need to be applied. The most common approach in this case is a sequential clustering, which needs to resolve the same operating issues as are present in an agglomerative clustering: what kind of data representation should be applied, how should the clusters be modeled, and what similarity measure should be used? Therefore, we decided to focus our research only on those components that are essential for the good performance of the speaker clustering, regardless of the approach that is being used.

2.2 Speaker clustering via the Bayesian Information Criterion

The most common approach for speaker clustering in speaker-diarization systems is agglomerative (bottom-up) clustering, where the Bayesian Information Criterion (BIC) is used as a similarity measure (Chen & Gopalakrishnan, 1999). The approach can be described in three main steps by following the agglomerative scheme presented in the previous section:

1. *initialization step:*
   - each segment C_i represents one cluster;
   - the initial clustering is $\vartheta_0 = \{C_i \mid i = 1, \ldots, N\}$
2. *merging procedure:*

   Repeat:
   - From among all the possible pairs of clusters (C_r, C_s) in ϑ_{t-1} find the one, say (C_i, C_j), such that

$$\Delta_{BIC}(C_i, C_j) = \min_{C_r, C_s} \Delta_{BIC}(C_r, C_s) \tag{1}$$

   - Define $C_q = C_i \cup C_j$ and produce new clustering

$$\vartheta_t = (\vartheta_{t-1} - \{C_i, C_j\}) \cup \{C_q\} \tag{v}$$

3. *stopping criterion:*
   - The merging procedure is repeated until in ϑ_t there exists such pairs (C_r, C_s), for which

$$\Delta_{BIC}(C_r, C_s) < 0.0 \tag{3}$$

In the *merging procedure* the joining of clusters is performed by searching for the minimum Δ_{BIC} score among all the possible pair-wise combinations of clusters. The Δ_{BIC} measure is defined as:

$$\begin{aligned}\Delta_{BIC}(C_i, C_j) = \frac{1}{2}\Big(\big(K_{C_i} + K_{C_j}\big)\log|\Sigma_{C_i \cup C_j}| - K_{C_i}\log|\Sigma_{C_i}| - K_{C_j}\log|\Sigma_{C_j}|\Big) \\ -\frac{\lambda}{2}\Big(d + \frac{1}{2}d(d+1)\Big)\log\big(K_{C_i} + K_{C_j}\big),\end{aligned} \tag{4}$$

where the clusters C_i, C_j and $C_i \cup C_j$ are modeled by the full-covariance Gaussian distributions $N(\mu_{C_i}, \Sigma_{C_i})$, $N\left(\mu_{C_j}, \Sigma_{C_j}\right)$ and $N(\mu_{C_i \cup C_j}, \Sigma_{C_i \cup C_j})$, respectively. K_{C_i} and K_{C_j} are the number of sample vectors in the clusters C_i and C_j, respectively, and d is a vector dimension. λ is an open parameter, the default value of which is 1.0. Note that the term $\log|\Sigma|$ corresponds to the log of a determinant of a given full-covariance matrix Σ.

The $\Delta_{BIC}(C_i, C_j)$, defined in equation (4), operates as a model-selection criterion between two competing models, estimated from the data in clusters C_i and C_j. The first model is represented by a single Gaussian distribution, estimated from the data in $C_i \cup C_j$, while the second model is represented by two Gaussians, one estimated from the data in cluster C_i and the other from the data in cluster C_j. The first model assumes that all the data are derived from a single Gaussian process and therefore belong to one speaker, while the second model assumes that the data are drawn from two different Gaussian processes, and therefore belong to two different speakers. As such, the Δ_{BIC} represents the difference between the BIC scores of both models, where the first term in equation (4) corresponds to the difference in the quality of the match between the models and the data, while the second term is a penalty for the difference in the complexities of the models, with λ allowing the tuning of the balance between the two terms. Consequently, Δ_{BIC} scores above 0.0 correspond to better modeling with one Gaussian and thus favor one speaker, while Δ_{BIC}

scores below 0.0 favor the model with two separate Gaussians and thus support the hypothesis of two speakers.

While using the Δ_{BIC} measure in the merging process of speaker clustering, those clusters that produce the biggest negative difference in terms of Δ_{BIC} among all the pair-wise combinations of clusters are joined together. The merging process is *stopped* when the lowest BIC score from among all the combinations of clusters in the current clustering is higher than a specified threshold, which in our case was set to 0.0.

The most important role in such clustering is played by the penalty term in the BIC measure, which is weighted by the open parameter λ. In the original definition of BIC the parameter λ is set to 1.0 (Schwartz, 1976), but it was found that the speaker clustering performed much better if λ is considered as an open parameter that is tuned on the development data. The λ influences both the merging and the stopping criteria and needs to be chosen carefully to have the optimum effect. To avoid this, several modifications of the above approach have been proposed, but they all had only moderate success, since they either introduced a new set of open parameters (Ajmera & Wooters, 2003) or increased the computational cost of the speaker clustering (Zhu et al., 2005).

2.3 Speaker clustering with Gaussian Mixture Models

An alternative approach, which does not rely only on the BIC measure, was introduced in (Barras et al., 2006). The main idea was to improve the initial clustering with the BIC measure by introducing another stage of agglomerative clustering with Gaussian Mixture Models (GMMs).

This approach tends to stop the initial clustering stage (the BIC stage) early, and use the results to seed a second clustering stage with more initial data per cluster. As a result, the second stage can estimate more complex models for the speakers. In (Barras et al., 2006) they modeled clusters at this stage with GMMs by using methods from speaker-recognition tasks.

In this case the initial clustering is performed using the BIC method, described in the previous section, which is then continued by introducing the GMMs in the second stage of clustering. Before clustering, a Universal Background Model (UBM) with diagonal Gaussians is built on training data to represent the general speakers. In addition, some kind of feature normalization is applied to reduce the effects of the different acoustic environments. Next, the clustering is performed using the agglomerative clustering scheme presented in Section 2.2. The clusters are represented as GMMs and a cross log-likelihood ratio (Gauvain & Lee, 1994) is used as a similarity measure. The GMM for each cluster is obtained by a MAP adaptation (Reynolds et al., 2000) of the means of the pre-trained UBM. Explicitly, for each cluster C_i its model M_i is MAP adapted from the UBM B using the feature vectors x_i belonging to that cluster. Then, the cross log-likelihood ratio between the two clusters C_i and C_j is defined as (Barras et al., 2004):

$$CLR(C_i, C_j) = \log\frac{L(x_i|M_j)}{L(x_i|B)} + \log\frac{L(x_j|M_i)}{L(x_j|B)}, \tag{5}$$

Where the $L(x|M)$ in all four cases represents the average likelihood per frame of data x, given the model M. The pair of clusters with the highest *CLR* is merged and a new model is created. The process is repeated until the highest *CLR* is below a predefined threshold, chosen from the development data.

Several refinements can be made at all the stages of the presented speaker clustering. In order to reduce the effects of different acoustic environments, different types of feature-normalization techniques have been proposed. The most common is the feature-warping technique, which aims to reshape the histogram of the feature data, derived from the cluster segments, into a Gaussian distribution (Pelecanos & Sridharan, 2001). As far as the UBM is concerned, different UBMs can be trained and used, corresponding to the different gender and channel conditions that are expected in the audio data (Barras et al., 2004). Another method is to build a new UBM directly from the processing audio data prior to the data clustering (Moraru et al., 2005). Several improvements to the similarity measure have also been proposed. In the case where several UBMs are used in the speaker clustering, the GMMs are obtained through a MAP adaptation from the gender- and channel-matched UBM, and only these models (clusters) are then compared using the *CLR* measure (Barras et al., 2006). Alternative measures to the CLR have also been tested within this approach, like an upper-bound estimation of the Kullback-Leibler measure (Do, 2003; Ramos-Castro et al., 2005) or a penalized likelihood criterion, based on the BIC (Žibert, 2006b).
We implemented this approach by applying feature-warping normalization before the clustering, while just one general UBM was used for all the MAP adaptations of the GMMs. The UBM was trained directly from the processing audio data, and the derived GMMs were represented by diagonal-covariance Gaussians with 32 mixtures. We decided to use these rather small mixture-size GMMs (in the original approach (Barras et al., 2006) 128 mixtures were used), since we did not gain any improvement in the speaker clustering on the development data by increasing the number of mixtures in the GMMs. The second reason was that by using GMMs with a rather small number of parameters, we removed the need for running the initial stage of the BIC clustering in order to obtain more initial data per cluster.

3. Including prosodic information in the speaker clustering

Both the previously presented speaker-clustering approaches perform the clustering by measuring the similarity between the speaker data, based only on the acoustic representations. These selected acoustic representations perform reliably in most speaker-recognition systems, and they were an obvious choice in speaker-clustering approaches. Lately, however, several speaker-recognition systems have attempted to include prosodic information as well as acoustics for the representation of the speakers (Kajarekar et al., 2003; Reynolds et al., 2003; Shriberg et al., 2005; Baker et al., 2005). The fusing of both representations was an attempt to reduce the need for speaker modeling in various acoustic environments and to provide additional information about the speaker's speech characteristics.
We developed an approach to speaker clustering that included both acoustic and prosodic representations of the speakers. The main objectives were to derive the prosodic features from the speaker-cluster data and to integrate them into the basic acoustic representations of the speaker. In order to achieve this, we needed to adopt the presented agglomerative speaker-clustering approach to merge the cluster data by defining a new similarity measure that was able to fuse the similarity scores from both representations. In the following sections a derivation of the prosodic features for the speaker clustering and a new speaker-clustering approach, based on these features, are presented.

3.1 Prosodic features for the speaker clustering

The development of the prosodic features for the speaker clustering was inspired by a derivation of similar features for speaker recognition (Shriberg et al., 2005), where they focused on capturing the longer-range stylistic features of a person's speaking behavior. We followed this approach by producing three groups of prosodic features, which were related to pitch, energy and duration measurements in speech signals and were designed in such a way as to be suitable for speaker clustering.

A standard approach to extracting prosodic information from speech signals is to define the basic units of speech and then produce different features from the duration, pitch and energy measurements associated with these units (Noth et al., 2002). A key question is what kind of speech units should be applied and how much data is needed for a reliable estimation of the prosodic events? When prosodic information is modeled in combination with automatic speech-recognition systems, the usual way of producing prosodic features is to use recognized words as the basic speech units (Noth et al., 2002). In this case a large amount of training data should be available, which is not the case when modeling the prosodic information of the speakers from the speaker clusters. Consequently, the basic speech units should be defined on sub-word speech regions. In (Shriberg et al., 2005) the prosodic features were extracted from the syllable-based regions of speech, while we decided to use the voiced-unvoiced (VU) regions. Using the VU regions in speaker clustering has several advantages over the syllable-based representation. Both types of sub-word units operate at nearly the same speech-region levels and thus the same techniques for computing prosodic features can be applied, but the VU regions can be detected without the use of large-vocabulary speech-recognition systems and are language independent, which is not the case when the speech units are represented by syllables or words.

The procedure for computing the prosodic features from speech segments, using the VU regions as the basic speech units, was as follows. The energy and pitch measurements were made at the frame level, which in our case was set to 10 ms. The short-term energy was computed as the log of the signal energy, i.e., as the sum of the squared speech-signal amplitudes in the window-size range, which in our case was set to 32 ms. The energy was computed using the feature-extraction tool in the *HTK Toolkit* (Young et al., 2004). The pitch (*f0*) is estimated using the *get_f0* function in the *ESPS/Waves* toolkit (Talkin, 1995) and then post-processed using the median filtering. For the detection of voiced (V), unvoiced (U) and silent (S) regions in the speech, a generic phoneme-based speech recognizer was used. The recognizer was the same as the one presented in (Žibert et al., 2007), which was already applied in a speech-detection task, where it proved to be language independent. In addition, we aligned the voiced regions with the *f0* trajectory, where the voiced regions were either shortened or extended according to the *f0* values or some missing *f0* values were added in the cases of detected voiced regions. After the extraction and alignment of these measurements, we created three groups of prosodic features related to the energy, duration and pitch values in voiced-unvoiced-silent (V-U-S) regions:

Energy features:

1. *energy mean:* the estimated mean of the short-term energy frames in the speech segment;
2. *energy variance:* the estimated variance of the short-term energy frames in the speech segment;
3. *rising energy frame rate:* the number of rising short-term energy frames in the speech segment divided by the total number of energy frames;

4. *falling energy frame rate:* the number of falling short-term energy frames in the speech segment divided by the total number of energy frames.

Duration features:

5. *normalized VU speaking rate:* the number of changes of the V, U, S units in the speech segment divided by the speech-segment duration;
6. *normalized average VU duration rate:* the absolute difference between the average duration of the voiced parts and the average duration of the unvoiced parts, divided by the average duration of all the V, U units in the speech segment;

Pitch features:

7. *f0 mean:* the estimated mean of the *f0* frames computed only in the V regions of the speech segment;
8. *f0 variance:* the estimated variance of the *f0* frames computed only in the V regions of the speech segment;
9. *rising f0 frame rate:* the number of rising *f0* frames in the speech segment divided by the total number of *f0* frames;
10. *falling f0 frame rate:* the number of falling *f0* frames in the speech segment divided by the total number of *f0* frames.

All the above features were obtained from the individual speech segments associated within each cluster. The features were designed by following the approach for prosody modeling of speaker data (Shriberg et al., 2005) and the development of the prosodic features for word-boundary detection in automatically transcribed speech data (Gallwitz et al., 2002). Note that the features in 5 and 6 are the same as those used in speech detection based on phoneme-recognition features (Žibert et al., 2007). We decided to implement only those features that can be reliably estimated from relatively short speech segments and were suitable for prosody modeling in speaker clustering. A normalization of each feature was provided by averaging the selected measurements, either by segment duration or by the total number of counted frames in a segment.

The 10 presented features were carefully designed to capture the speaker-oriented prosodic patterns from relatively short speech segments; however, to obtain reliable prosodic information about a speaker there should be several segments present in a cluster. Therefore, the above prosodic features should be treated as a supplementary representation of the cluster data, which can provide a considerable improvement in speaker-clustering performance when larger amounts of cluster data are available.

3.2 Fusing of acoustic and prosodic information in speaker clustering

The development of prosodic features represented the first step of including prosodic information in speaker clustering. The next step was to provide an appropriate comparison of the different clusters represented by this set of features and to integrate the acoustic and prosodic information into a single, unified speaker-clustering approach. We decided to implement the same speaker-clustering approach as was used in the baseline BIC clustering, presented in Section 2.2, but we extended it by including both types of information in the merging process of clustering.

The main reason for integrating the prosodic features into the speaker clustering was to provide information in addition to the basic acoustic features in order to gain some improvement in the speaker clustering in the case of adverse acoustic conditions. Thus, a new clustering approach was designed, which enabled us to control the amount of each type

of information in the merging process of speaker clustering. To achieve this, two important issues had to be resolved:

1. an appropriate similarity measure for the comparison of the clusters represented by prosodic features had to be designed;
2. a fuzzy-based merging criterion had to be defined, which should appropriately combine the similarity scores of the acoustic and prosodic representations of the clusters.

In the baseline speaker-clustering approach the BIC was applied as the similarity measure between the clusters represented by the acoustic, MFCC features. In the merging stage of the baseline clustering approach two clusters were joined, providing their Δ_{BIC} score achieved the minimum among all the Δ_{BIC} scores. A similarity measure based on the prosodic features was needed to operate in the same manner: lower scores should correspond to more similar clusters and higher scores to less similar clusters. Both similarity measures were also required to be easily integrated into the fuzzy-based merging criterion of the speaker clustering. This could be ensured by enabling the normalization of the similarity scores of both measures and by the appropriate weighting of both similarities.

Taking all this into account, a new prosodic measure was proposed. The measure was defined on speaker clusters by computing the Mahalanobis distance between the principal components of the speaker segments represented by the prosodic feature vectors. This procedure involved the following steps:

1. Each segment s_i is represented by the vector $\boldsymbol{v}_{s_i}^{pros}$ constructed from 10 prosodic features, defined in Section 3.1.
2. A Principal Component Analysis (PCA; Theodoridis & Koutroumbas, 2003) is performed on all the processing segments s_i, represented by the vectors $\boldsymbol{v}_{s_i}^{pros}$. This involves computing the correlation matrix R^{pros} of the vectors $\boldsymbol{v}_{s_i}^{pros}$ and decomposing the eigenvalue $R^{pros} = P \cdot \Lambda \cdot P^T$, where P represents the matrix of eigenvectors ordered by the eigenvalues, which are stored in the diagonal matrix Λ.
3. The Mahalanobis distance between the principal components of the speaker segments is computed:

$$d_{pros}(s_i, s_j) = \sum_{n=1}^{10} \frac{\left(w_{s_i}^n - w_{s_j}^n\right)^2}{\lambda_n}, \tag{6}$$

where $w_{s_i}^n$ is the principal component of $\boldsymbol{w}_{s_i} = \boldsymbol{P}^T \cdot \boldsymbol{v}_{s_i}^{pros}$ at the eigenvalue λ_n, n=1,…,10. $w_{s_j}^n$ is defined in a similar fashion.

4. The similarity measure between the speaker cluster C_i, composed of the speaker segments $\{ s_i \mid i = 1,\ldots, N_i\}$, and the speaker cluster C_j, composed of the speaker segments $\{ s_j \mid j = 1,\ldots, N_j\}$ is then defined as the average of the all the pair-wise combinations of segments from both clusters:

$$pros(C_i, C_j) = \frac{1}{N_i N_j} \sum_{s_i \in C_i} \sum_{s_j \in C_j} d_{pros}(s_i, s_j) \tag{7}$$

Lower scores in (7) correspond to a better similarity between the clusters represented by the corresponding prosodic features.

A development of the above prosodic measure was inspired by similar approaches of constructing distance measures on clusters with distances that are defined only on cluster samples (Theodoridis & Koutroumbas, 2003). In our case we used the Mahalanobis distance, which was computed for principal components of the prosodic features derived from the corresponding speaker segments. This was done so as to reduce the possible correlation effects of the selected prosodic features and to remove the influences of the different scalar ranges of the features on the distance computations. Note that the prosodic measure, defined in (7), operates in the same fashion as the BIC measure, defined in (4): lower scores correspond to a better similarity between clusters.

To integrate both the similarity scores into a merging criterion of speaker clustering some kind of score normalization needs to be applied to both similarity measures and the appropriate fusion scheme of joining both scores has to be defined. We decided to use the *min-max score normalization* of both similarity measures (Jain et al., 2005). The normalized version of the BIC measure from (4) was defined as:

$$norm_{\Delta_{BIC}}(C_r, C_s) = \frac{\Delta_{BIC}(C_r, C_s) - \min\limits_{C_i,C_j} \Delta_{BIC}(C_i, C_j)}{\max\limits_{C_i,C_j} \Delta_{BIC}(C_i, C_j) - \min\limits_{C_i,C_j} \Delta_{BIC}(C_i, C_j)}, \tag{8}$$

and a normalized version of the prosodic measure from (7) was defined as:

$$norm_{pros}(C_r, C_s) = \frac{pros(C_r, C_s) - \min\limits_{C_i,C_j} pros(C_i, C_j)}{\max\limits_{C_i,C_j} pros(C_i, C_j) - \min\limits_{C_i,C_j} pros(C_i, C_j)}. \tag{9}$$

The minimum and maximum values in equations (8) and (9) were computed from among all the pair-wise cluster combinations at the current step of merging. A controllable fusion of both representations of the speaker clusters in the merging criterion was obtained by producing a weighted sum of the normalized versions of both similarity measures:

$$fus(C_r, C_s) = \alpha \cdot norm_{\Delta_{BIC}}(C_r, C_s) + (1 - \alpha) \cdot norm_{pros}(C_r, C_s), \tag{10}$$

where α represents a weighting factor between the acoustic and prosodic representations of the speaker clusters. A merging of the clusters was then achieved by finding a minimum score among all the pair-wise combinations of clusters at the current step of clustering:

$$fus(C_i, C_j) = \min_{C_r,C_s} fus(C_r, C_s). \tag{11}$$

By using the above merging criterion the speaker clustering was performed by following the same clustering procedure as described in Section 2.2. The only difference was in step 2 of the procedure, where instead of a minimum of the Δ_{BIC} score in the merging step in equation (1), a minimum from among the fusion of scores from equation (11) was used. In this way we were able to include prosodic information in the baseline speaker-clustering approach.

4. Evaluation of the speaker-clustering approaches

An evaluation of all three presented clustering approaches was performed in two speaker-diarization tasks on broadcast news data. The evaluation experiments were conducted by

following the *NIST Rich Transcription Evaluation,* which has been the major evaluation technique for the speaker diarization of broadcast news data (Fiscus et al., 2004). A similar evaluation was also performed in the *ESTER Evaluation* using French radio broadcast news data (Galliano et al., 2005).
Our experiments were carried out on two broadcast news databases. The first includes 33 hours of BN shows in Slovene and is called the SiBN database (Žibert & Mihelič, 2004). The second was a multilingual speech database, COST278, which is composed of 30 hours of BN shows in nine European languages (Vandecatseye et al., 2004), and was already used for an evaluation of different language- and data-independent procedures in the processing of audio BN shows, (Žibert et al., 2005).

4.1 Evaluation measure

The speaker-clustering approaches were evaluated by measuring the speaker-diarization performance in terms of the diarization error rate (DER), (Fiscus et al., 2004). The DER measures the differences in the reference and hypothesized speaker segmentations. This is accomplished by finding a one-to-one mapping of the reference speaker segments to the hypothesis speaker labels so as to maximize the total overlap of the reference and the (corresponding) mapped hypothesis speakers. The speaker-diarization performance is then expressed in terms of the miss (speaker in reference but not in hypothesis), false-alarm (speaker in hypothesis but not in reference) and speaker-match (mapped reference speaker is not the same as the hypothesized one) error rates. While the miss and false-alarm error rates correspond to the speech/non-speech detection errors, the speaker-match error rate corresponds to the speaker-clustering errors. The overall diarization error (DER) is the sum of these three components.
We additionally modified the DER measure in order to more closely analyze the performance of the speaker-clustering approaches, regardless of the stopping criteria used in the clustering. We achieved this by computing the overall diarization-error-rate trajectory as the average of the DER trajectories of each processed audio file. The DER trajectory per each file was constituted from the DER values computed for a different number of speaker clusters. The number of speaker clusters was not defined in absolute figures, but as the relative difference compared to the actual number of speakers in each processed audio file. This enabled us to align the DER values of each file at the same evaluation points and produce the average trajectory as the final result. Such evaluations provided us with more valuable insights into how the different speaker-data representations could affect the speaker clustering and how well the merging process of clustering can be performed without any additional tuning of the proper stopping thresholds, since it is well known that an improper selection of the stopping thresholds can seriously degrade the speaker-clustering performance.

4.2 Experimental setup

We evaluated all three speaker-clustering approaches: a baseline system with the BIC, a GMM-based approach and a fusion-based approach, presented in Sections 2.2, 2.3, and Section 3, respectively.
Since we only wanted to assess the performance of the speaker-clustering approaches we used the same speech/non-speech-detection and audio-segmentation procedures in all the evaluation experiments. The speech/non-speech detection used the approach presented in (Žibert et al., 2007), while the audio segmentation used the approach presented in (Chen & Gopalakrishnan, 1999).

In all the tested speaker-clustering approaches we needed to set different open parameters. The parameters were chosen according to the optimal speaker-diarization performance of the corresponding clustering approaches on the development dataset, which was composed of 7 hours of BN audio data from the SiBN database. Detailed information of the experimental setup for each individual clustering approach is presented in the following list:

- **The baseline BIC approach:** (described in Section 2.2)
  The audio data were represented by *MFCC* features, which were composed of the first 12 cepstral coefficients (without the 0th coefficient) and a short-term energy with the addition of the Δ*MFCC* features. The Δ*MFCC* features were computed by estimating the first-order regression coefficients from the static *MFCC* features. The features were derived from audio signals every 10 ms by using 32-ms analysis windows, (Young et al., 2004). For the estimations of the Δ_{BIC} measure from equation (4) each cluster was modeled using full-covariance Gaussian distributions, and the penalty factor λ was set to 3.0, which was chosen according to the optimal clustering performance on the development dataset.
  This approach is referred to as the **clust_REF_BIC** approach in our experiments.
- **The UBM-MAP-CLR approach:** (described in Section 2.3)
  The audio data were represented by the same feature set as was used in the baseline BIC approach, but with the addition of feature warping (Pelecanos & Sridharan, 2001), which was performed on each segment separately. All the GMMs were constructed from 32 diagonal-covariance Gaussian mixtures. The UBM was estimated directly from the processing audio data by using the expectation-maximization algorithm (Theodoridis & Koutroumbas, 2003). No separate gender-derived models were trained. The MAP adaptation of (only) the UBM means was performed on each cluster to derive cluster-based GMMs. Next, the clusters where the highest CLR score in equation (5) was achieved were merged at each step of the merging process.
  This approach is referred to as the **clust_UBM_MAP_CLR** approach in our experiments.
- **The FUSION approach:** (described in Sections 3.1–3.2)
  The fusion of acoustic and prosodic representations is described by equation (10). The acoustic representation of the audio data was implemented by the same MFCC-based features as were used in the above approaches. The prosodic features were derived at every speaker segment and were not changed during the clustering. When combining the Δ_{BIC} measure from equation (8) and the prosodic measure from equation (9) into the weighted sum (10), the weighting parameter α needed to be set. This parameter was tuned on the development dataset and set to a value of 0.85. This was in accordance with our expectation that the main discriminative information for speaker clustering is stored in the acoustics, while the prosody provides only supplementary information. Note that we used the same penalty factor, λ=3.0, in the Δ_{BIC} measure as was used in the baseline BIC approach.
  This approach is referred to as the **clust_FUSION** approach in our experiments.

4.3 Evaluation results

An assessment of the selected clustering approaches was performed on the SiBN and the COST278 BN databases. The experiments were conducted in such a way as to evaluate the performance of the clustering approaches in various acoustic conditions. The SiBN database

consists of BN shows of one TV station, including the same set of speakers, and was collected in unchanged recording conditions. For this reason it was considered to represent relatively homogeneous data. On the other hand, the COST278 BN database consists of BN shows in different languages from several TV and radio stations, it includes a wide range of speakers and the data were collected under different recording conditions. As such it represented relatively inhomogeneous audio data in terms of different speakers and acoustic environments.

The speaker-diarization results, which were produced by running all three speaker-clustering approaches on the SiBN and COST278 BN databases, are shown in Figures 2 and 3, respectively. The DER results, plotted in Figures 2 and 3, should be interpreted as follows: the DER results at the evaluation point 0 correspond to the average of the DER across all the evaluated audio files, where the number of clusters is equal to the actual number of speakers in each file, the DER results at evaluation point +5 correspond to the average of the DER across all the evaluated audio files, where the number of clusters exceeds the actual number of speakers in each file by 5, and analogously, the DER results at evaluation point -5 correspond to the average of the DER across all the evaluated audio files, where the number of clusters is 5 clusters lower than the actual number of speakers in each file, and so on.

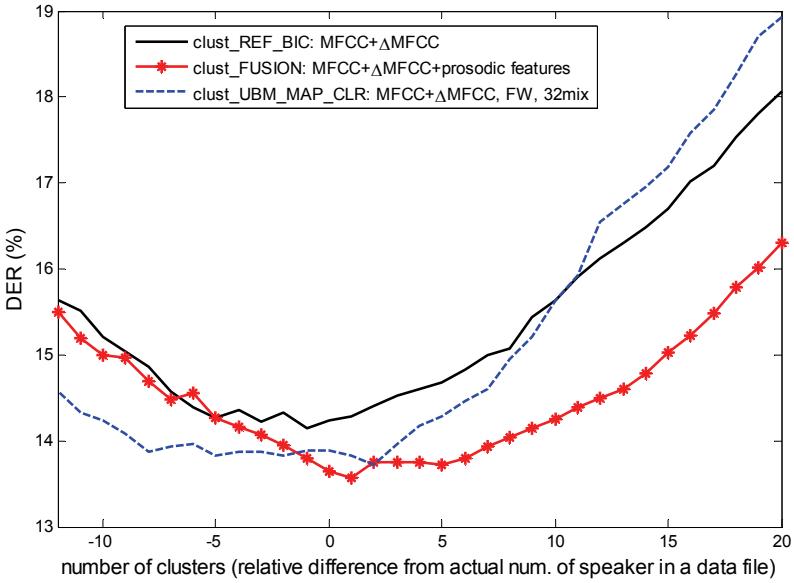


Fig. 2. Speaker-diarization results on the SiBN database when using different clustering procedures. The lower DER values correspond to better performance

The speaker-diarization results in Figure 2 correspond to the speaker-clustering performance of the tested approaches on the SiBN data. The overall performance of the speaker-clustering approaches varies between 13.5% and 16%, measured using the overall DER. The *clust_UBM_MAP_CLR* and *clust_FUSION* approaches perform slightly better than the baseline *clust_REF_BIC* approach across the whole range of evaluation points. When the

clust_FUSION and the *clust_REF_BIC* approaches are compared, it is clear that the SiBN results display significant differences in the speaker diarization performance of both approaches, which is in favor of the *clust_FUSION* approach. This indicates that adding the prosodic characteristics of speakers to the basic acoustic information could improve the speaker clustering. The same can be concluded from comparing the *clust_UBM_MAP_CLR* approach with the baseline BIC approach. The performance of the *clust_UBM_MAP_CLR* approach improved when enough clustering data were available for the GMM estimations, which resulted in lower DERs in comparison to the baseline BIC approach, when the number of clusters shrinks (the DER results display a better performance for the *clust_UBM_MAP_CLR* approach in the range below the evaluation point +10 in Figure 2).
It is also interesting to note that the DER trajectories of all the approaches achieved their minimum DER values around the evaluation point 0. This means that if all the clustering approaches were to be stopped when the number of clusters is equal to the number of actual speakers in the data, all the approaches would exhibit their optimum speaker-diarization performance. At that point the best clustering result was achieved with the *clust_FUSION* approach.

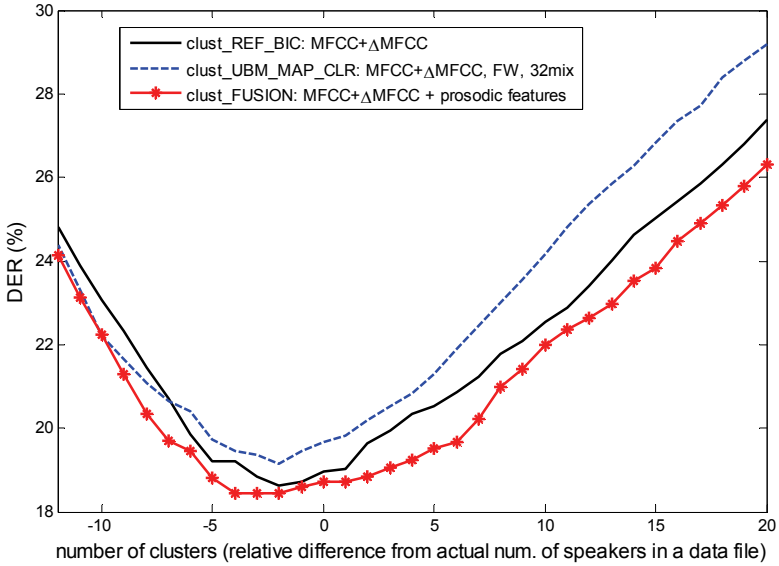


Fig. 3. Speaker-diarization results on the COST278 BN database when using different clustering procedures. The lower DER values correspond to a better performance.

Another interesting conclusion can be drawn from observing the flatness of the DER trajectories. Since the proposed evaluation measure aimed to compute the DER values at the relative numbers of clusters in each file, no stopping criteria needed to be applied; however, in practice the proper stopping of the clustering should be ensured. The optimum stopping criteria should end the merging process at the point with the lowest DER, which should coincide with the evaluation point 0, where the number of clusters is equal to the number of actual speakers in the data. Around this point it is better for the approaches to produce

relatively flat DER trajectories, which would result in a small loss of speaker-diarization performance, when the stopping criteria would not find the exact position for ending the merging process. In the case of the SiBN results, the DER trajectory, produced by the *clust_FUSION* approach, is flatter around the evaluation point 0 than the DER trajectories, produced by the *clust_REF_BIC* and *clust_UBM_MAP_CLR* approaches.
The speaker-diarization results in Figure 3 were produced by running the tested clustering approaches on the COST278 BN database. The results demonstrate the similar clustering performance of the approaches as in the case of the SiBN data, even though the overall DERs are higher than in the SiBN case. This was expected, since the COST278 BN data includes many more speakers in various acoustic environments than the SiBN data, and thus the clustering problem was more complex. In this situation the *clust_FUSION* approach produced the best overall speaker-diarization results, while the *clust_REF_BIC* approach performed slightly better than the *clust_UBM_MAP_CLR* approach. This means that in the case of adverse acoustic conditions it is better to model the cluster data by adding prosodic information to the cluster representations rather than modeling them just with acoustic representations (the *clust_REF_BIC* approach) or by a more precise acoustic modeling with the GMMs (the *clust_UBM_MAP_CLR* approach).

5. Discussion

In short, we have looked at three speaker-clustering approaches. The first was a standard approach using a bottom-up agglomerative clustering principle with the BIC as a merging criterion. In the second system an alternative approach was applied, also using bottom-up clustering, but the representations of the speaker clusters and the merging criteria are different. In this approach the speaker clusters were modeled by GMMs. In the clustering procedure during the merging process the universal background model was transformed into speaker-cluster GMMs using the MAP adaptation technique. The merging criterion in this case was a cross log-likelihood ratio (CLR). A totally new approach was developed within the fusion speaker-clustering system, where the speaker segments are modeled by acoustic and prosodic representations. The idea was to additionally model the speaker's prosodic characteristics and add them to the basic acoustic information. We constructed 10 basic prosodic features derived from the energy of the audio signals, the estimated the pitch contours, and the recognized voiced-unvoiced regions in the speech, which represented the basic speech units. By adding prosodic information to the basic acoustic features the baseline clustering procedure had to be changed to work with the fusion of both representations.
We performed two evaluation experiments where the overall diarization error rate was used as an assessment measure for the three tested clustering approaches. Experiments were performed on the SiBN and the COST278 BN databases. The evaluation results showed better performance for the tested systems in the SiBN case. This is due to the fact that the SiBN data included more homogeneous audio segments than the COST278 data, which resulted in an about 5% better performance for all of the clustering approaches. Furthermore, it was shown that speaker clustering, where the segments are modeled by speaker-oriented representations (speaker GMMs, prosodic features), were more stable and more reliable than the baseline system, where the segments are represented just by

acoustic information. The best overall results were achieved with the fusion system, where the clustering involved joining the acoustic and prosodic features. From this it can be concluded that the proposed fusion approach aimed at improving the speaker-diarization performance, especially in the case of processing BN data, where the speaker's speech characteristics across one BN show do not change significantly, but the speaker's clustering data can be biased due to different acoustic environments or background conditions.

6. Conclusion

Speaker clustering represents the last step in the speaker-diarization process. While the aim of the speech detection and speaker- and acoustic-segmentation procedures is to provide the proper segmentation of audio data streams, the purpose of the speaker clustering is to connect together segments that belong to the same speakers. In this chapter we solved this problem by applying agglomerative clustering methods. We concentrated on developing proper representations of the speaker segments for clustering and researched different similarity measures for joining the speaker segments that would result in a minimization of the overall diarization error for such systems. We realized three speaker-clustering systems, two of them operated on acoustic representations of speech, while the newly proposed one was designed to include prosodic information in addition to the basic acoustic representations. In this way we were able to impose higher-level information in the representations of the speaker segments, which led to improved clustering of the segments in the case of similar speaker acoustic characteristics in adverse acoustic conditions.

7. Acknowledgment

This work was supported by Slovenian Research Agency (ARRS), development project M2-0210 (C) entitled "AvID: Audiovisual speaker identification and emotion detection for secure communications."

8. References

Ajmera, J. & Wooters, C. (2003). A Robust Speaker Clustering Algorithm, *Proceedings of IEEE ASRU Workshop*, pp. 411-416, St. Thomas, U.S. Virgin Islands, November 2003.

Anastasakos, T.; McDonough, J.; Schwartz, R.; & Makhoul J. (1996) A Compact Model for Speaker-Adaptive Training, *Proceedings of International Conference on Spoken Language Processing (ICSLP1996),* pp. 1137-1140, Philadelphia, PA, USA, 1996.

Baker, B.; Vogt, R. & Sridharan, S. (2005). Gaussian Mixture Modelling of Broad Phonetic and Syllabic Events for Text-Independent Speaker Verification, *Proceedings of Interspeech 2005 - Eurospeech*, Lisbon, Portugal, September 2005.

Barras, C.; Zhu, X.; Meignier, S. & Gauvain, J.-L. (2004). Improving Speaker Diarization, *Proceedings of DARPA Rich Transcription Workshop 2004,* Palisades, NY, USA, November, 2004.

Barras, C.; Zhu, X.; Meignier, S. & Gauvain, J.-L. (2006). Multistage Speaker Diarization of Broadcast News. *IEEE Transactions on Speech, Audio and Language Processing, Special Issue on Rich Transcription*, Vol. 14, No. 5, (September 2006), pp. 1505-1512.

Beyerlein, P.; Aubert, X.; Haeb-Umbach, R.; Harris, M.; Klakow, D.; Wendemuth, A.; Molau, S.; Ney, H.; Pitz, M. & Sixtus, A. (2002). Large vocabulary continuous speech recognition of Broadcast News – The Philips/RWTH approach. *Speech Communication*, Vol. 37, No. 1-2, (May 2002), pp. 109-131.

Chen, S. S. & Gopalakrishnan, P. S. (1998). Speaker, environment and channel change detection and clustering via the Bayesian information criterion, *Proceedings of the DARPA Speech Recognition Workshop*, pp. 127-132, Lansdowne, Virginia, USA, February, 1998.

Delacourt, P.; Bonastre, J.; Fredouille, C.; Merlin, T. & Wellekens, C. (2000). A Speaker Tracking System Based on Speaker Turn Detection for NIST Evaluation, *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP2000)*, Istanbul, Turkey, June, 2006.

Do, M. N. (2003). Fast Approximation of Kullback-Lebler Distance for Dependence Trees and Hidden Markov Models. *Signal Processing Letters*, Vol. 10, (2003), pp. 115-118.

Fiscus, J. G.; Garofolo, J. S.; Le, A.; Martin, A. F.; Pallett, D. S.; Przybocki M. A. & Sanders, G. (2004). Results of the Fall 2004 STT and MDE Evaluation, *Proceedings of the Fall 2004 Rich Transcription Workshop*, Palisades, NY, USA, November, 2004.

Galliano, S.; Geoffrois, E.; Mostefa, D.; Choukri, K.; Bonastre, J.-F. & Gravier, G. (2005). The ESTER phase II evaluation campaign of rich transcription of French broadcast news, *Proceedings of Interspeech 2005 - Eurospeech*, pp. 1149-1152, Lisbon, Portugal, September, 2005.

Gallwitz, F.; Niemann, H.; Noth, E. & Warnke, V. (2002). Integrated recognition of words and prosodic phrase boundaries. *Speech Communication*, Vol. 36, No. 1-2, January 2002, pp. 81–95.

Gauvain, J. L.; & Lee, C. H. (1994). Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech Audio Processing*, Vol. 2, No. 2, (April 1994), pp. 291-298.

Gauvain, J. L.; Lamel, L. & Adda, G. (2002). The LIMSI Broadcast News transcription system. *Speech Communication*, Vol. 37, No. 1-2, (May 2002), pp. 89-108.

Istrate, D.; Scheffer, N.; Fredouille, C. & Bonastre, J.-F. (2005). Broadcast News Speaker Tracking for ESTER 2005 Campaign, *Proceedings of Interspeech 2005 - Eurospeech*, pp. 2445-2448, Lisbon, Portugal, September, 2005.

Jain, A.; Nandakumar, K. & Ross, A. (2005). Score normalization in multimodal biometric systems. *Pattern Recognition*, Vol. 38, No. 12, (December 2005), pp. 2270-2285.

Kajarekar, S.; Ferrer, L.; Venkataraman, A.; Sonmez, K., Shriberg, E.; Stolcke, A. & Gadde, R.R. (2003). Speaker Recognition Using Prosodic and Lexical Features, *Proceedings of IEEE ASRU Workshop,* St. Thomas, U.S. Virgin Islands, November 2003.

Makhoul, J.; Kubala, F.; Leek, T.; Liu, D.; Nguyen, L.; Schwartz, R. & Srivastava, A. (2000). Speech and language technologies for audio indexing and retrieval. *Proceedings of the IEEE*, Vol. 88, No. 8, (2000) pp. 1338-1353.

Matsoukas, S.; Schwartz, R.; Jin, H. & Nguyen, L. (1997). Practical Implementations of Speaker-Adaptive Training, *Proceedings of the 1997 DARPA Speech Recognition Workshop*, Chantilly VA, USA, February, 1997.

Meignier, S.; Bonastre, J.-F.; Fredouille, C. & Merlin T. (2000). Evolutive HMM for Multi-Speaker Tracking System, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Istanbul, Turkey, June 2000.

Moh, Y.; Nguyen, P. & Junqua, J.-C. (2003). Towards Domain Independent Clustering, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 85-88, Hong Kong, April 2003.

Moraru, D.; Ben, M. & Gravier, G. (2005). Experiments on speaker tracking and segmentation in radio broadcast news, *Proceedings of Interspeech 2005 - Eurospeech*, Lisbon, Portugal, September 2005.

Nedic, B.; Gravier, G.; Kharroubi, J.; Chollet, G.; Petrovska, D.; Durou, G.; Bimbot, F.; Blouet, R.; Seck, M.; Bonastre, J.-F.; Fredouille, C.; Merlin, T.; Magrin-Chagnolleau, I.; Pigeon, S.; Verlinde, P. & Cernocky J. (1999). The Elisa'99 Speaker Recognition and Tracking Systems, *Proceedings of IEEE Workshop on Automatic Advanced Technologies*, 1999.

Noth, E.; Batliner, A.; Warnke, V.; Haas, J.; Boros, M.; Buckow, J.; Huber, R.; Gallwitz, F.; Nutt, M. & Niemann, H. (2002). On the use of prosody in automatic dialogue understanding. *Speech Communication*, Vol. 36, No. 1-2, January 2002, pp. 45–62.

Pelecanos, J. & Sridharan, S. (2001). Feature warping for robust speaker verification. *Proceedings of Speaker Odyssey*, pp. 213–218, Crete, Greece, June 2001.

Picone, J. W. (1993). Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, Vol. 81, No. 9, (1993) pp. 1215-1247.

Ramos-Castro, D,; Garcia-Romero, D.; Lopez-Moreno, I. & Gonzalez-Rodriguez, J. (2005). Speaker verification using fast adaptive TNORM based Kullback-Leibler divergence, *Third COST 275 Workshop: Biometrics on the Internet*, University of Hertfordshire, Great Britain, October, 2005.

Reynolds, D. A.; Quatieri, T. F. & and R. B. Dunn, R. B. (2000). Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, Vol. 10, No. 1, January 2000, pp. 19–41.

Reynolds, D. A.; Campbell, J. P.; Campbell, W. M.; Dunn, R. B.; Gleason, T. P.; Jones, D. A.; Quatieri, T. F.; Quillen, C.B.; Sturim, D. E. & Torres-Carrasquillo, P. A. (2003). Beyond Cepstra: Exploiting High-Level Information in Speaker Recognition, *Proceedings of the Workshop on Multimodal User Authentication*, pp. 223-229, Santa Barbara, California, USA, December, 2003.

Reynolds, D. A. & Torres-Carrasquillo, P. (2004). The MIT Lincoln Laboratory RT-04F Diarization Systems: Applications to Broadcast Audio and Telephone Conversations, Proceedings of the Fall 2004 Rich Transcription Workshop. Palisades, NY, USA, November, 2004.

Schwartz, G. (1976). Estimating the Dimension of a Model. *Annals of Statistics*, Vol. 6, pp. 461-464.

Shriberg, E.; Ferrer, L.; Kajarekar, S.; Venkataraman, A. & Stolcke, A. (2005). Modeling prosodic feature sequences for speaker recognition. *Speech Communication*, Vol. 46, No. 3-4, (July 2005), pp. 455--472.

Sinha, R.; Tranter, S. E.; Gales, M. J. F. & Woodland, P. C. (2005). The Cambridge University March 2005 Speaker Diarisation System, *Proceedings of Interspeech 2005 - Eurospeech*, pp. 2437-2440, Lisbon, Portugal, September, 2005.

Talkin, D. (1995). A robust algorithm for pitch tracking (RAPT). In: *Speech Coding and Synthesis*. W. B. Kleijn & K. K. Paliwal, (Eds.), Elsevier Science, 1995.

Theodoridis, S. & Koutroumbas, K. (2003). *Pattern Recognition, second edition*. Academic Press, ISBN 0-12-685875-6, Elsevier, USA.

Tranter, S. & Reynolds, D. (2006). An Overview of Automatic Speaker Diarisation Systems. *IEEE Transactions on Speech, Audio and Language Processing, Special Issue on Rich Transcription*, Vol. 14, No. 5, (September 2006), pp. 1557-1565.

Tritschler, A. & Gopinath, R. (1999). Improved speaker segmentation and segments clustering using the Bayesian information criterion, *Proceedings of EUROSPEECH 99*, pp. 679-682, Budapest, Hungary, September, 1999.

Vandecatseye, A.; Martens, J.-P.; Neto J.; Meinedo, H.; Garcia-Mateo, C; Dieguez, J.; Zibert, J.; Mihelic, F.; Nouza, J.; David, P.; Pleva M.; Cizmar, A.; Papageorgiou, H.; Alexandris, C.; & Mihelič, F. (2004). The COST278 pan-European Broadcast News Database, *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2004)*, pp. 873-876, Lisbon, Portugal, May 2004.

Woodland, P. C. (2002). The development of the HTK Broadcast News transcription system: An overview. *Speech Communication*, Vol. 37, No. 1-2, (May 2002), pp. 47--67.

Žibert, J. & Mihelič, F. (2004). Development of Slovenian Broadcast News Speech Database, *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2004)*, pp. 2095-2098, Lisbon, Portugal, May 2004.

Žibert, J.; Mihelič, F.; Martens, J.-P.; Meinedo, H.; Neto, J.; Docio, L.; Garcia-Mateo, C.; David, P.; Zdansky, J.; Pleva, M.; Cizmar, A.; Žgank, A.; Kačič, Z.; Teleki, C. & Vicsi, K. (2005). The COST278 Broadcast News Segmentation and Speaker Clustering Evaluation - Overview, Methodology, Systems, Results, *Proceedings of Interspeech 2005 - Eurospeech*, pp. 629--632, Lisbon, Portugal, September, 2005.

Žibert, J.; Pavešić, N. & Mihelič, F. (2006a). Speech/Non-Speech Segmentation Based on Phoneme Recognition Features. *EURASIP Journal on Applied Signal Processing*, Vol. 2006, No. 6, Article ID 90495, pp. 1-13.

Žibert, J. (2006b).*Obdelava zvočnih posnetkov informativnih oddaj z uporabo govornih tehnologij, PhD thesis (in Slovenian language),* Faculty of Electrical Engineering, University of Ljubljana, Slovenia.

Žibert, J.; Vesnicer, B. & Mihelič, F. (2007). Novel Approaches to Speech Detection in the Processing of Continuous Audio Streams. In: *Robust Speech Recognition and Understanding*, M. Grimm and K. Kroschel, (Eds.), 23-48, I-Tech Education and Publishing, ISBN 978-3-902613-08-0, Croatia.

Zhou, B. & Hansen, J. (2000). Unsupervised Audio Stream Segmentation and Clustering via the Bayesian Information Criterion, *Proceedings of International Conference on Spoken Language Processing (ICSLP 2000)*, pp. 714-717, Beijing, China, October, 2000.

Zhu, X.; Barras, C.; Meignier, S. & Gauvain, J.-L. (2005). Combining Speaker Identification and BIC for Speaker Diarization, *Proceedings of Interspeech 2005 - Eurospeech*, pp. 2441-2444, Lisbon, Portugal, September, 2005.

Young, S.; Evermann, G.; Gales, M.; Hain, T.; Kershaw, D.; Moore, G.; Odell, J.; Ollason, D.; Povey, D.; Valtchev, V. & Woodland, P. C. (2004). *The HTK Book (for HTK Version 3.2)*, Cambridge University Engineering Department, Cambridge, United Kingdom.

20

Gender Classification in Emotional Speech

Mohammad Hossein Sedaaghi
Sahand University of Technology
Iran

1. Introduction

The emotion accompanying with the voice is considered as a salient aspect in human communication. The effects of emotion in speech tend to alter the voice quality, timing, pitch and articulation of the speech signal. Gender classification, on the other hand, is an interesting field for psychologists to foster human-technology relationships. Automatic gender classification take on an increasingly ubiquitous role in myriad of applications, e.g., demographic data collection. An automatic gender classifier assists the development of improved male and female voice synthesizers (Childers et. al., 1988). Gender classification is also used to improve the speaker clustering task which is useful in speaker recognition. By separately clustering each gender class, the search space is reduced when evaluating the proposed hierarchical agglomerative clustering algorithm (Tranter and Reynolds, 2006). It also avoids segments having opposite gender tags being erroneously clustered together. Gender information is time-invariant, phoneme-independent, and identity-independent for speakers of the same gender (Wu & Childers, 1991). In (Xiaofan & Simske, 2004), an accent classification method is introduced on the top of gender classification. Vergin et al. (Vergin, 1996) claim that the use of gender-dependent acoustic-phonetic models reduces the word error rate of the baseline speech recognition system by 1.6%. In (Harb & Chen, 2005), a set of acoustic and pitch features along with different classifiers is tested for gender identification. The fusion of features and classifiers is shown to perform better than any individual classifier. A gender classification system is proposed in (Zeng et. al., 2006) based on Gaussian mixture models of speech features. Metze et al. have compared four approaches for age and gender recognition using telephone speech (Metze et. al., 2007). Gender cues elicited from the speech signal are useful in content-based multimedia indexing as well (Harb & Chen, 2005). Gender-dependent speech emotion recognizers have been shown to perform better than gender-independent ones for five emotional state (Ververidis & Kotropoulos, 2004; Lin & Wei, 2005) in DES (Engberg & Hansen, 1996). However, gender information is taken for granted there. The most closely related work to the present one is related to the research by Xiao et al. (Xiao et. al., 2007), where gender classification was incorporated in emotional speech recognition system using a wrapper approach based on back-propagation neural networks with sequential forward selection. An accuracy of 94.65% was reported for gender classification on the Berlin dataset (Burkhardt et. al., 2005).

In this research, we employ several classifiers and assess their performance in gender classification by processing utterances from DES (Engberg & Hansen, 1996), SES (Sedaaghi, 2008) and GES (Burkhardt et. al., 2005) databases. They all contain affective speech. In

particular, we test the Bayes classifier with sequential floating forward feature selection (SFFS) (Fukunaga & Narendra, 1975; Pudil et. al., 1994), the probabilistic neural networks (Specht, 1990), the support vector machines (Vapnik, 1998), and the *K*-nearest neighbor classifiers (Fix & Hodges, 1991-a; Fix & Hodges, 1991-b). Although techniques based on hidden Markov models could be applied for gender classification in principle, they are not included in this study, because temporal information is ignored.

2. Database

The first dataset stems from Danish Emotional Speech (DES) database, which is publicly available and well annotated (Engberg & Hansen, 1996). The recordings in DES include utterances expressed by two professional actors and two actresses in five different emotional states (anger, happiness, neutral, sadness, and surprise). The utterances correspond to isolated words, sentences, and paragraphs. The complete database comprise approximately 30 minutes of speech.

Sahand Emotional Speech (SES) database (Sedaaghi, 2008) comprise utterances expressed by five male and five female students in five emotional states similar to the emotions employed in DES. Twenty four words, short sentences and paragraphs spoken in Farsi by each student are included in SES database leading to 1200 utterances and about 50 minutes recording.

As the third database, the database of German Emotional Speech (GES) is investigated. An emotional database comprising 6 basic emotions (anger, joy, sadness, fear, disgust and boredom) as well as neutral speech is recorded (Burkhardt et. al., 2005). Ten professional native German actors (5 female and 5 male) have simulated these emotions, producing 10 utterances (5 short and 5 longer sentences). The recorded speech material of about 800 sentences have been evaluated with respect to recognizability and naturalness in a forced-choice automated listening-test by 20-30 judges. Those utterances for which the emotion is recognized by at least 80% of the listeners are used for further analysis (i.e., 535 sentences) (Burkhardt et. al., 2005).

3. Feature extraction

The automatic gender classification is mainly achieved based on the average value of the fundamental frequency (i.e., F0). Also, the distinction between men and women have been represented by the location in the frequency domain of the first 3 formants for vowels (Peterson & Barney, 1952). To improve the efficiency, more features should be considered. The statistical features employed in our study are grouped in several classes and have been demonstrated in Table 1. They have been adopted from (Ververidis & Kotropoulos, 2006).

	Formant features
1-4	Mean value of the first, second, third, and fourth formant.
5-8	Maximum value of the first, second, third, and fourth formant.
9-12	Minimum value of the first, second, third, and fourth formant.
13-16	Variance of the first, second, third, and fourth formant.
	Pitch features
17-21	Maximum, minimum, mean, median, interquartile range of pitch values.
22	Pitch existence in the utterance expressed in percentage (0-100%).

23-26	Maximum, mean, median, interquartile range of durations for the plateaux at minima.
27-29	Mean, median, interquartile range of pitch values for the plateaux at minima.
30-34	Maximum, mean, median, interquartile range, upper limit (90%) of durations for the plateaux at maxima.
35-37	Mean, median, interquartile range of the pitch values within the plateaux at maxima.
38-41	Maximum, mean, median, interquartile range of durations of the rising slopes of pitch contours.
42-44	Mean, median, interquartile range of the pitch values within the rising slopes of pitch contours.
45-48	Maximum, mean, median, interquartile range of durations of the falling slopes of pitch contours.
49-51	Mean, median, interquartile range of the pitch values within the falling slopes of pitch contours.
	Intensity (Energy) features
52-56	Maximum, minimum, mean, median, interquartile range of energy values.
57-60	Maximum, mean, median, interquartile range of durations for the plateaux at minima.
61-63	Mean, median, interquartile range of energy values for the plateaux at minima.
64-68	Maximum, mean, median, interquartile range, upper limit (90%) of duration for the plateaux at maxima.
69-71	Mean, median, interquartile range of the energy values within the plateaux at maxima.
72-75	Maximum, mean, median, interquartile range of durations of the rising slopes of energy contours.
76-78	Mean, median, interquartile range of the energy values within the rising slopes of energy contours.
79-82	Maximum, mean, median, interquartile range of durations of the falling slopes of energy contours.
83-85	Mean, median, interquartile range of the energy values within the falling slopes of energy contours.
	Spectral features
86-93	Energy below 250, 600, 1000, 1500, 2100, 2800, 3500, 3950 Hz.
94-100	Energy in the frequency bands 250-600, 600-1000, 1000-1500, 1500-2100, 2100-2800, 2800-3500, 3500-3950 Hz.
101-106	Energy in the frequency bands 250-1000, 600-1500, 1000-2100, 1500-2800, 2100-3500, 2800-3950 Hz.
107-111	Energy in the frequency bands 250-1500, 600-2100, 1000-2800, 1500-3500, 2100-3950 Hz.
112-113	Energy ratio between the frequency bands (3950-2100) and (2100-0) and between the frequency bands (2100-1000) and (1000-0).

Table 1. List of extracted features adopted from (Ververidis & Kotropoulos, 2006).

Not all the features can be extracted from each utterance. For example, some pitch contours do not have plateaux below 45% of their maximum pitch value, or some utterances do not have pitch at all because they are unvoiced. When a large number of missing feature values is met, the corresponding feature is discarded. The features with NaN (not a number) values are replaced with the mean value of the corresponding feature. The outliers (features with value 10000 times greater or smaller than the median value) are then eliminated. Also the features with bias are investigated. Then all features are normalized. The discarded features are as follows.

- DES: 8, 17-51, 57-85, 105 (47 features remained),
- SES: 8, 23-29, 33-34, 41, 48, 57-63, 67, 75, 82, 94, 96, 98, 103-105, 109-113 (80 features preserved),
- GES: 8, 23-29, 33-34, 41, 60, 67, 75, 82, 94, 96, 98-99, 103-107, 109-113 (84 features retained).

4. Classifiers

The output of the gender classifier on emotional speech is a prediction value (label) of the actual speaker's gender. In order to evaluate the performance of a classifier, the repeated s-fold cross-validation method is used. According to this method if s=20, the utterances in the data collection are divided into a training set containing 80% of the available data and a disjoint test set containing the remaining 20% of the data. The procedure is repeated for s=20 times. The training and the test set are selected randomly. The classifier is trained using the training set and the classification error is estimated on the test set. The estimated classification error is the average classification error over all repetitions (Efron & Tibshirani , 1993).

The following classifiers have been investigated:

1. Naive Bayes classifier using the SFFS feature selection method (Pudil et. al., 1994). The SFFS consists of a forward (inclusion) step and a conditional backward (exclusion) step that partially avoids local optima. In the proposed method, feature selection is used in order to determine a set of 20 features that yields the lowest prediction error for a fixed number of cross-validation repetitions. Ten best sorted features among the 20 best selected features are as follows.
   - 10 best features for DES: {112, 15, 10, 107, 96, 52, 102, 14, 13, 99},
   - 10 best features for SES: {6, 32, 51, 3, 76, 20, 44, 52, 17, 22},
   - 10 best features for GES: {38, 69, 43, 80, 42, 40, 63, 8, 15, 6}.
2. Probabilistic Neural Networks (PNNs) (Specht, 1990). PNNs are a kind of radial basis function (RBF) networks suitable for classification problems. A PNN employs an input, a hidden, and an output layer. The input nodes forward the values admitted by patterns to the hidden layer ones. The hidden layer nodes are as many as the input nodes. They are simply RBFs that nonlinearly transform pattern values to activations. The nodes at the output layer are as many as the classes. Each node sums the activation values weighted possibly by proper weights. The input pattern is finally classified to the class associated to the output node whose value is maximum. PNNs with a spread parameter equal to 0.1 are found to yield the best results. If the spread parameter is near zero, the network acts as a nearest neighbor classifier. As the spread parameter becomes large, the network takes into account several nearby patterns.

3. Support vector machines (SVMs) (Vapnik, 1998). SVMs with five different kernels, have been used. Training was performed by the least-squares method. The following kernel functions have been tested:
   - Gaussian RBF (denoted SVM1): $K(x_i,x_j)= \exp\{-\gamma ||x_i-x_j||^2\}$ with γ =1;
   - multilayer perceptron (denoted SVM2): $K(x_i, x_j)=S(x_i^T x_j -1)$, where $S(.)$ is a sigmoid function;
   - Quadratic kernel (denoted SVM3): $K(x_i,x_j)= (x_i^T x_j + 1)^2$;
   - Linear kernel (denoted SVM4): $K(x_i,x_j)= x_i^T x_j$;
   - Polynomial kernel (denoted SVM5): $K(x_i,x_j)= (x_i^T x_j + 1)^3$. A polynomial kernel of degree 4 is found to yield the same results with the cubic kernel.
4. For *K*-NNs, it is hard to find systematic methods for selecting the optimum number of the closest neighbors and the most suitable distance. Four *K*-NNs have been employed with different distance functions, such as the Euclidean denoted as KNN1, cityblock (i.e., sum of absolute differences) denoted as KNN2, cosine-based (i.e. one minus the cosine of the included angle between patterns) denoted as KNN3 and correlation-based (i.e., one minus the sample correlation between patterns) denoted as KNN4, respectively. We have selected *K*=2 in all experiments. Other values of K did not affect the classification accuracy unless the consensus rule was applied instead of the normal rule. In this case, none of the results of the *K*-NN would be stable and thus valid for classification.
5. Gaussian Mixture model (GMM) have been employed in many fields, e.g., speech and speaker recognition (Stephen & Paliwal, 2006; Reynolds & Rose, 1995). In GMM, during the training phase, pdf (probability density function) parameters for each class (gender) are estimated. Then, during the classification phase, a decision is taken for each test utterance by computing the maximum likelihood criterion. GMM is a combination of *K* Gaussian laws. Each law in the mixture is weighted and specified by two parameters: the mean and the covariance matrix (Σ_k).

5. Comparative results

Figure 1 illustrates the correct classification rates achieved by each of the aforementioned 11 classifiers on DES database, when 20% of the total utterances have been used for testing.
For each classifier, columns ``Total'', ''Male'', and "Female" correspond to the total correct classification rate, the rate of correct matches between the actual gender and the predicted one by the classifier for utterances uttered by male speakers, and the rate of correct matches between the actual gender and the predicted one by the classifier for utterances uttered by female speakers, respectively. The leftmost column shows the total correct classification rate. The middle and the rightmost columns are the classification rates that correspond to correct matches between the actual speaker gender (i.e. the ground truth) and the gender prediction by the classifier for male and female speakers, separately. In the sequel, the total correct classification rate, the correct classification rate for male speakers, and the correct classification rate for female speakers are abbreviated as TCCR, MCCR, and FCCR, respectively. In Figure 1, the maximum and minimum TCCR for DES were obtained by the SVM1 (90.94%) and the SVM2 (57.33%), respectively. The maximum and minimum MCCR for DES were related to GMM (95.42%) and SVM2 (58.11%), respectively. For FCCR on DES, the maximum and minimum values were obtained by the Bayes classifier with SFFS (91.07%) and SVM2 (56.54%), respectively. The best results for TCCR, MCCR and FCCR are marked with ``↓'' sign.

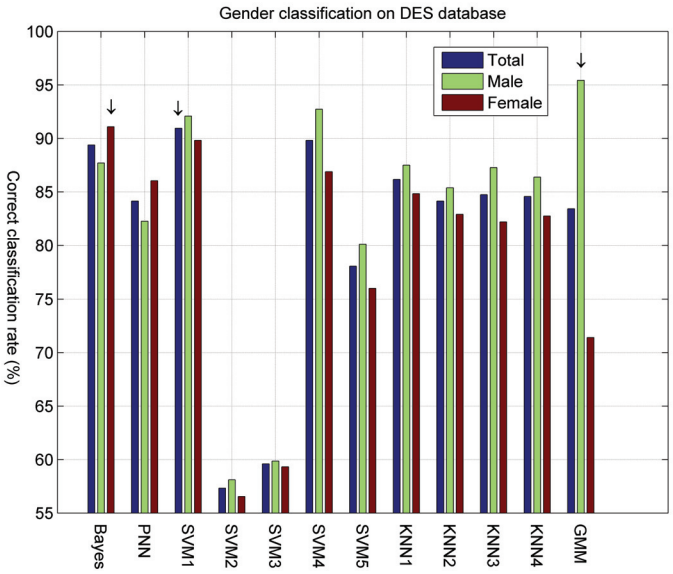


Fig. 1. Correct classification rates on DES database for the different methods when the size of test utterances is 20% of the total utterances.

Figures 2 & 3 demonstrate similar results for SES and GES databases, respectively.

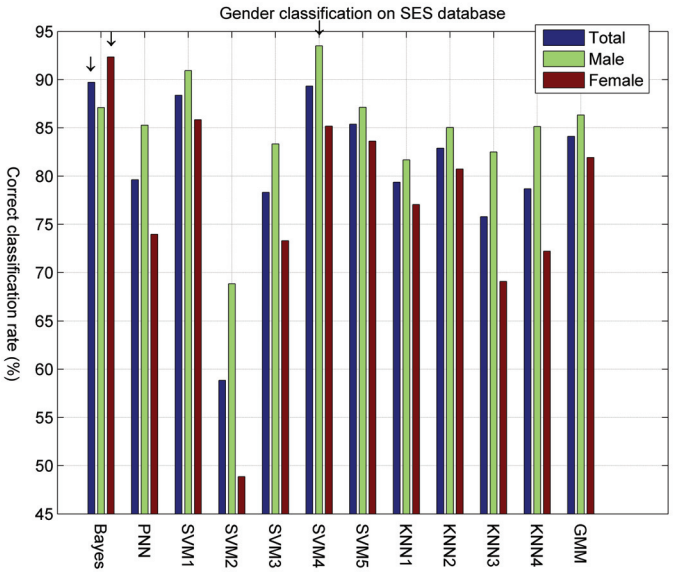


Fig. 2. Correct classification rates on SES database for the different methods when the size of test utterances is 20% of the total utterances.

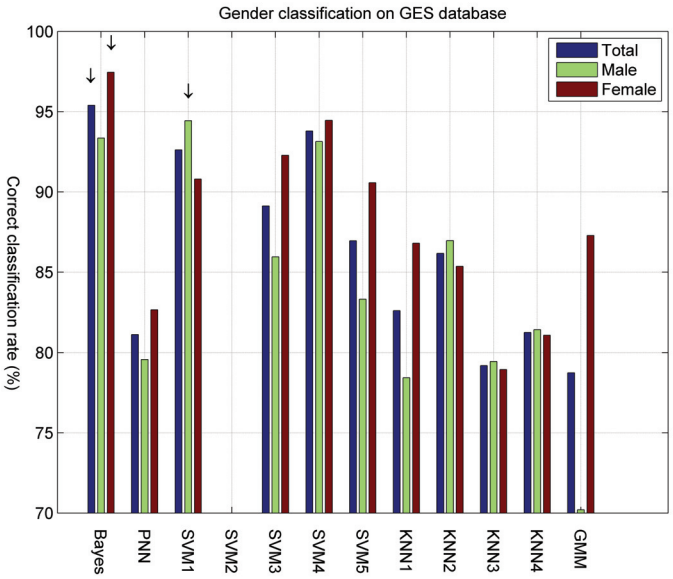


Fig. 3. Correct classification rates on GES database for the different methods when the size of test utterances is 20% of the total utterances.

In Figure 2, the maximum and minimum TCCR for SES were obtained by the the Bayes classifier using SFFS (89.73%) and the SVM2 (58.83%), respectively. The maximum and minimum MCCR for SES were related to SVM4 (93.51%) and SVM2 (68.83%), respectively. For FCCR on SES, the maximum and minimum values were obtained by the Bayes classifier with SFFS (92.36%) and SVM2 (48.86%), respectively.

In Figure 3, the maximum and minimum TCCR for GES were obtained by Bayes+SFFS (95.40%) and the GMM (78.74%), respectively. This is where SVM2 failed to classify at all. The maximum and minimum MCCR for GES were related to SVM1 (94.43%) and GMM (70.20%), respectively. The maximum and minimum values for FCCR on GES, were achieved by the Bayes classifier with SFFS (97.45%) and KNN3 (78.94%), respectively.

In the following, we concentrate on the top methods, i.e., SVM1, SVM4, GMM, and the Bayes classifier with SFFS. Table 2 demonstrates the confusion matrix for gender classification of the top methods after running each method several times and taking the mean value. The correct classification rates for each gender are shown in boldface. SVM1 outperforms the other methods achieving a correct classification rate of 90.94% (TCCR) with a standard deviation of 0.65. GMM is the best classifier, when the correct matches are between the actual gender and the predicted one by the classifier are measured for actors' utterances, yielding a rate of 95.42% (MCCR). The Bayes classifier using SFFS achieves a rate of 91.07%, when the correct matches between the actual gender and the predicted one by the classifier are measured for actresses' utterances (FCCR).

Similarly, Tables 3 & 4 show the confusion matrices for gender classification of the top methods on SES and GES databases, respectively. The Bayes classifier using SFFS outperforms the other methods achieving a correct classification rate of 89.74% (TCCR) with a standard deviation of 0. 1.03 on SES. It is also the best classifier for FCCR with 92.36% on

SES. SVM4 is considered as the best classifier for MCCR with 93.51% on SES. Also Bayes classifier using SFFS outperforms other classifiers for TCCR with 95.40% on GES with a standard deviation of 1.16. Moreover, it is the best classifier for FCCR with 97.45% on GES. SVM1 is the best classifier for MCCR with 94.43% on GES.

GMM	Response (%)			**Bayes-SFFS**	Response (%)	
Ground Truth ↓	Male	Female		Ground Truth ↓	Male	Female
Male	**95.42**	4.58		Male	**87.69**	12.31
Female	28.59	**71.41**		Female	8.93	**91.07**
Correct rate	83.42%			Correct rate	89.38%	
SVM1	Response (%)			**SVM4**	Response (%)	
Ground Truth ↓	Male	Female		Ground Truth ↓	Male	Female
Male	**92.08**	7.92		Male	**92.72**	7.28
Female	10.19	**89.81**		Female	13.12	**86.88**
Correct rate	90.95%			Correct rate	89.80 %	

Table 2. Confusion matrix for the 4 best methods when 20% of the utterances of DES database are used for testing.

GMM	Response (%)			**Bayes-SFFS**	Response (%)	
Ground Truth ↓	Male	Female		Ground Truth ↓	Male	Female
Male	**86.34**	13.66		Male	**87.11**	12.89
Female	18.08	**81.92**		Female	7.64	**92.36**
Correct rate	84.13%			Correct rate	89.74%	
SVM1	Response (%)			**SVM4**	Response (%)	
Ground Truth ↓	Male	Female		Ground Truth ↓	Male	Female
Male	**90.94**	9.06		Male	**93.51**	6.49
Female	14.16	**85.84**		Female	14.83	**85.17**
Correct rate	88.39%			Correct rate	89.34%	

Table 3. Confusion matrix for the 4 best methods when 20% of the utterances of SES database are used for testing.

GMM	Response (%)			**Bayes-SFFS**	Response (%)	
Ground Truth ↓	Male	Female		Ground Truth ↓	Male	Female
Male	**70.20**	29.80		Male	**93.34**	6.66
Female	12.72	**87.28**		Female	2.55	**97.45**
Correct rate	78.74%			Correct rate	95.40%	
SVM1	Response (%)			**SVM4**	Response (%)	
Ground Truth ↓	Male	Female		Ground Truth ↓	Male	Female
Male	**94.43**	5.57		Male	**93.13**	6.87
Female	9.21	**90.79**		Female	5.55	**94.45**
Correct rate	92.61%			Correct rate	93.79%	

Table 4. Confusion matrix for the 4 best methods when 20% of the utterances of GES database are used for testing.

In the following, the behaviour of the best classifiers are investigated against changing the parameters. Figures 4, 5 & 6 highlight the behaviour of the Bayes classifier with SFFS on DES, SES and GES databases, respectively, for varying numbers of cross-validation repetitions and varying portions of utterances engaged in testing. The flatness of the shapes confirms that if we select 20% of the utterances for testing and 20 repetitions, our judgements are fair.

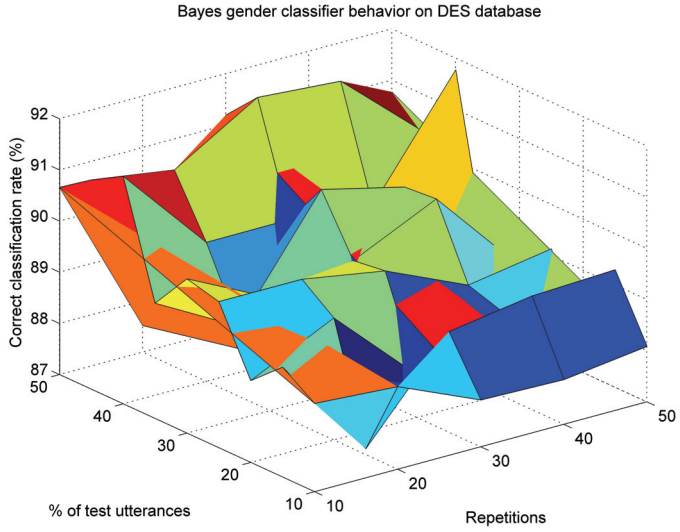


Fig. 4. Probability of correct classification of the Bayes classifier with SFFS on DES database for varying repetitions and portions of the utterances used during testing.

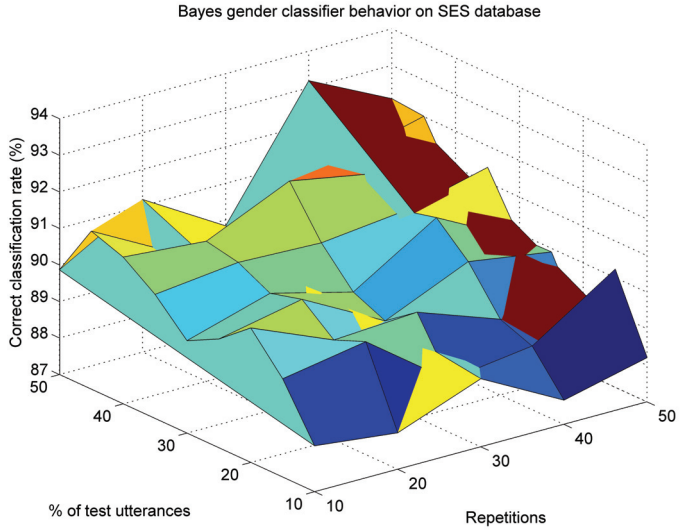


Fig. 5. Probability of correct classification of the Bayes classifier with SFFS on SES database for varying repetitions and portions of the utterances used during testing.

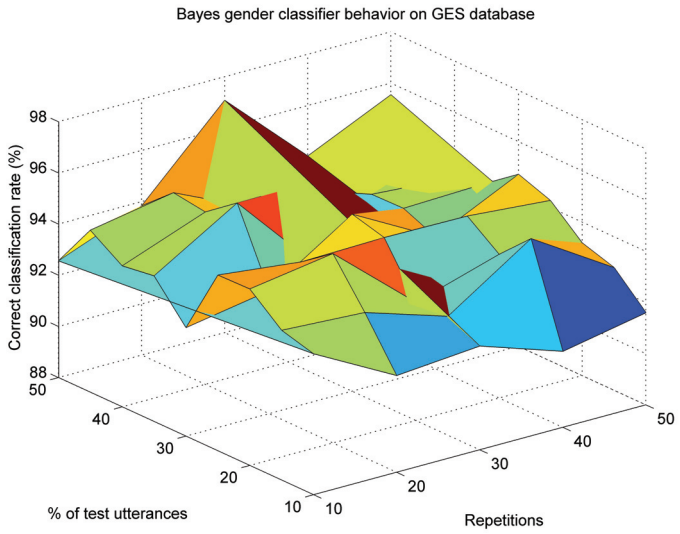


Fig. 6. Probability of correct classification of the Bayes classifier with SFFS on GES database for varying repetitions and portions of the utterances used during testing.

Tables 5, 6 and 7 investigate, in detail, the minimum and maximum rates measured for the Bayes classifier with SFFS on DES, SES and GES databases, respectively. The minimum TCCR for DES, SES and GES was measured when 20, 40 and 40 repetitions were made using 15%, 10% and 45% of utterances for testing, respectively. The maximum TCCR for DES, SES and GES was measured by making 30, 40 and 30 repetitions and employing 45%, 50% and 50% of the available utterances for testing, respectively. The minimum MCCR for DES, SES and GES was measured when 50, 40 and 40 repetitions were made while using 30%, 10% and 45% of utterances for testing, respectively. The maximum MCCR for DES, SES and GES was measured by making 40, 50 and 30 repetitions and employing 45%, 50% and 50% of the available utterances for testing, respectively. For FCCR on DES, SES and GES, 20, 10 and 40 repetitions and 50%, 50% and 45% of utterances for testing yield the minimum rate, respectively, while 30, 40 and 20 repetitions and 45%, 50% and 50% of the utterances engaged in testing are required for the maximum rate, respectively.

Rates	Min (%)	Max (%)	Mean (%)	Std (%)
TCCR	87.11	91.82	89.46	1.22
MCCR	83.94	92.58	87.71	1.71
FCCR	87.07	93.75	91.21	1.49

Table 5. Behaviour of Bayes classifier with SFFS for gender classification on DES database.

Rates	Min (%)	Max (%)	Mean (%)	Std (%)
TCCR	87.66	93.19	90.10	1.22
MCCR	83.96	90.28	87.42	1.55
FCCR	89.83	96.49	92.79	1.52

Table 6. Behaviour of Bayes classifier with SFFS for gender classification on SES database.

Rates	Min (%)	Max (%)	Mean (%)	Std (%)
TCCR	89.67	97.08	93.69	1.34
MCCR	85.66	97.74	90.59	2.25
FCCR	93.67	99.47	96.79	1.21

Table 7. Behaviour of Bayes classifier with SFFS for gender classification on GES database.

Tables 8-10 illustrate the behaviour of SVM1 on DES, SES and GES databases, respectively, when the size of the test utterances ranges between 10% and 50% of the available utterances. For TCCR on DES, SES and GES, 50%, 45% and 45% of the available utterances yield the minimum value, while 40%, 10% and 10% of the utterances yield the maximum value, respectively. For MCCR, 25%, 50% and 35% the test utterances yield the minimum value while 40%, 10% and 25% of the available utterances yield the maximum value for MCCR. For FCCR, 15%, 45% and 45% of the utterances engaged during testing yield the minimum value, while 20%, 15% and 15% of the utterances yield the maximum value.
Tables 11 and 12 show the behaviour of SVM4 on DES, SES and GESdatabases. The size of the test utterances ranges between 10% and 50% of the available utterances. For TCCR on DES, SES and GES, 30%, 35% and 50% of the available utterances yield the minimum value,

Rates	Min (%)	Max (%)	Mean (%)	Std (%)
TCCR	88.91	91.17	89.79	0.82
MCCR	89.45	94.14	91.38	1.50
FCCR	86.18	89.81	88.19	1.32

Table 8. Behaviour of SVM1 on DES database for gender classification when the size of the test utterances varies between 10% and 50% of the utterances.

Rates	Min (%)	Max (%)	Mean (%)	Std (%)
TCCR	86.14	89.71	87.59	1.25
MCCR	88.32	93.14	90.21	1.45
FCCR	82.97	86.37	84.96	1.31

Table 9. Behaviour of SVM1 on SES database for gender classification when the size of the test utterances varies between 10% and 50% of the utterances.

Rates	Min (%)	Max (%)	Mean (%)	Std (%)
TCCR	89.49	92.93	91.32	1.23
MCCR	90.73	95.41	93.01	2.01
FCCR	87.79	91.48	89.64	1.23

Table 10. Behaviour of SVM1 on GES database for gender classification when the size of the test utterances varies between 10% and 50% of the utterances.

Rates	Min (%)	Max (%)	Mean (%)	Std (%)
TCCR	87.19	90.46	88.68	1.19
MCCR	88.19	92.72	90.40	1.39
FCCR	83.81	89.17	86.96	1.78

Table 11. Behaviour of SVM4 on DES database for gender classification when the size of the test utterances varies between 10% and 50% of the utterances.

Rates	Min (%)	Max (%)	Mean (%)	Std (%)
TCCR	86.87	89.56	88.09	0.93
MCCR	91.05	94.09	92.75	0.98
FCCR	80.17	86.23	83.44	2.03

Table 12. Behaviour of SVM4 on SES database for gender classification when the size of the test utterances varies between 10% and 50% of the utterances.

Rates	Min (%)	Max (%)	Mean (%)	Std (%)
TCCR	90.30	93.91	92.38	1.32
MCCR	87.86	93.88	91.07	2.10
FCCR	92.72	94.69	93.68	0.76

Table 13. Behaviour of SVM4 on GES database for gender classification when the size of the test utterances varies between 10% and 50% of the utterances.

while 15%, 25% and 10% of the utterances yield the maximum value, respectively. For MCCR, 30%, 15% and 50% the test utterances yield the minimum value while 20%, 10% and 15% of the available utterances yield the maximum value for MCCR. For FCCR, 10%, 35% and 25% of the utterances engaged during testing yield the minimum value, while 40%, 15% and 10% of the utterances yield the maximum value.

Tables 14-16 illustrate the behaviour of GMM on DES and SES databases, respectively, when the size of the test utterances ranges between 10% and 50% of the available utterances (GMM is not a good classifier for GES). For TCCR on DES and SES, 40% and 50% of the available utterances yield the minimum value, while 50% and 10% of the utterances yield the maximum value, respectively. For MCCR, 50% and 20% of the test utterances yield the minimum value while 10% and 50% of the available utterances yield the maximum value for MCCR. For FCCR, 40% and 50% of the utterances engaged during testing yield the minimum value, while 50% and 10% of the utterances yield the maximum value.

Rates	Min (%)	Max (%)	Mean (%)	Std (%)
TCCR	82.02	84.81	83.59	0.89
MCCR	91.73	96.06	94.71	1.33
FCCR	68.88	77.89	72.47	2.70

Table 14. Behaviour of GMM on DES database for gender classification when the size of the test utterances varies between 10% and 50% of the utterances.

Rates	Min (%)	Max (%)	Mean (%)	Std (%)
TCCR	69.66	85.03	79.97	5.53
MCCR	86.34	92.49	88.92	2.37
FCCR	46.83	83.67	71.02	13.36

Table 15. Behaviour of GMM on SES database for gender classification when the size of the test utterances varies between 10% and 50% of the utterances.

The computational speed was measured using a PC P4, 3GHz CPU and 1 GB RAM while a virus shield was active.

Classifier	DES	SES	GES
GMM	21.81	31.80	44.36
Bayes+SFFS	30.22	48.11	30.76
PNN	7.88	1.33	1.98
SVM1	2.90	1.78	0.34
SVM2	1.62	1.73	0.33
SVM3	1.34	1.46	0.28
SVM4	1.30	1.43	0.28
SVM5	1.38	1.47	0.28
KNN1	0.87	1.00	0.30
KNN2	0.69	0.99	0.30
KNN3	0.40	0.41	0.16
KNN4	0.44	0.42	0.18

Table 17. Computational time (in sec) for different classifiers on different databases.

Accordingly, SVM1 outperforms the other methods with respect to all the four factors: TCCR, MCCR, FCCR, and speed for emotional speech.
However, for non-emotional speech, we recommend GMM.

6. Conclusions

We have investigated several popular methods for gender classification by processing emotionally colored speech from the DES, SES and GES databases. Based on the results, several conclusions can be drawn. The SVM with a Gaussian RBF kernel (SVM1) has demonstrated to yield the most accurate results considering other parameters such as the computation speed. The correct gender classification rates have been more than 90% when emotional speech utterances from both genders were processed, or when emotional speech utterances of male or female speakers were used. Another acceptable alternative is the Bayes classifier using sequential floating forward feature selection.

7. References

F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier and B. Weiss (2005). A database of German Emotional Speech. In Proc. Interspeech 2005 Conf. Lisbon, Portugal.

D. G. Childers, K. Wu, and D. M. Hicks (1987). Factors in voice quality: acoustic features related to gender. In Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, volume 1, pages 293–296.

B. Efron and R. E. Tibshirani (1993), An Introduction to the Bootstrap, Chapman & Hall/CRC, N.Y..

I. S. Engberg and A. V. Hansen (1996). Documentation of the Danish Emotional Speech database (DES). Technical Report Internal AAU report, Center for Person, Kommunikation, Aalborg Univ., Denmark.

E. Fix and J. Hodges (1991-a). Discriminatory analysis, nonparametric discrimination, consistency properties. In B. Dasarathy, editor, Nearest Neighbor Pattern Classification Techniques, pages 32–39. IEEE Computer Society Press, Los Alamitos, CA.

E. Fix and J. Hodges (1991-b). Discriminatory analysis: samll sample performance. In B. Dasarathy, editor, Nearest Neighbor Pattern Classification Techniques, pages 40–56. IEEE Computer Society Press, Los Alamitos, CA.

K. Fukunaga and P. M. Narendra (1975). A branch and bound algorithm for computing k-nearest neighbors. IEEE Trans. Computers, 24:750–753.

H. Harb and L. Chen (2005). Voice-based gender identification in multimedia applications. J. Intelligent Information Systems, 24(2):179–198.

Y. L. Lin and G. Wei (2005). Speech emotion recognition based on HMM and SVM. In Proc. IEEE Int. Conf. Machine Learning and Cybernetics, volume 8, pages 4898–4901. Guangzhou, China.

F. Metze, J. Ajmera, R. Englert, U. Bub, F. Burkhardt, J. Stegmann, C. Muller, R. Huber, B. Andrassy, J. G. Bauer, and B. Little (2007). Comparison of four approaches to age and gender recognition for telephone applications. In Proc. 2007 IEEE Int. Conf. Acoustics, Speech and Signal Processing, volume 4, pages 1089–1092. Honolulu.

G. Peterson and H. Barney (1952). Control methods used in a study of vowels. Journal of Acoustical Society of America, 24, 175-184.

P. Pudil, J. Novovicova, and J. Kittler (1994). Floating search methods in feature selection. Pattern Recognition Letters, 15(11):1119–1125.

D. Reynolds and R. Rose (1995). Robust text-independent speaker identification using gaussian mixture speaker models. IEEE Transactions on Speech and Audio Processing, vol. 3(1): 72–83.

M. H. Sedaaghi (2008). Documentation of the Sahand Emotional Speech database (SES). Technical Report, Department of Electrical Eng., Sahand Univ. of Tech, Iran.

D. F. Specht (1990). Probabilistic neural networks. Neural Networks, 3:109–118.

S. Stephen and K. K. Paliwal (2006). Scalable distributed speech recognition using Gaussian mixture model-based block quantisation, Speech Communication, vol. 48: 746-758.

S.E. Tranter and D. A. Reynolds (2006). An Overview of Automatic Speaker Diarisation Systems. IEEE Trans. Speech & Audio Processing: Special issue on Rich Transcription, 14(5): 1557-1565.

V. N. Vapnik (1998). The Nature of Statistical Learning Theory. Springer, N.Y..

R. Vergin, A. Farhat, and D. O'Shaughnessy (1996). Robust gender-dependent acoustic phonetic modelling in continuous speech recognition based on a new automatic male/female classification. In Proc. Int. IEEE Conf. Acoustics, Speech, and Signal Processing (ICASSP-96), volume 2, pages 1081–1084. Atlanta.

D. Ververidis and C. Kotropoulos (2004). Automatic speech classification to five emotional states based on gender information. In Proc. European Signal Processing Conf. (EUSIPCO 04), volume 1, pages 341–344. Vienna, Austria.

D. Ververidis and C. Kotropoulos (2006), Fast sequential floating forward selection applied to emotional speech features estimated on DES and SUSAS data collections, in Proc. 14th. European Signal Processing Conf. Florence, Italy.

K. Wu and D. G. Childers (1991). Gender recognition from speech. Part I: Coarse analysis. J. Acoust. Soc. of Am., 90(4):1828–1840.

Z. Xiao, E. Dellandr´ea, W. Dou, and L. Chen (2007). Hierarchical classification of emotional speech. Technical Report RR-LIRIS-2007-06, LIRIS UMR 5205 CNRS.

L. Xiaofan and S. Simske (2004). Phoneme-less hierarchical accent classification. In Proc. 38th. Asilomar Conf. Signals, Systems and Computers, volume 2, pages 1801–1804. California.

Y. Zeng, Z. Wu, T. Falk, and W. Y. Chan (2006). Robust GMM based gender classification using pitch and RASTA-PLP parameters of speech. In Proc. 5th. IEEE Int. Conf. Machine Learning and Cybernetics, pages 3376–3379. China.

Emotion recognition

21

Recognition of Paralinguistic Information using Prosodic Features Related to Intonation and Voice Quality

Carlos T. Ishi
ATR
Japan

1. Introduction

Besides the linguistic (verbal) information conveyed by speech, the paralinguistic (non-verbal) information, such as intenning the classification of paralinguistic information. Among the several paralinguistic items extions, attitudes and emotions expressed by the speaker, also convey important meanings in communication. Therefore, to realize a smooth communication between humans and spoken dialogue systems (such as robots), it becomes important to consider both linguistic and paralinguistic information.

There is a lot of past research concerpressing intentions, attitudes and emotions, most previous research has focused on the classification of the basic emotions, such as anger, happiness and sadness (e.g., Fernandez et al., 2005; Schuller et al., 2005; Nwe et al., 2003; Neiberg et al., 2006). Other works deal with the identification of attitudes and intentions of the speaker. For example, Fujie et al. (2003) report about the identification of positive/negative attitudes of the speaker, while Maekawa (2000) reports about the classification of paralinguistic items like admiration, suspicion, disappointment and indifference. In Hayashi (1999), paralinguistic items like affirmation, asking again, doubt and hesitation were also considered. In the present work, aiming at smooth communication in dialogue between humans and spoken dialogue systems, we consider a variety of paralinguistic information, including intentions, attitudes and emotions, rather than limiting our focus to the basic emotions.

The understanding of paralinguistic information becomes as important as linguistic information in spoken dialogue systems, especially in interjections such as “eh”, “ah”, and “un”. Such interjections are frequently used to express a reaction to the conversation partner in a dialogue scenario in Japanese, conveying some information about the speaker’s intention, attitude, or emotion. As there is little phonetic information represented by such interjections, most of the paralinguistic information is thought to be conveyed by its speaking style, which can be described by variations in prosodic features, including voice quality features.

So far, most previous research dealing with paralinguistic information extraction has focused only on intonation-related prosodic features, using fundamental frequency (F0), power and duration (e.g., Fujie et al., 2003; Hayashi, 1999). Others also consider segmental features like cepstral coefficients (e.g., Schuller et al., 2005; Nwe et al., 2003). However,

analyses of natural conversational speech have shown the importance of several voice quality features caused by non-modal phonations (e.g., Klasmeyer et al., 2000; Kasuya et al., 2000; Gobl et al., 2003; Campbell et al., 2003; Fujimoto et al., 2003; Erickson, 2005).

The term "voice quality" can be used in a broad sense, as the characteristic auditory colouring of an individual speaker's voice, including qualities such as nasalized, dentalized, and velarized, as well as those brought about by changing the vocal tract length or hypopharyngeal area (e.g., Imagawa et al., 2003; Kitamura et al., 2005; Dang et al., 1996). Here, we use it in a narrow sense of the quality deriving solely from laryngeal activity, i.e., from different vibration modes of the vocal folds (different phonation types), such as breathy, whispery, creaky and harsh voices (Laver, 1980).

Such non-modal voice qualities are often observed especially in expressive speech utterances, and should be considered besides the classical intonation-related prosodic features. For example, whispery and breathy voices are characterized by the perception of a turbulent noise (aspiration noise) due to air escape at the glottis, and are reported to correlate with the perception of fear (Klasmeyer et al., 2000), sadness, relaxation and intimateness in English (Gobl et al., 2003), and with disappointment (Kasuya et al., 2000; Fujimoto et al., 2003) or politeness in Japanese (Ito, 2004). Vocal fry or creaky voices are characterized by the perception of very low fundamental frequencies, where individual glottal pulses can be heard, or by a rough quality caused by an alternation in amplitude, duration or shape of successive glottal pulses. Vocal fry may appear in low tension voices correlating with sad, bored or relaxed voices (Klasmeyer et al., 2000; Gobl et al., 2003), or in pressed voices expressing attitudes/feelings of admiration or suffering (Sadanobu, 2004). Harsh and ventricular voices are characterized by the perception of an unpleasant, rasping sound, caused by irregularities in the vocal fold vibrations in higher fundamental frequencies, and are reported to correlate with anger, happiness and stress (Klasmeyer et al., 2000; Gobl et al., 2003).

Further, in segments uttered by such voice qualities (caused by non-modal phonation types), F0 information is often missed by F0 extraction algorithms due to the irregular characteristics of the vocal fold vibrations (Hess, 1983). Therefore, in such segments, the use of only prosodic features related to F0, power and duration, would not be enough for their complete characterization. Thus, other acoustic features related to voice quality become important for a more suitable characterization of their speaking style.

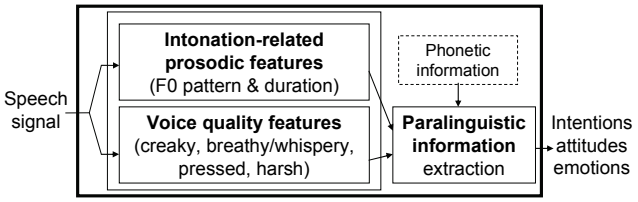


Fig. 1. Block diagram of the proposed framework for paralinguistic information extraction.

Fig. 1 shows our framework proposed for extraction of paralinguistic information, by using information of voice quality features, in addition to intonation-related prosodic features. In our previous research, we have proposed several acoustic parameters for representing the features of intonation and specific voice qualities (Ishi, 2004; Ishi, 2005; Ishi et al., 2005). In the present chapter, evaluation on the performance of the acoustic parameters in the automatic recognition of paralinguistic information is presented.

The rest of the chapter is organized as follows. In Section 2, the speech data and the perceptual labels used in the analysis are described. Section 3 describes the acoustic parameters representing prosodic and voice quality features. In Section 4, the automatic detection of paralinguistic information is evaluated by using the acoustic parameters described in Section 3, and Section 5 concludes the chapter.

2. Description of the speech data and the perceptual data

2.1 Description of the speech data for analysis and experimental setup

In the present research, the Japanese interjections "e" and "un" (including variations such as "e", "eh", "ee", "eeee" and "un", "uun", "hun", "nnn", "uhn", etc.) are chosen for analysis and evaluation. These interjections are often used to express a reaction in Japanese conversational speech, and convey a large variety of paralinguistic information depending on its speaking style (Campbell et al., 2004). Possible paralinguistic information (intentions, attitudes or emotions) transmitted by varying the speaking styles of the interjections "e" and "un" are listed in Table 1.

Original adjectives and translations	PI item
"koutei", "shoudaku" (affirm, accept)	***Affirm***
"dooi", "rikai", "nattoku" (agree, understand, consent)	***Agree***
"aiduchi" (backchannel: agreeable responses)	***Backchannel***
"hitei" (deny, negate)	***Deny***
"kangaechuu", "filler" (thinking, filler)	***Thinking***
"mayoi", "konwaku", "tomadoi", "nayamu", "chuucho" (embarrassed, undecided, hesitated)	***Embarrassed***
"kikikaeshi" (ask for repetition)	***AskRepetition***
"bikkuri", "odoroki" (surprised, amazed, astonished)	***Surprised***
"igai" (unexpected)	***Unexpected***
"utagai", "gimon" (suspicious, doubt)	***Suspicious***
"hinan", "kyozetsu" (blame, criticise, reject)	***Blame***
"ken'o", "iya" (disgusted, disliked)	***Disgusted***
"fuman" (dissatisfied, frustrated)	***Dissatisfied***
"kanshin" (admired)	***Admired***
"senbou", "urayamashii" (envious)	***Envious***

Table 1. List of paralinguistic information conveyed by the interjections "e" and "un".

The list of Table 1 was obtained by referring to a list of speech acts annotated for the interjections "e" and "un" in the JST/CREST ESP conversational speech database (JST/CREST ESP Project homepage). The items of the list have been obtained by free-text annotations of four subjects, in "e" and "un" utterances appearing in natural conversations. The annotated words have been arranged by the four subjects for reducing redundancies. We do not guarantee that this list contains all the possible paralinguistic information that the interjections "e" and "un" can convey. However, we consider that this list is rich enough for our purposes of human-machine communication.

The list of Table 1 includes paralinguistic items expressing some intention, such as affirm and ask for repetition, some attitude, such as suspicious and dissatisfied, and some emotion, such as surprised and disgusted. These items are more related to intentions or speech acts

conveyed by the utterances, rather than the basic emotions, such as anger, happy and sadness. Since it is difficult to clearly classify these items as intentions, attitudes or emotions, in the present research we simply call them paralinguistic information (PI) items.
In the present research, speech data was newly recorded in order to get a balanced data in terms of the PI conveyed by the interjections "e"/"un". For that purpose, sentences were elaborated in such a way to induce the subject to produce a specific PI. Some short sentences were also elaborated to be spoken after the interjections "e"/"un", in order to get a reaction as natural as possible. Two sentences were elaborated for each PI item of Table 1, by two native speakers of Japanese. (Part of the sentences is shown in the Appendix.)
The sentences were first read by one native speaker. These sentences will be referred to as "inducing utterances". Then, subjects were asked to produce a target reaction, i.e., utter in a way to express a specific PI, through the interjection "e", after listening to each pre-recorded inducing utterance. The same procedure was conducted for the interjection "un". A short pause was required between "e"/"un" and the following short sentences. Further, the utterance "he" (with the aspirated consonant /h/ before the vowel /e/) was allowed to be uttered, if the subject judged that it was more appropriate for expressing some PI.
Utterances spoken by six subjects (two male and four female speakers between 15 to 35 years old) are used for analysis and evaluation. In addition to the PI list, speakers were also asked to utter "e" and "un" with a pressed voice quality, which frequently occurs in natural expressive speech (Sadanobu, 2004), but was found more difficult to naturally occur in an acted scenario. Of the six speakers, four could produce pressed voice utterances. The data resulted in 173 "e" utterances, and 172 "un" utterances.
For complementing the data in terms of voice quality variations, another dataset including utterances of a natural conversational database (JST/CREST ESP database) was also prepared for evaluation. The dataset is composed of 60 "e" utterances containing non-modal voice qualities, extracted from natural conversations of one female speaker (speaker FAN), resulting in a total of 405 utterances for analysis.
All the "e" and "un" utterances were manually segmented for subsequent analysis and evaluation.

2.2 Perceptual labels of paralinguistic information (PI) items

Perceptual experiments were conducted to verify how good the intended (induced) PI items could be correctly recognized when listening only to the monosyllabic utterances, i.e., in a context-free situation. The purpose is to verify the ambiguity in the expression of a PI item, since the same speaking style could be used to express different PI items, in different contexts.
Three untrained subjects (who are different from the speakers) were asked to select from the PI item list of Table 1, one or multiple items that could be expressed by each stimuli (i.e., the segmented "e"/"un" utterances). Multiple items were allowed since the same speaking style could be used to express different PI items. As a result of the multiple selections, two, three and more than three labels were selected in 40%, 14% and 4% of the utterances, respectively. Regarding the subjects' agreement, in 51% of the utterances, all three subjects agreed in assigning the same PI items, while in 93% of the utterances, at least two of the three subjects agreed in assigning the same PI items.
Fig. 2 shows the matches, mismatches and ambiguities between intended and perceived PI items, when listening only to the "e" utterances, i.e., in a context-free situation. The perceptual degrees (between 0 to 1) are computed by counting the number of labels of a

perceived PI item in all utterances of an intended PI item, and dividing by the number of utterances, and by the number of subjects.

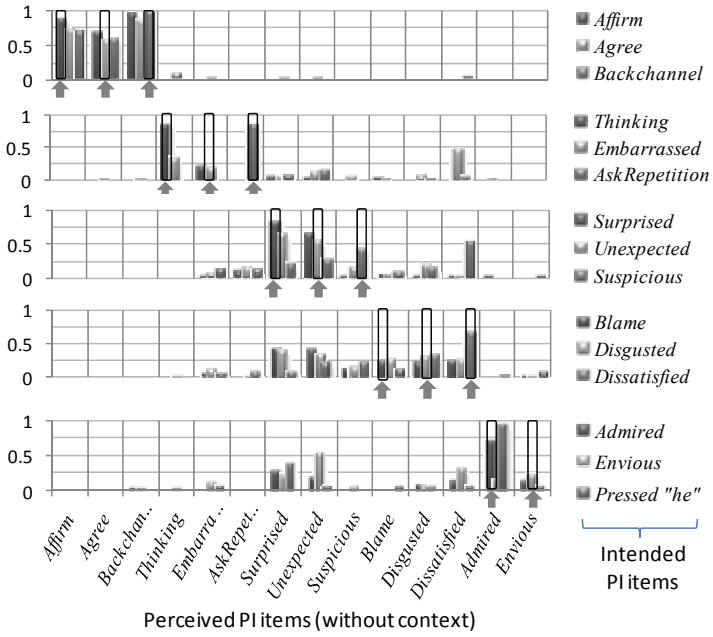


Fig. 2. Perceptual degrees of the intended PI items of "e" utterances (without context, i.e., by listening only to the interjections). The bars indicated by arrows show the matching degrees between intended and perceived items.

First, regarding the matching between intended and perceived PI items, it can be observed in the bars indicated by arrows in Fig. 2 that Affirm, Backchannel, Thinking, AskRepetition and Surprised show high matching degrees, while Agree, Unexpected, Suspicious, Dissatisfied and Admired show moderate matching. However, Embarrassed, Blame, Disgusted, and Envious show very low matching degrees, indicating that the intended PI could not be perceived in most of their utterances, in a context-free situation.

The mismatches and ambiguities between intended and perceived PI items are shown by the bars excluding the ones indicated by arrows in Fig. 2. Among the PI items with large mismatches, most in Embarrassed is perceived as Thinking or Dissatisfied, while most in Unexpected is perceived as Surprised. Some of the confusions are acceptable, since there may be situations where the speaker is thinking while embarrassed, or where the speaker feels surprised and unexpected at the same time. Confusion is also found between samples of Blame, Disgusted, Dissatisfied and Suspicious. This is also an acceptable result, since all these PI items express negative reactions.

However, Surprised, Unexpected and Dissatisfied are perceived in the stimuli of several intended PI items. This indicates that the identification of these PI items would only be possible by considering context information, for example, by taking into account the sentence following the "e" utterances.

Further, even among the PI items where a good matching was achieved between intended and perceived items, ambiguity may exist between some of the PI items. For example, there is high confusion between the stimuli of Affirm, Agree and Backchannel, but no confusion between these and other PI items.
Finally, regarding the pressed voice samples, pressed "he" was mostly perceived as Admired, while pressed "e" (omitted from Fig. 2) was perceived as Embarrassed, Disgusted or Dissatisfied.
The results above imply that an automatic detection of these ambiguous items will also probably be difficult based only on the speaking style of the utterance "e", i.e., without using context information.
From the results above, we can predict that many of the PI items can not be identified without context information. However, we can expect that some groups of PI items can be roughly discriminated, even when context is not considered: {Affirm/Agree/Backchannel}, {Thinking/Embarrassed}, {AskRepetition}, {Surprised/Unexpected}, {Blame/Disgusted/ Dissatisfied/Suspicious}, and {Admired/Envious}. These PI groups will be used as a basis to evaluate how much they can be discriminated by the use of intonation and voice quality-related prosodic features in "e"/"un" utterances (i.e., without context information).
Finally, 35 of the 405 utterances, corresponding to the mismatches between different PI groups, were considered as badly-acted, and were removed from the subsequent evaluation of automatic identification.

2.3 Perceptual voice quality labels and relationship with paralinguistic information

Perceptual voice quality labels are annotated for two purposes. One is to verify their effects in the representation of different PI items. Another is to use them as targets for evaluating the automatic detection of voice qualities.
The perceptual voice quality labels are annotated by one subject with knowledge about laryngeal voice qualities (the first author), by looking at the waveforms and spectrograms, and listening to the samples. Samples for several voice quality labels can be listened in the Voice quality sample homepage. The voice quality labels are annotated according to the following criteria.

- m: modal voice (normal phonation).
- w: whispery or breathy voices (aspiration noise is perceived throughout the utterance).
- a: aspiration noise is perceived in the offset of the last syllable of the utterance.
- h: harsh voice (rasping sound, aperiodic noise) is perceived.
- c: creaky voice or vocal fry is perceived.
- p: pressed voice is perceived.
- Combination of the above categories: for example, hw for harsh whispery, and pc for pressed creaky.

A question mark "?" was added for each voice quality label, if their perception were not clear. Fig. 3 shows the distributions of the perceived voice quality categories for each PI item.
We can first observe in Fig. 4 that soft aspiration noise (w?) is perceived in some utterances of almost all PI items. In contrast, strong aspiration noise (w), harsh or harsh whispery voices (h, hw) and syllable offset aspiration noise (a, a?) are perceived in PI items expressing some emotion or attitude (Surprised, Unexpected, Suspicious, Blame, Disgusted, Dissatisfied and Admired). This indicates that the detection of these voice qualities (w, h,

hw, a) could be useful for the detection of these expressive PI items. The soft aspiration noise (w?) appearing in emotionless items (Affirm, Agree, Backchannel and Thinking) is thought to be associated to politeness (Ito, 2004).

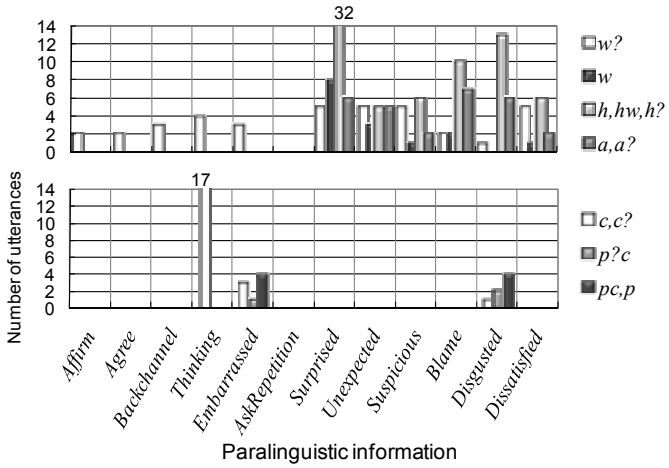


Fig. 3. Distribution of perceived categories of whispery/breathy/aspirated (w, a), harsh (h, hw), creaky (c), and pressed (p) voice qualities, for each paralinguistic information item.

Regarding to creaky voices (c), we can observe in the figure that they are perceived in Thinking, Embarrassed, Disgusted and Admired. However, the additional perception of pressed voices (p) is important to discriminate between emotionless fillers (Thinking), and utterances expressing some emotion or attitude (Admired, Disgusted and Embarrassed).
It is worth mentioning that the use of non-modal voice qualities is not strictly necessary for expressing an emotion or attitude, since different speakers may use different strategies to express a specific PI item. However, the results of the present section imply that when a non-modal voice quality occurs in an utterance, it will probably be associated with an emotion or an attitude.

3. Acoustic parameters representing prosodic features related to F0, duration and voice quality

In this section, we describe acoustic parameters that potentially represent the perception of intonation and voice quality-related prosodic features, which are responsible for the discrimination of different speaking styles, and verify their performance in automatic detection.

3.1 Acoustic parameters related to intonation-related prosodic features: F0move and duration

The main acoustic features used for intonation-related prosodic features are fundamental frequency (F0), power and segmental duration. In the present study, we avoid the use of power as a prosodic feature, due to its large variability caused by the microphone gains, the difference in microphone types, the distance between the mouth and the microphone, and the background noise.

In Ishi (2005), a set of parameters was proposed for describing the intonation of phrase finals (phrase final syllables), based on F0 and duration information. Here, we use a similar set of parameters with some modifications, for the monosyllabic "e" and "un" utterances.
For the pitch-related parameters, the F0 contour is estimated as a first step. In the present research, F0 is estimated by a conventional method based on autocorrelation. Specifically, the normalized autocorrelation function of the LPC inverse-filtered residue of the pre-emphasized speech signal is used. However, any algorithm that can reliably estimate F0 could be used instead. All F0 values are then converted to a musical (log) scale before any subsequent processing. Expression (1) shows a formula to produce F0 in semitone intervals.

$$F0[\text{semitone}] = 12 * \log2\,(F0[\text{Hz}]) \quad (1)$$

In the second step, each (monosyllabic) utterance is broken in two segments of equal length, and representative F0 values are extracted for each segment. In Ishi (2005), several candidates for the representative F0 values have been tested, and here, we use the ones that best matched with perceptual scores of the pitch movements. For the first segment, an average value is estimated using F0 values within the segment (F0avg2a). For the second segment, a target value is estimated as the F0 value at the end of the segment of a first order regression line of F0 values within the segment (F0tgt2b). In other words, for the first portion of the utterance, an average or predominant F0 value is perceived, while in the final portion, a target value to where F0 is moving is perceived.
A variable called F0move is then defined as the difference between F0tgt2b and F0avg2a, as shown in expression (2), quantifying the amount and direction of the F0 movement within a syllable.

$$F0move2 = F0tgt2b - F0avg2a \quad (2)$$

F0move is positive for rising F0 movements, and negative for falling movements. It has been shown that F0move parameters match better with the human pitch perception, rather than linear regression-based slope parameters. Details about the evaluation of the correspondence between F0move and perceptual features can be found in Ishi (2005).
The representation of F0 movements by F0move is valid when F0 only rises, only falls, or does not change within a syllable. This condition is true for most cases in Japanese syllables. However, there are cases where F0 falls down and then rises up within the same syllable. For example, a fall-rise intonation is commonly used in "un" utterances for expressing a denial.
In the present research, we proposed a method for detecting fall-rise movements, by searching for negative F0 slopes in the syllable nucleus, and positive F0 slopes in the syllable end portion. Here, syllable nucleus is defined as the 25 % to 75 % center portion of the syllable duration, while the syllable end is defined as the 40 % to 90 % portion of the syllable. The initial and final portions of the syllable are removed from the slope searching procedure, in order to avoid misdetection of F0 movements due to co-articulation effects.
If a fall-rise movement is detected, the syllable is divided in three portions of equal length. The representative F0 value of the first portion is estimated as the average F0 value (F0avg3a). For the second portion, the minimum F0 value (F0min3b) is estimated. Finally, for the last portion, a target value (F0tgt3c) is estimated in the same way of F0tgt2b. Then, two F0move values are estimated according to the expressions

$$F0move3a = F0min3b - F0avg3a, \quad (3)$$

$$F0move3b = F0tgt3c - F0min3b, \quad (4)$$

representing the falling and rising degrees, respectively. Fig. 4 shows a schematic procedure for F0move estimation.

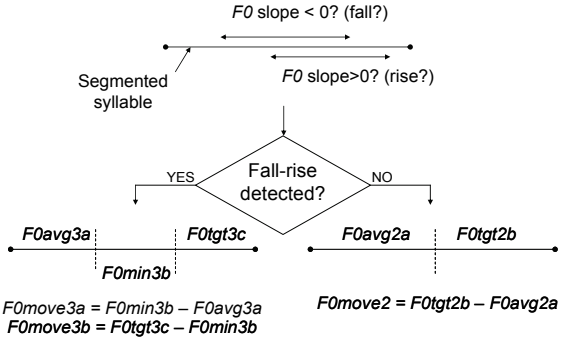


Fig. 4. Schematic procedure for estimation of F0move parameters in the monosyllabic utterances.

Fall-rise tones were correctly detected in all "un" utterances expressing denial. They were also detected in two "e" utterances. However, in these two cases, the F0move values of the falling movement were smaller than 2 semitones, and were not perceived as a fall-rise movement. In contrast, the F0move values for the "un" utterances expressing denial were all larger than 3 semitones, and clearly perceived as fall-rise tones.

For utterance duration, the manually segmented boundaries could be directly used, since the utterances are monosyllabic. However, as the manual segmentation may contain some silence (non-speech) portions close to the segmentation boundaries, an automatic procedure was further conducted, by estimating the maximum power of the syllable, and moving the boundaries until the power becomes 20 dB weaker than the maximum power. The newly segmented boundary intervals were used as segmental duration.

3.2 Detection of vocal fry (creaky voice): PPw, IFP, IPS

Vocal fry or creaky voices are characterized by the perception of very low fundamental frequencies, where individual glottal pulses can be heard, or by a rough quality caused by an alternation in amplitude, duration or shape of successive glottal pulses.

In the present research, we use the algorithm proposed in Ishi et al. (2005) for detection of vocal fry segments. A simplified block diagram of the detection algorithm is shown in Fig. 5. The algorithm first searches for power peaks in a "very short-term" power contour (obtained by using 4 ms frame length each 2 ms), which reflects the impulse-like properties of the glottal pulses in very low fundamental frequencies, characteristic of vocal fry signals. Then, it checks for constraints of periodicity and similarity between successive glottal pulses.

The periodicity constraint is to avoid the misdetection of a modal (periodic) segment between two glottal pulses. The similarity constraint is to avoid the misdetection of impulsive noises, assuming that during speech the vocal tract moves slowly enough so that the shapes of consecutive glottal pulses are similar.

The algorithm depends basically on three parameters.

- PPw : power thresholds for detection of power peaks in the very short-term power contour;
- IFP: intra-frame periodicity, which is based on the normalized autocorrelation function;
- IPS: inter-pulse similarity, which is estimated as a cross-correlation between the speech signals around the detected peaks.

Here, vocal fry segments are detected by using PPw larger than 7 dB, IFP smaller than 0.8, and IPS larger than 0.6. These thresholds are based on the analysis results reported in Ishi et al. (2005). Details about the evaluation of each parameter can be found in Ishi et al. (2005).

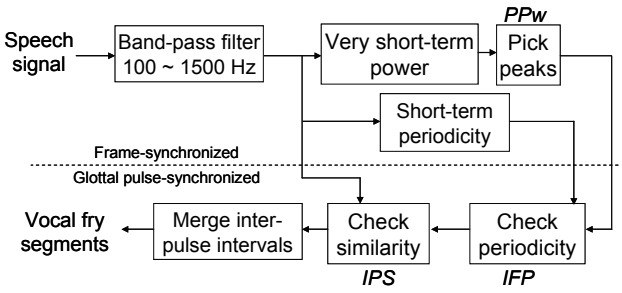


Fig. 5. Simplified block diagram of the vocal fry detection procedure.

3.3 Detection of pressed voice: H1'-A1'

Creaky voice (vocal fry) utterances may be pressed or lax. Lax creaky voices occur when the vocal folds are relaxed and are related to boredom or sadness (Gobl & Ní Cassaide, 2003). On the other hand, pressed creaky voices indicate strong attitudes/feelings of admiration or suffering (Sadanobu, 2004) in Japanese. Therefore, detection of pressed voices is necessary for PI disambiguation.

The production mechanism of pressed voice is not clearly explained yet, but it is thought that pressed voice has features similar with "tense voices" (Gordon & Ladefoged, 2001). A difference between "tense voice" and "lax voice" is reported to appear in the spectral tilt, since in tense voices, the glottal excitations become more impulse-like, and the higher frequency components are emphasized in relation to the fundamental frequency component. Acoustic parameters like H1-H2 and H1-A1 (Gordon & Ladefoged, 2001), and H1-A3 (Hanson, 1997) are proposed to reflect the effects of spectral tilt, where H1 is the amplitude power of the first harmonic (fundamental frequency), H2 is the amplitude power of the second harmonic, and A1 and A3 are the amplitude powers of the harmonic closest to the first and third formant, respectively.

However, in creaky or harsh voices, the irregularities in periodicity cause disturbances in the harmonic structure of their spectrum, so that it becomes difficult or unviable to extract harmonic components from the spectrum. In the present research, when periodicity is not detected, instead of H1, we use the maximum peak power of a low frequency band of 100 to 200 Hz (H1'). Also, as an automatic formant extraction is difficult, instead of A1, we use the maximum peak power in the frequency band of 200 to 1200 Hz (A1'), where the first formant is likely to appear. If periodicity is detected, H1' is equalized to H1. H1'-A1' values are estimated for each frame. Preliminary experiments indicate that pressed voice can be detected, when H1'-A1' is smaller than -15 dB.

3.4 Detection of aspiration noise: F1F3syn, A1-A3

Aspiration noise refers to turbulent noise caused by an air escape at the glottis, due to insufficient closure of the vocal folds during whispery and breathy phonations. Although there is a distinction between whispery and breathy voices from a physiological viewpoint (Laver, 1980), a categorical classification of voices in whispery or breathy is difficult in both acoustic and perceptual spaces (Kreiman & Gerratt, 2000). Further, aspiration noise can also occur along with harsh voices, composing the harsh whispery voices (Laver, 1980). In the present research, we use a measure of the degree of aspiration noise as indicative of such voice qualities.

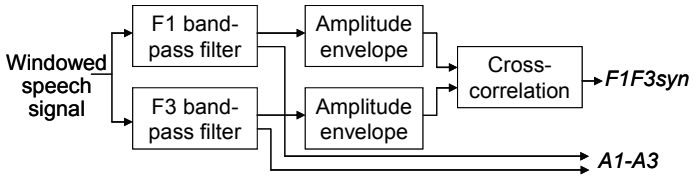


Fig. 6. Simplified block diagram of the acoustic parameters for aspiration noise detection.

The aspiration noise detection is based on the algorithm proposed in Ishi (2004), and its block diagram is shown in Fig. 6. The algorithm depends basically on two parameters.

- F1F3syn: synchronization measure between the amplitude envelopes of the signals in the first and third formant bands;
- A1-A3: difference (in dB) of the amplitudes of the signals in the first and third formant bands.

The main parameter, called F1F3syn, is a measure of synchronization (using a cross-correlation measure) between the amplitude envelopes of the signals obtained by filtering the input speech signal in two frequency bands, one around the first formant (F1) and another around the third formant (F3). This parameter is based on the fact that around the first formant, the harmonic components are usually stronger than the noisy component in modal phonation, while around the third formant, the noisy component becomes stronger than the harmonic components in whispery and breathy phonations (Stevens, 2000). Thus, when aspiration noise is absent, the amplitude envelopes of F1 and F3 bands are synchronized, and F1F3syn takes values close to 1, while if aspiration noise is present, the amplitude envelopes tend to be dissynchronized, and F1F3syn takes values closer to 0.

The second parameter, called A1-A3, is a measure of the difference (in dB) between the powers of F1 and F3 bands. This parameter is used to constrain the validity of the F1F3syn measure, when the power of F3 band is much lower than that of F1 band, so that aspiration noise could not be clearly perceived. Thus, when A1-A3 is big (i.e., the power of F1 band is much stronger than the power of F3 band), it is possible that the noisy components of F3 band are not perceived, and consequently, there is no sense to evaluate the F1F3syn measure.

The F1 band is set to 100 ~ 1500 Hz, while the F3 band is set to 1800 ~ 4500 Hz. The amplitude envelopes are obtained by taking the Hilbert envelope (Schroeder, 1999) of the signals filtered in each frequency band. Aspiration noise is detected for each frame, when F1F3syn is smaller than 0.4 and A1-A3 is smaller than 25 dB. These thresholds are based on the analysis results reported in Ishi (2004). More details about the evaluation of the method can be found in Ishi (2004).

3.5 Detection of harsh voice

A very simple procedure is adopted in the present research for detection of aperiodicity which is characteristic of harsh voices. The aperiodicity of harsh voices is here detected when neither periodicity nor vocal fry is detected. Note that vocal fry is usually aperiodic but does not sound harsh, so the aperiodicity in vocal fry segments has to be eliminated. Further, the initial and final 3 frames (30 ms) of each utterance are also eliminated, for avoiding the misdetection of aperiodicity due to effects of F0 disturbances at the onset and offset of the syllables.

Note that such simple procedure is valid, since we are evaluating only monosyllabic utterances and assuming that the voiced segments are known. Otherwise, the development of a more elaborated algorithm will be necessary for detecting harshness.

3.6 Evaluation of automatic detection of voice qualities

Fig. 7 shows a summary of the results for automatic detection of the voice qualities discussed in the previous sections.

The detection of creaky voice (or vocal fry) is evaluated by an index called VFR (Vocal Fry Rate), defined as the duration of the segment detected as vocal fry (VFdur) divided by the total duration of the utterance. Fig. 7 shows the results of detection of creaky segments, by using a criterion of VFR is larger than 0.1. We can note that all creaky segments are correctly detected (about 90% for c, c?), with only a few insertions (non c).

For evaluating pressed voice detection, an index called PVR (Pressed Voice Rate) is defined as the duration of the segment detected as pressed (PVdur), divided by the utterance duration. An utterance is detected as pressed, if PVR is larger than 0.1, and PVdur is larger than 100 ms, indicating that the segment has to be long enough to be perceived as pressed. 69 % of the pressed voice utterances were correctly identified in (p, pc, p?). Among them, most "e" utterances were correctly identified, while the detection failed in most of "un" utterances. This is probably because the nasal formant in "un" (around 100 to 300 Hz) increases the spectral power in the lower frequencies, consequently raising the H1'-A1' value. More robust acoustic features have to be investigated for detecting pressed voice in nasalized vowels.

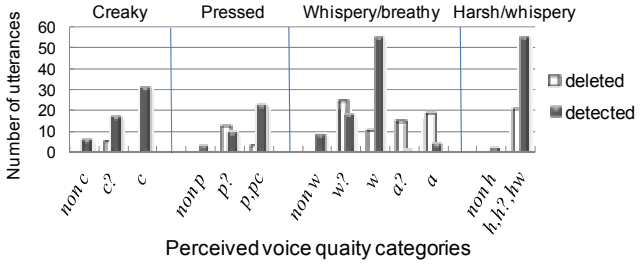


Fig. 7. Results of automatic detection of voice qualities, for each perceived category.

As in the previous voice qualities, an index called ANR (Aspiration Noise Rate) is defined as the duration of the segment detected as aspirated (ANdur), divided by the total duration of the utterance. Utterances containing aspiration noise are detected by using a criterion of ANR larger than 0.1. Most of the utterances where strong aspiration noise was perceived throughout the utterance (w) could be correctly detected (81%). However, for the utterances where aspiration noise was perceived in the syllable offsets (a? and a), most utterances could

not be detected by using ANR, as shown by the white bars in Fig. 8. This is because these syllable offset aspirations are usually unvoiced, and very short in duration. Other methods have to be investigated for the detection of the syllable offset aspirations.
Finally, regarding harsh and/or whispery voices, no clear distinction in functionality could be observed between harsh, harsh whispery and whispery voices (h, hw, w, a), as shown in Fig. 3. All these voice qualities are then described by an index called HWR (Harsh Whispery Rate). HWR is defined as the summation of HVdur (duration of the segment detected as harsh voice) and ANdur, divided by the utterance duration. 73 % of the utterances perceived as harsh and/or whispery (h,h?,hw) could be detected by using HWR > 0.1, and only a few insertion errors were obtained (non h), as shown in Fig. 7.

4. Discrimination of paralinguistic information based on intonation and voice quality-related prosodic features

In this section, we evaluate the contributions of intonation and voice quality-related prosodic features in "e"/"un" utterances, for discrimination of the PI items.

In 29 of the total of 370 utterances for evaluation, F0move could not be estimated due to missing F0 values. These missing values are due to non-modal phonations causing irregularities in the periodicity of the vocal folds. Fig. 8 shows a scatter plot of the intonation-related prosodic features (F0move vs. duration), excluding the utterances where F0move could not be obtained due to missing F0 values, and the ones where fall-rise intonation was detected.

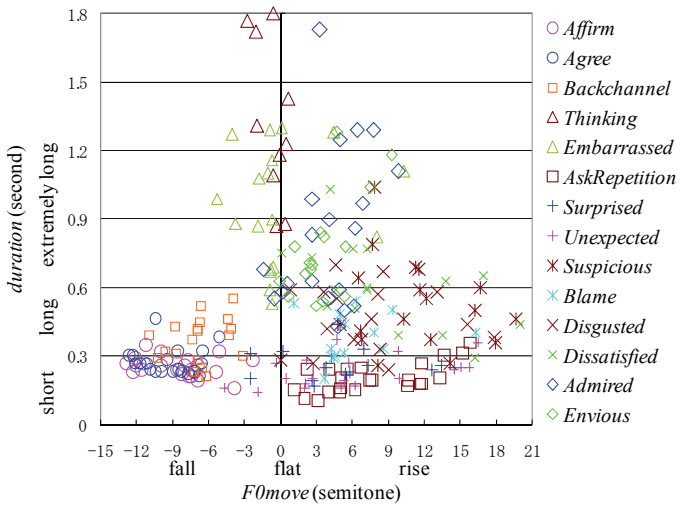


Fig. 8. Distribution of the intonation-related prosodic features (F0move vs. duration) for each PI.

Thresholds for F0move and duration were set, based on a preliminary evaluation of classification trees for discriminating the present PI items. A threshold of -3 semitones was set for F0move to discriminate falling tones (Fall), while a threshold of 1 semitone was set for rising tones (Rise). Utterances where F0move is between -3 and 1 semitone were considered as flat tones (Flat). The 29 utterances, where F0move could not be obtained, were

also treated as flat tones in the evaluation of automatic detection. Two thresholds were also set for duration. Utterances shorter than 0.36 seconds are called Short, while utterances with duration between 0.36 and 0.6 seconds are called Long. Utterances longer than 0.6 seconds are called extremely long (Ext.L).

Fig. 9 shows the distributions the prosodic categories (intonation and voice quality features) for each PI item. The discrimination of all PI items is difficult since many PI items share the same speaking styles. For example, there is no clear distinction in speaking style between Affirm and Agree, or between Surprised and Unexpected. The PI items which share similar speaking styles and which convey similar meanings in communication were then grouped (according to the perceptual evaluations in Section 2.2), for evaluating the automatic detection. Vertical bars in Fig. 9 separate the PI groups.

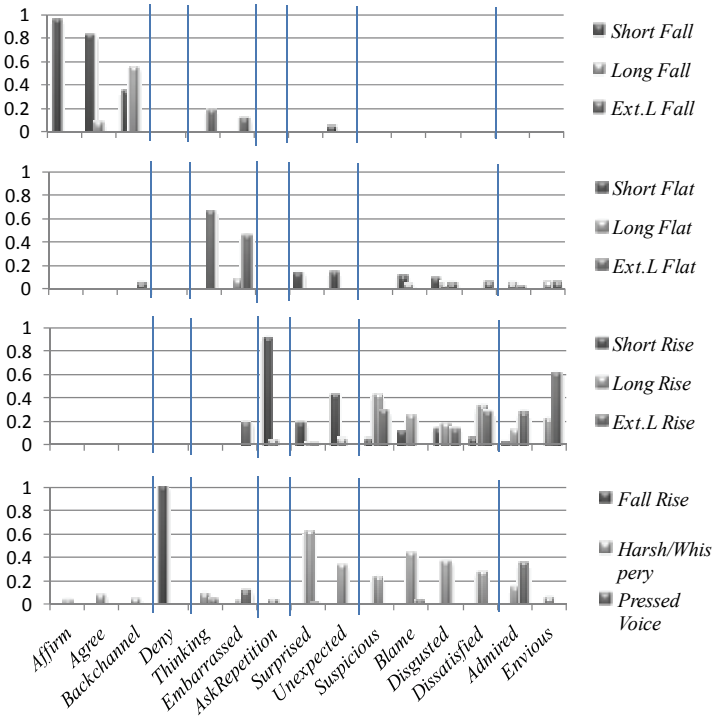


Fig. 9 Distribution of the prosodic categories for each PI item. Vertical bars separate PI groups.

Among the positive reactions, Affirm tends to be uttered by Short Fall intonation, while longer utterances (Long Fall) are more likely to appear in Backchannel. Extremely long fall or flat tones (Ext.L Fall, Ext.L Flat) were effective to identify Thinking. Note that the intonation-related prosodic features were effective to discriminate groups of PI items expressing some intentions or speech acts (Affirm/Agree/Backchannel, Deny, Thinking, and AskRepetition).

Short Rise tones can identify AskRepetition, Surprised and Unexpected, from the other PI items. Part of the Surprised/Unexpected utterances in Short Rise could be discriminated

from AskRepetition by the detection of harsh/whispery voice quality. However, many utterances in Surprised/Unexpected have similar speaking styles with AskRepetition. In these cases, context information would be necessary for their discrimination. The Fall-Rise tone detection was enough for the identification of Deny in the "un" utterances.

The big overlap of several PI items in the Rise tones shown in Fig. 8 resulted in lower discrimination between Surprised/Unexpected, Suspicious/Blame/Disgusted/Dissatisfied, and Admired/Envious, as shown in the second panel of Fig. 9. However, the use of voice quality features was effective to improve part of their detection rates. Note that the utterances where these non-modal voice qualities were detected pertain to the groups of PI items expressing strong emotions or attitudes.

Harsh and/or whispery voice detection (in the bottom panel of Fig. 9) was effective to disambiguate part of Surprised/Unexpected and AskRepetition sharing Short Rise tones, and Suspicious/Blame/Disgusted/Dissatisfied and Thinking/Embarrassed sharing Long and Extremely Long Flat tones.

Pressed voice detection was effective for identifying part of the utterances expressing admiration. The discrimination of Pressed utterances appearing in Disgusted and Embarrassed, from the ones in Admired, would need context information. However, it was observed that most utterances in Admired were "he", while most utterances in Disgusted and Embarrassed were "e", so that the detection of the aspirated consonant could be useful to discriminate part of the utterances.

Table 2 summarizes the detection rates without and with inclusion of voice quality features, for each PI group. Results indicate detection rates higher than 90 %, for Affirm/Agree/ Backchannel, Deny, and AskRepetition, regardless the use of voice quality features. For the three bottom PI groups (listed in Table 2) expressing strong emotions and attitudes, a significant improvement is obtained by the inclusion of voice quality features. However, the detection rates for Surprised/Unexpected and Suspicious/Blame/Disgusted/Dissatisfied were poor (41.9 % and 57.9 %). Improvements on the voice quality detection algorithms could still reduce part of these detection errors. However, most of the detection errors are thought to be due to the use of the same speaking style for different PI items, implying context dependency. Note that the confusions between PI items in Fig. 9 are basically the same as the ones obtained for the perceptual experiments in Section 2.2 (Fig. 2).

	Total	Detection rate (%) (without VQ)	Detection rate (%) (with VQ)	
Affirm/Agree/Backchannel	68	97.1	97.1	
Deny	12	100.0	100.0	
Thinking/Embarrassed	47	89.4	89.4	
AskRepetition	23	95.6	95.6	
Surprised/Unexpected	74	27.0	41.9	
Suspicious/Blame/Disgusted/Dissatisfied	88	38.6	57.9	83.6
Admired/Envious	58	39.7	63.8	
All PI items	370	57.3	69.2	86.2

Table 2. Detection rates of PI groups, without and with inclusion of voice quality (VQ) features.

The overall detection rate using simple thresholds for discrimination of the seven PI groups shown in Table 3 was 69.2 %, where 57.3 % was due to the only use of intonation-related prosodic features, while 11.9 % was due to the inclusion of voice quality parameters.
Finally, if the three PI groups Surprised/Unexpected, Suspicious/Blame/Disgusted/ Dissatisfied and Admired/Envious could be considered as a new group of PI items expressing strong emotions or attitudes, the detection rate of the new group would increase to 83.6 %, while the overall detection rate would increase to 86.2 %, as shown in the right-most column of Table 2. This is because most of the confusions in the acoustic space were among these three groups.

5. Conclusion

We proposed and evaluated intonation and voice quality-related prosodic features for automatic recognition of paralinguistic information (intentions, attitudes and emotions) in dialogue speech. We showed that intonation-based prosodic features were effective to discriminate paralinguistic information items expressing some intentions or speech acts, such as affirm, deny, thinking, and ask for repetition, while voice quality features were effective for identifying part of paralinguistic information items expressing some emotion or attitude, such as surprised, disgusted and admired. Among the voice qualities, the detection of pressed voices were useful to identify disgusted or embarrassed (for "e", "un"), and admiration (for "he"), while the detection of harsh/whispery voices were useful to identify surprised/unexpected or suspicious/disgusted/blame/dissatisfied.
Improvements in the detection of voice qualities (harshness, pressed voice in nasalized voices, and syllable offset aspiration noise) can still improve the detection rate of paralinguistic information items expressing emotions/attitudes.
Future works will involve improvement of voice quality detection, investigations about how to deal with context information, and evaluation in a human-robot interaction scenario.

6. Acknowledgements

This research was partly supported by the Ministry of Internal Affairs and Communications and the Ministry of Education, Culture, Sports, Science and Technology-Japan. The author thanks Hiroshi Ishiguro (Osaka University), Ken-Ichi Sakakibara (NTT) and Parham Mokhtari (ATR) for advice and motivating discussions.

7. References

Campbell, N., & Erickson, D. (2004). What do people hear? A study of the perception of non-verbal affective information in conversational speech. *Journal of the Phonetic Society of Japan*, 8(1), 9-28.

Campbell, N., & Mokhtari, P. (2003). Voice quality, the 4th prosodic dimension. In *Proceedings of 15th International Congress of Phonetic Sciences (ICPhS2003),* Barcelona, (pp. 2417-2420).

Dang, J., Honda, K. (1966). Acoustic characteristics of the piriform fossa in models and humans. *J. Acoust. Soc. Am.*, 101(1), 456-465.

Erickson, D. (2005). Expressive speech: production, perception and application to speech synthesis. *Acoust. Sci. & Tech.*, *26*(4), 317-325.

Fernandez, R., & Picard, R.W. (2005). Classical and novel discriminant features for affect recognition from speech. In *Proceedings of Interspeech 2005*, Lisbon, Portugal (pp. 473-476).

Fujie, S., Ejiri, Y., Matsusaka, Y., Kikuchi, H., & Kobayashi, T. (2003) Recognition of paralinguistic information and its application to spoken dialogue system. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding* (*ASRU '03*), St. Thomas, U.S. (pp. 231-236).

Fujimoto, M., & Maekawa, K. (2003) Variation of phonation types due to paralinguistic information: An analysis of high-speed video images. In *Proceedings of 15th International Congress of Phonetic Sciences (ICPhS2003),* Barcelona, (pp. 2401-2404).

Gobl, C., & Ní Chasaide, A. (2003). The role of voice quality in communicating emotion, mood and attitude. *Speech Communication*, 40, 189-212.

Gordon, M., & Ladefoged, P. (2001). Phonation types: a cross-linguistic overview. *J. of Phonetics*, 29, 383-406.

Hanson, H. (1997). Glottal characteristics of female speakers: Acoustic correlates. *J. Acoustic Society of America*, 101, 466-481.

Hayashi, Y. (1999). Recognition of vocal expression of emotions in Japanese: using the interjection "eh". In *Proceedings of ICPhS 99*, San Francisco, USA (pp. 2355-2359).

Hess, W. (1983). Pitch Determination of Speech Signals. Vol. 3 of *Springer Series of Information Sciences*, Berlin, Heidelberg, New York: Springer-Verlag.

Imagawa, H., Sakakibara, K., Tayama, N., & Niimi, S. (2003). The effect of the hypopharyngeal and supra-glottic shapes for the singing voice. In *Proceedings of the Stockholm Music Acoustics Conference (SMAC 2003)*, II, (pp. 471-474).

Ishi, C.T. (2004). A new acoustic measure for aspiration noise detection. In *Proceedings of Interspeech 2004-ICSLP*, Jeju, Korea (pp. 941-944).

Ishi, C.T. (2005). Perceptually-related F0 parameters for automatic classification of phrase final tones. *IEICE Trans. Inf. & Syst.*, E88-D(3), 481-488.

Ishi, C.T., Ishiguro, H., & Hagita, N. (2005). Proposal of acoustic measures for automatic detection of vocal fry. In *Proceedings of Interspeech 2005*, Lisbon, Portugal (pp. 481-484).

Ito, M. (2004). Politeness and voice quality – The alternative method to measure aspiration noise. In *Proceedings of Speech Prosody 2004*, Nara, Japan (pp. 213-216).

JST/CREST ESP Project homepage, http://feast.atr.jp/esp/esp-web/

Kasuya, H., Yoshizawa, M., & Maekawa, K. (2000). Roles of voice source dynamics as a conveyer of paralinguistic features. In *Proceedings of International Conference on Spoken Language Processing (ICSLP2000)*, Beijing, (pp.345-348).

Kitamura, T., Honda, K., Takemoto, H. (2005). Individual variation of the hypopharyngeal cavities and its acoustic effects. Acoust. Sci. & Tech 26(1), 16-26.

Klasmeyer, G., & Sendlmeier, W.F. (2000). Voice and Emotional States. In R.D. Kent & M.J. Ball (Eds.), *Voice Quality Measurement*, San Diego: Singular Thomson Learning, 339-358.

Kreiman, J., & Gerratt, B. (2000). Measuring vocal quality. In R.D. Kent & M.J. Ball (Eds.), *Voice Quality Measurement*, San Diego: Singular Thomson Learning, 73-102.

Laver, J. (1980). Phonatory settings. In *The phonetic description of voice quality*. Cambridge: Cambridge University Press, 93-135.

Maekawa, K. (2004). Production and perception of 'Paralinguistic' information. In *Proceedings of Speech Prosody 2004*, Nara, Japan (pp. 367-374).

Neiberg, D., Elenius, K., & Laskowski, K. (2006). Emotion recognition in spontaneous speech using GMMs. In *Proceedings of Interspeech 2006*, Pittsburgh, USA (pp. 809-812).

Nwe, T.L., Foo, S.W., & De Silva, L.C. (2003). Speech emotion recognition using hidden Markov models. *Speech Communication* 41, 603-623.

Sadanobu, T. (2004). A natural history of Japanese pressed voice. *J. of Phonetic Society of Japan*, 8(1), 29-44.

Schroeder, M.R. (1999). Hilbert envelope and instantaneous frequency. In *Computer speech – Recognition, compression, synthesis*, Berlin: Springer, 174-177.

Schuller, B., Muller, R., Lang, M., & Rigoll, G. (2005). Speaker independent emotion recognition by early fusion of acoustic and linguistic features within ensembles. In *Proceedings of Interspeech 2005*, Lisbon, Portugal (pp. 805-808).

Stevens, K. (2000). Turbulence noise at the glottis during breathy and modal voicing. In *Acoustic Phonetics*. Cambridge: The MIT Press, 445-450.

Voice quality sample homepage, http://www.irc.atr.jp/~carlos/voicequality/

22

Psychological Motivated Multi-Stage Emotion Classification Exploiting Voice Quality Features

Marko Lugger and Bin Yang
University of Stuttgart
Germany

1. Introduction

Paralinguistic properties play a more and more decisive role in recent speech processing systems like automatic speech recognition (ASR) or natural text-to-speech systems (TTS). Besides the linguistic information, the so called paralinguistic properties can help solving ambiguities in man-machine-interaction. Nowadays, such man-machine-interfaces can be found for example in call-centers, in driver assistance systems, or at the personal computer at home. There are many different applications for the recognition of paralinguistic properties, e.g. gender, age, voice quality, emotion, or alcohol consumption. Among these properties, the emotional state of a speaker has a superior position because it strongly affects the acoustic signal produced by the speaker in all kind of conversational speech. Emotion recognition has its applications in various fields e.g. in call centers to detect angry customers, in entertainment electronics, in linguistics, and even in politics to analyse speeches of politicians to train the candidates for election campaigns.
Various attempts show quite good results in the case of speaker dependent classification (Lugger & Yang, 2006; McGilloway et al., 2000; Nogueiras et al., 2001). But the hardest task and also the most relevant in practice is the speaker independent emotion recognition. Speaker independent means that the speaker of the classified utterances is not included in the training database of the system. He is unknown for the classifier and the deduced learning rules in the training phase. Up to now, a good speaker independent emotion recognition could only be achieved by using very large feature sets in combination with very complex classifiers (Schuller et al., 2006; Lee & Narayanan, 2005). In (Schuller et al., 2006), an average recognition rate of 86.7% was achieved for seven emotions by using 4000 features and support vector machine as classifier.
In this work, our goal is to further improve the classification performance of the speaker independent emotion recognition and make it more applicable for real-time systems. Therefore, we use the same German database consisting of 6 basic emotions: sadness, boredom, neutral, anxiety, happiness, and anger. For lack of relevance disgust is ignored. But in contrast to other approaches, we focus on the extraction of less but more adapted features for emotion recognition. Because for real-time systems the feature extraction is the most time consuming part in the whole process chain, we try to reduce the number of features. At the same time we study multi-stage classifiers to optimally adjust the reduced feature number to the different class discriminations during classification. In comparison to support vector machines or neural networks, the Bayesian classifier we use can be

implemented on processors with lower computational power. By using only 346 features and a multi-stage Bayesian classifier, we achieve improved results by dramatically reducing computational complexity.
We improve the speaker independent emotion recognition in two ways. First, we propose a novel voice quality parameter set. It is an extension of the parameter set reported in (Lugger et al., 2006). We observed that one can exploit the existence of different phonation types within the human speech production for emotion classification. In our study, we show that our voice quality parameters outperform mel frequency cepstral coefficients in the application of emotion recognition. We further investigate how prosodic and voice quality features overlap or complement each other. Second, our observation that the optimal feature set strongly depends on the emotions to be classified, leads to a hierarchical emotion classification strategy. The best results are achieved by a classification that is motivated by the psychological model of emotion dimensions. After activation recognition in the first stage, we classify the potency and evaluation dimension in following classification stages. A 2-stage and a 3-stage hierarchical classification approach are presented in this article. For each classification, the optimal feature set is selected separately.
This chapter is organized as follows: First, the theory of emotion and the relevance of the database used in this study are discussed in section 2. Then, the relevant acoustic features for different emotion dimensions are introduced in section 3. The performance of the different feature groups is studied and voice quality parameters are compared with mel frequency cepstral coefficients. In section 4, the results of classifying six emotions using different strategies and combinations of feature sets are presented. Finally, some conclusions are drawn.

2. Emotion definitions

Emotion theory has been an important field of research for a long time. Generally, emotions describe subjective feelings in short periods of time that are related to events, persons, or objects. There are different theoretical approaches about the nature and the acquisition of emotions (Cowie and Douglas-Cowie, 2001). Since the emotional state of humans is a highly subjective experience it is very hard to find objective definition or universal terms. That is why there are several approaches to model emotions in the psychological literature (Tischer, 1993). The two most important approaches are the definition of discrete emotion categories, the so called basic emotions, and the utilization of continuous emotion dimensions. These two approaches can also be utilized for the application of automatic emotion recognition. The two models result in different advantages and disadvantages for automatic emotion recognition. The usage of emotion dimensions has the advantage that we can find acoustic features which are directly correlated with certain emotion dimensions. But in listening tests, which are used to obtain a reference for the acoustic data, it is hard for a proband to work with different dimensions. In this case, it is more appropriate to use basic emotions. In the following, the two approaches are briefly explained.

2.1 Categorical emotion approach

Within this approach, the emotional state of a speaker during natural conversation is defined by discrete classes. Ekman defined so called basic emotions (Ekman, 1972). These are happiness, sadness, anger, anxiety, boredom, disgust, and neutral. More detailed emotions can be designed by mixtures of the basic emotions. In our work we use the same

basic emotions as defined by Ekman except for disgust. On the one hand, we benefit from the fact that people are familiar with these terms. But on the other hand, there is also a lack of differentiation possibility. For example there is hot anger and cold anger, or silent sadness and whiny-voiced sadness where the basic emotion model does not distinguish between. Generally, there are no acoustic features that are directly correlated with a single basic emotion.

2.2 Dimensional emotion approach

The second approach of psychological emotion research says that we can locate different emotions in a two- or three-dimensional space (Schlosberg, 1954). The most often used dimensions are activation (arousal), potency (power), and evaluation (pleasure), see Figure 1. As we will see below, most of the features used in acoustical emotion recognition, mainly prosodic features, describe the activation dimension. This is why emotions which do not obviously differ in the activation dimension can not be well separated by classical acoustic features. These are, for example, anger, happiness, and anxiety with a high activation or neutral, boredom, and sadness with a low activation level. So our task is to find novel acoustic features that describe more the other dimensions, e.g. the evaluation to distinguish between positive and negative emotions and the potency discriminating dominant and submissive emotions.

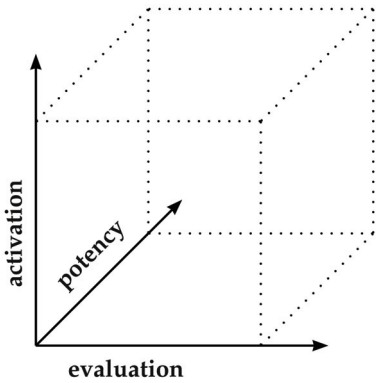


Fig. 1. Three-dimensional emotional space

2.3 Emotions and voice quality

Besides the standard prosodic aspects, voice quality is an important factor in conveying emotional information (Klasmeyer & Sendlmeier, 2000). That is why voice quality is also called as the 4th dimension of prosody (Campbell & Mokhtari, 2003). Listening tests showed that there is a strong relation between voice quality and emotion perception (Gobl & Ni Chasasaide, 2003; Yanushevskaya et al., 2008). Many speakers express their emotional state by altering their voice quality. This can happen consciously by supporting their affective expression by changing the voice quality from modal to nonmodal. But most of the times it is an unconscious process, where the glottal muscles are influenced by the affective state of the speaker and the glottal phonation process is affected indirectly. The here studied emotions differ considerably in the utilized phonation type. The following relations between emotions and phonation types can be observed: For the production of a sad and disgusted

emotional state, a creaky phonation is often used. Rough voice is usually used to support an angry emotion. The anxious emotion shows sometimes parts of breathy voice. To express happiness as well as the neutral emotional state, the modal voice quality is exclusively used.

2.4 Emotional database

In emotion recognition the used database plays a crucial role. The most important databases are listed in (Ververidis & Kotropoulos, 2003). Thereby, we have to distinguish between three kinds of databases: acted speech, elicited speech, and spontaneous (natural) speech. In this study we used a well known German database of acted speech, called the Berlin emotional database (Burkhardt et al., 2005). It provides short acted utterances approximately between two and five seconds of length. We try to classify six emotions: anger, happiness, sadness, boredom, anxiety, and neutral. There are 694 utterances, which are more than 100 patterns per emotion. We use a combination of both categorical and dimensional emotion definition, by locating 6 basic emotions in a 3-dimensional emotion space. Figure 2 shows the six basic emotions and their location in the three dimensional emotion space.

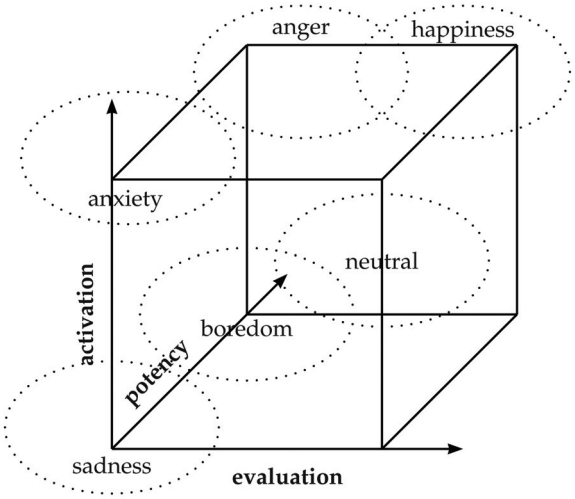


Fig. 2. 3-dimensional emotion space and 6 basic emotions

3. Acoustic features

Discrete affective states experienced by the speaker are reflected in specific patterns of acoustic cues in the speech (Lewis et al., 2008). This means, information concerning the emotional state of a speaker is encoded in vocal acoustics and subsequently decoded by the receiving listeners. For automatic emotion recognition two basic tasks occur. The first is to find the manner how speaker encode their emotional state in the speech. This problem is basically the extraction of emotion correlated features from the acoustic speech signal. After that, the second task is to solve a pattern recognition problem to decode the emotional state from the extracted speech features. This second problem is discussed in section 4.

In the field of emotion recognition mainly suprasegmental features are used. The most important group is called prosodic features. Sometimes segmental spectral parameters as mel

frequency ceptral coefficients (MFCC) are added. But according to (Nwe et al., 2003), MFCC features achieve poor results for emotion recognition. In our approach, the common prosodic features are combined with a large set of so called voice quality parameters (VQP). Their performance in speaker independent emotion classification is compared with that of MFCC parameters and their contribution in addition to the standard prosodic features is studied.

3.1 Prosodic features

There are three main classes of prosodic features: pitch, energy, and duration. Pitch is describing the intonation of a sentence. The intensity of the uttered words is covered by energy features. Duration stands for the speed of talking and for the number of pauses. Two more classes that do not belong directly to prosody are articulation (formants and bandwidths) and zero crossing rate. These deduced features are obtained by measuring statistical values of their corresponding extracted contours. Mean, median, minimum, maximum, range, and variance are the most used measurements. So they are describing both the mean level and the variability of the basic features. All together we extracted 201 prosodic features from the speech signal.

3.2 Voice quality parameters

As stated in subsection 2.3, phonation is one aspect besides articulation and prosody in generating emotional coloured speech. The theory of voice quality estimation is based on the source filter model of speech production (Fant, 1960). In this model the speech signal is assumed as a source signal produced at the glottis that is filtered by a system defined by the supralaryngeal setting of the vocal tract. Whereas the process at the glottis is called phonation, articulation is the mechanism of producing different sounds by the vocal tract. In contrast to other spectral speech features, the voice quality parameters (VQP) describe basically the properties of the phonation process. By inverse filtering, the influence of articulation is compensated to a great content. Thus, the parameter values specify the phonation type that is used by the speaker. The feature set we use is a parameterization of the voice quality in the frequency domain. We extract various gradients in the excitation spectrum. The method of using spectral gradients was first introduced by Stevens and Hanson (Stevens & Hanson, 1994). The detailed computation is given in the following sections. The definition and the robustness of VQP are also reported in (Lugger et al., 2006). Altogether there are 67 voice quality parameters. As we will see later in section 4, the VQP parameters have an obvious contribution to the discrimination of different emotions beyond the prosodic features.

3.2.1 Measurement of basic speech features

First, we estimate some well known basic speech features from windowed, voiced segments of the speech signal, see Table 1. We perform the voiced-unvoiced decision and the pitch estimate F_0 according to the RAPT algorithm (Talkin et al., 1995) that looks for peaks in the normalized cross correlation function. As measuring points for the spectral gradients, we use higher harmonics. To get a fixed number of 20 gradients for all pitch frequencies, we extract the harmonics F_{pk} next to fixed frequencies at multiples of 200 Hz. So all together 21 harmonics are used, which cover the relevant frequency range for voice quality up to 4000 Hz. The frequencies and bandwidths of the first four formants are estimated by an LPC analysis (Talkin, 1987).

feature	meaning
$F_{p0} = F_0$	pitch
F_1, F_2, F_3, F_4	formant frequencies
B_1, B_2, B_3, B_4	formant bandwidths
$F_{p0}, \ldots, F_{p20}$	frequency of harmonics
$H_0, \ldots, H_{20}$	amplitude at $F_{p0}, \ldots, F_{p20}$ [dB]

Table 1. Basic speech features for estimation of spectral gradients

3.2.2 Compensation of the vocal tract influence

Since the voice quality parameters shall only depend on the excitation and not on the articulation process, the influence of the vocal tract has to be compensated. This is done by subtracting terms which represent the vocal tract influence from the amplitudes of each harmonic H_k as described in (Lugger et al., 2006). The amplitudes of the compensated harmonics are $\tilde{H}_k$.

3.2.3 Estimation of the voice quality parameters

Up to now only 4 spectral gradients were used to characterize the glottal source signal. In order to better parameterize the glottal excitation signal the parameter set is extended to 20 gradients. Figure 3 illustrates the definition of the spectral gradients.

$$\mathrm{SG_k} = \frac{\tilde{H}_0 - \tilde{H}_k}{F_{pk} - F_{p0}} \qquad (k = 1, \ldots, 20) \tag{1}$$

In addition to these 20 gradients normalized to the linear frequency difference $\Delta f_k = F_{p(k)} - F_{p0}$, the same amplitude differences $\tilde{H}_0 - \tilde{H}_k$ are also normalized to frequency differences in both octave and in bark scale. Octave is a logarithmic scale

$$octave\,(k) = \log_2 \frac{F_{pk}}{F_{p0}} \tag{2}$$

and the bark scale is based on the human auditory system:

$$bark\,(\Delta f_k) = 13 \tan^{-1}(0.00076 \cdot \Delta f_k) + 3.5 \tan^{-1}\left(\left(\frac{\Delta f_k}{7500}\right)^2\right) \tag{3}$$

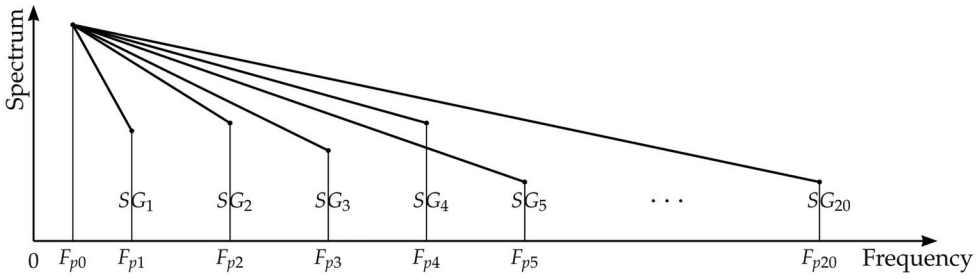


Fig. 3. Spectral gradients at fixed frequencies

In addition, the four formant bandwidths B_n normalized to the corresponding formant frequencies F_n are calculated.

$$\mathrm{IC_n} = \frac{B_n}{F_n} \qquad (n = 1, \ldots, 4) \tag{4}$$

The last three voice quality parameters describe the voicing, the harmonicity, and the periodicity of the signal, see (Lugger & Yang, 2006). In total, we obtain a set of 67 voice quality features.

3.3 Mel frequency cepstral coefficients

The cepstrum of a signal is the inverse Fourier transform of the logarithm of the Fourier transform. In comparison to the standard cepstrum, MFCC uses frequency bands which are positioned logarithmically based on the mel scale motivated by the human auditory system. MFCC is the state of the art spectral parameter set in automatic speech recognition. According to (Nwe et al., 2003), its performance is, however, not satisfying for emotion recognition. So we want to know whether VQP is a better alternative than MFCC for emotion recognition. For this comparison, we use in our study the mean value as well as the 2nd to the 6th central moments of 13MFCC. The total number of MFCC features is thus 78. The implementation we use was first published in (Orsak & Etter, 1995).

3.4 Feature selection

There are two main reasons for reducing the number of features from the original set. First, the number of training patterns had to be enormous if we want to use all features. Second, the training and classification would take a long time when using the whole feature set. So for all the classifications, the original number of 346 features is reduced by using an iterative selection algorithm. After the selection process the final feature number is reduced to 25 because for this feature number a local maximum in the classification rate was observed. We used the sequential floating forward selection algorithm (SFFS). It is an iterative method to find a subset of features that is near the optimal one. It was first proposed in (Pudil et al., 1994). In each iteration, a new feature is added to the subset of selected features and afterwards the conditionally least significant features are excluded. This process is repeated until the final dimension is obtained. As selection criterion the speaker independent recognition rate is used. In combination with the Bayesian classifier it turns out to be an efficient method for optimizing a feature set to a specific classification problem.

3.5 Comparison of VQP and MFCC

Now the performance of VQP is compared to that of MFCC for the recognition of emotions. Figure 4 shows the average classification rates of six emotions when using prosodic features only and when combining them with MFCC and/or VQP. In each of the four cases, feature sets with increasing number (up to 25) are selected by SFFS. A flat 1-stage classification is used. That means, the discrimination of all 6 emotions is performed by using only one classification. As we see, the classification rate by using additional MFCC is higher than using only prosodic features. But the classification rate when combining prosodic features with VQP outperforms that when combining with MFCC. In comparison to prosodic

features only, a gain of at least 3% is achieved. Adding both VQP and MFCC to prosodic features brings no noticeable improvement.
As we have seen, the voice quality parameters outperform the MFCC for emotion recognition because they are predestined for the recognition of different voice qualities that are used in emotion production. But two questions arise that we would like to answer in the sequel: Do the voice quality parameters contain some new information that is not included in the prosodic features? And how can we optimally combine both feature types to get the best classification result?

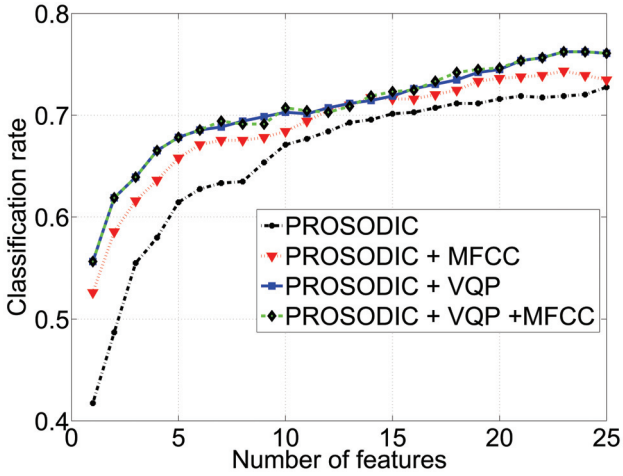


Fig. 4. Comparison between VQP and MFCC

4. Classification

In this section, the relationship between prosodic and voice quality features is studied and different classification strategies using a combined feature set are presented. First, we compare the classification rate of using only prosodic features with that of combining both feature types using a flat 1-stage classifier. Here, the gain of adding voice quality information to the prosodic information is investigated. After that, different strategies to optimally combine the information contained in both feature types are presented: a flat 1-stage classification, a hierarchical 2-stage, and a hierarchical 3-stage classification.
For all the classifications, a Bayesian classifier is used and the best 25 features are selected using SFFS. The speaker independent classification is performed by a "leaving-one-speaker-out" cross validation. The class-conditional densities are modelled as unimodal Gaussians. By using the Gaussian mixture model (GMM) with a variable number of Gaussians, we could not observe significant changes in the classification rate for this database. Mostly, only one Gaussian per feature and emotion was decided.
In our study, all the classification results are represented by confusion matrices. Every field is associated with a true and an estimated emotion. Thereby, the true emotion is given by the corresponding row. Every column stands for an estimated emotion. So the percentages of correct classification are located on the main diagonal whereas the other elements of the matrix represent the confusions.

4.1 Classification with prosodic features

First of all we classify with prosodic features only. The 201 features were reduced to an optimized set of 25 by using SFFS. The confusion matrix is shown in Table 2. The overall recognition rate of 72.8% is quite good, but there are mainly 3 problematic class pairs. The confusion between anger and happiness is with 26.2% respectively 11.8% unacceptably high. In general, happiness is least classified with a recognition rate of only 57.9%. Furthermore, the confusions between neutral and boredom as well as between happiness and anxiety are also unsatisfying. As we know from Figure 1, these emotions do not differ in the activation dimension and so prosodic features are not appropriate to distinguish between them.

emotion	happy	bored	neutral	sad	angry	anxious
happy	**57.9%**	1.9%	3.7%	0.0%	26.2%	10.3%
bored	1.8%	**79.3%**	7.2%	8.1%	1.8%	1.8%
neutral	7.8%	9.7%	**71.8%**	5.8%	1.0%	3.9%
sad	3.3%	13.3%	6.7%	**73.4%**	0.0%	3.3%
angry	11.8%	0.7%	0.7%	0.0%	**80.2%**	6.6%
anxious	15.9%	0.9%	2.7%	3.5%	6.2%	**70.8%**

Table 2. Classification with prosodic features only

On the other hand, the classification rate of angry, bored, and sad is quite good with 80.2%, 79.3%, respectively 73.4%. The reason is that in this database sadness is spoken very slowly and also with long pauses. On the opposite side, angry sentences are spoken very fast. Hence, duration features, one group of prosodic features, are well suitable to recognize sad and angry utterances. In German, boredom is realized with a very distinctive intonation contour. That is why pitch features, another group of prosodic features, can detect bored sentences very well. In general, we can state that by using prosodic features, the discrimination between high and low activation emotions is very good. The highest confusion between these two activation levels is only 7.8% (neutral vs. happy).

4.2 Classification with combined feature sets

Below, we classify with both prosodic and voice quality features. The result shown in Table 3 is the classification rate we would obtain by an ideal combination of a single prosodic and a single voice quality classifier. The prosodic classifier uses the best 25 prosodic features selected by SFFS and the voice quality classifier uses the best 25 voice quality parameters. *P* stands for the event "correctly classified by prosodic features" and *V* stands for the event "correctly classified by voice quality parameters". The second row in Table 3 shows the rate of patterns that are classified correctly by both classifiers. For the emotions happiness, boredom, and neutral the correctly classified patterns are quite disjoint, while for sadness, anger, and anxiety they are strongly overlapping. The third and fourth row show the patterns that are correctly classified by prosodic features but not by voice quality parameters and vice versa. In general, the prosodic classifier performs better. But for all the emotions except for boredom the voice quality classifier contributes to an improvement of more than 10%. Interestingly, for sadness and anxiety the voice quality classifier even outperforms the prosodic one. In the last row of Table 3 the overall classification rate for *P* OR *V* is given. This implies that we would have complete knowledge of which classifier performs correctly for every single given pattern. We only get a misclassification when both classifiers are

wrong. One can interpret this as a reference value for the classification rate with both feature sets. The result corresponds to an overall recognition rate of 86.2% that is at the level of human recognition rate. Clearly, the voice quality features improve considerably the classification beyond the prosodic information. The gain is biggest for the classes sadness and anxiety that make use of the nonmodal voice qualities creaky respectively breathy voice.

emotion	happy	bored	neutral	sad	angry	anxious
P AND V	31.7%	48.6%	40.8%	55.8%	61.0%	53.9%
P AND $\overline{V}$	26.2%	30.7%	31.0%	17.5%	19.1%	16.9%
$\overline{P}$ AND V	10.2%	8.1%	16.5%	18.4%	10.2%	17.8%
P OR V	**68.1%**	**87.4%**	**88.3%**	**91.7%**	**90.3%**	**88.6%**

Table 3. Reference value for the classification rate with prosodic and voice quality features

In practice we do not know which classifier performs correctly for every single pattern. So we have to define a general fusion method for all the patterns. In the following different strategies for the combination of prosodic and voice quality features are proposed. We will see that we can even exceed the reference value of 86.2% by a deliberate design of our classifier.

4.3 1-stage classification

In Table 4, the results using the best 25 features out of all (prosodic and voice quality features) are shown. Here, the features are jointly selected by SFFS. Among them, there are 18 prosodic and 7 voice quality features. With an overall recognition rate of 75.5%, this approach outperforms the results of Table 2. In general, the result is slightly better than using only prosodic features. But with this direct classification we are not able to exploit all the information that is contained in both feature sets, when we compare with Table 3. Only sadness and anxiety are distinctly improved by 9.1% respectively 11.5% To further benefit from the voice quality information we apply multi-stage classifications in the next sections.

emotion	happy	bored	neutral	sad	angry	anxious
happy	**58.9%**	1.9%	0.9%	0.0%	24.3%	14.0%
bored	0.0%	**79.3%**	8.1%	9.9%	1.8%	0.9%
neutral	1.9%	17.5%	**68.0%**	1.9%	1.9%	8.8%
sad	1.7%	9.2%	2.5%	**82.5%**	0.8%	3.3%
angry	12.5%	0.8%	0.0%	0.0%	**80.1%**	6.6%
anxious	8.0%	0.0%	1.8%	0.9%	7.0%	**82.3%**

Table 4. Classification with 18 prosodic and 7 voice quality features jointly selected by SFFS

4.4 Psychological motivated multi-stage classification

The main drawback of the previous approaches is that we do not consider which feature group classifies well for which emotions. Our investigation in (Lugger & Yang, 2007a) showed that the optimal feature set strongly depends on the emotions to be separated. This means, using one global feature set for the discrimination of all emotions is clearly suboptimal. This conclusion motivates a hierarchical classification strategy, consisting of different classification stages distinguishing different classes, and using different feature sets

(Lugger & Yang, 2007b). The fundamental observation that prosodic features are very powerful in discriminating different levels of activation and voice quality features perform better in discriminating the other emotion dimensions leads to the following multi-stage strategies. The stages chosen here are motivated by the emotion dimensions of the psychological model shown in Figure 1.

4.4.1 2-stage hierarchical classification

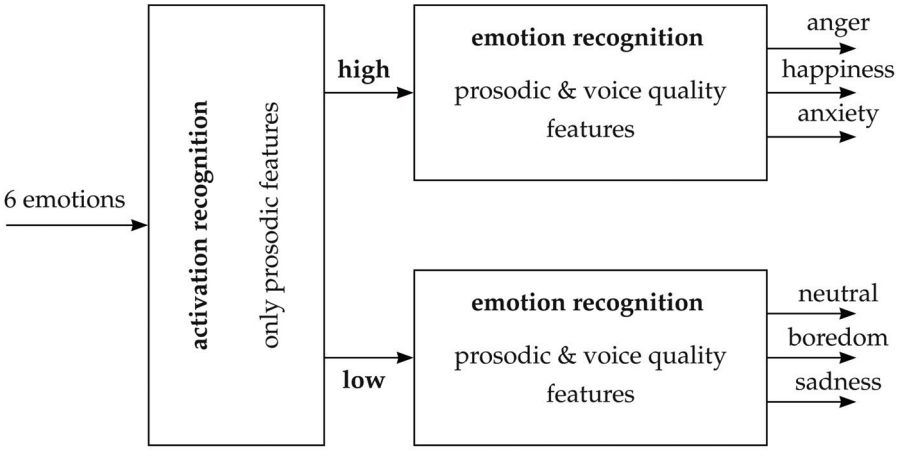


Fig. 5. 2-stage approach of emotion recognition

As shown in Figure 5, we separate the classification process in two stages. In the first stage, we classify for two different activation levels. One class including anger, happiness, and anxiety has a high activation level. The second class including neutral, boredom, and sadness has a low activation level. For this activation discrimination we achieve a very good classification rate of 98.8% on average with prosodic features only. Table 5 shows the corresponding confusion matrix. In this stage, we have observed that including voice quality features will not contribute to any improvements.

activation	high	low
high	**99.1%**	0.9%
low	1.8%	**98.2%**

Table 5. Classification of two activation levels

In the second stage, we classify the emotions inside each activation class. That means, all patterns that were classified to high activation in the first stage are classified to anger, happiness, and anxiety. Similarly, all patterns that were decided to have a low activation in the first stage were classified to neutral, boredom, and sadness. For the second stage, 2 instances of the joint SFFS-based combination of prosodic and voice quality features were used. Here, there is a clear advantage of including voice quality features in comparison to prosodic features only. Table 6 shows the classification results for the second stage. Needless to say, the few patterns that were incorrectly classified in the first stage cannot be corrected

in the second stage. By combining both stages in Figure 5, we obtain the overall confusion matrix shown in Table 7.

high activation				low activation			
emotion	**happy**	**angry**	**anxious**	**emotion**	**bored**	**neutral**	**sad**
happy	**67.3%**	21.5%	11.2%	**bored**	**91.0%**	2.7%	6.3%
angry	5.1%	**86.1%**	8.8%	**neutral**	15.0%	**81.0%**	4.0%
anxious	5.4%	3.6%	**91.0%**	**sad**	9.4%	1.7%	**88.9%**

Table 6. Classification of emotions with high or low activation

emotions	**happy**	**bored**	**neutral**	**sad**	**angry**	**anxious**
happy	**67.3%**	0.0%	0.0%	0.0%	21.5%	11.2%
bored	0.0 %	**91.0%**	2.7%	6.3%	0.0%	0.0%
neutral	0.0%	14.6%	**78.7%**	3.8%	0.0%	2.9%
sad	0.0%	9.1%	1.7%	**86.7%**	0.8%	1.7%
angry	5.1%	0.0%	0.0%	0.0%	**86.1%**	8.8%
anxious	5.3%	0.0%	0.9%	0.9%	3.5%	**89.4%**

Table 7. 2-stage hierarchical classification

Important is the fact that all three subclassifications in Figure 5 are trained separately by using different feature sets. Even the two feature sets used for emotion recognition with a high and a low activation level are different. In each case, the best 25 features were selected using SFFS. With this strategy and by using our new set of voice quality parameters, we achieved an overall classification rate of 83.5%. This is an improvement of another 8.0% compared to direct classification.

4.4.2 3-stage hierarchical classification

Motivated by the psychological emotion model, we found out that one can further improve the classification results by using only binary subclassifications (Lugger and Yang, 2008). That means, we perform 5 classifications in 3 stages for 6 emotions. Every frame in Figure 6

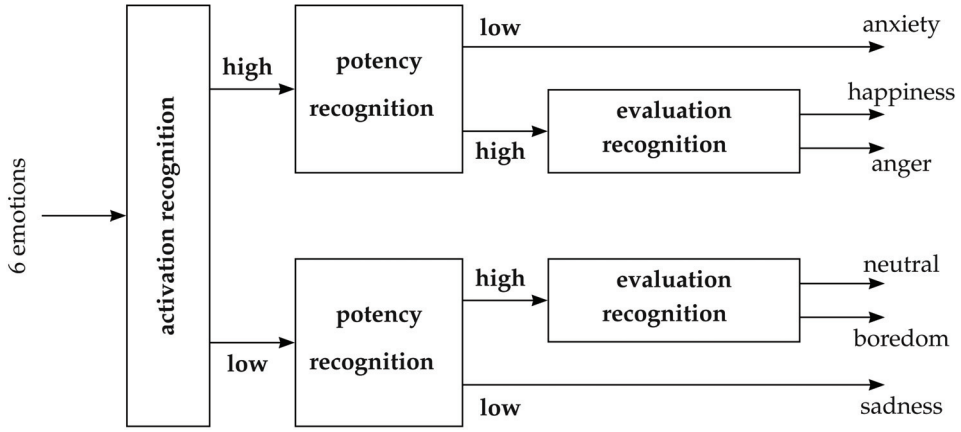


Fig. 6. 3-stage approach of emotion recognition

corresponds to one binary classification whose best 25 features are separately optimized by SFFS. In the first stage, we classify two different activation levels, in analogy to Figure 5. One class including anger, happiness, and anxiety has a high activation level while the second class including neutral, boredom, and sadness has a low activation level. For this activation discrimination, we achieve a very good classification rate of 98.8% on average. Table 8 shows the confusion matrix using 25 features, analog to Table 6.

activation	high	low
high	**99.1%**	0.9%
low	1.8%	**98.2%**

Table 8. Classification of 2 activation levels

In the second stage, we classify two potency levels within each activation class. That means, all patterns that were classified to high activation in the first stage are classified to one class containing happiness and anger or to a second class containing only anxiety. Similarly, all patterns that were classified to low activation in the first stage are classified to one class containing neutral and boredom or to a second class containing sadness. Table 9 shows the classification results for the second stage. Here, there is a noticeable advantage of including voice quality features.

	high activation		low activation	
potency	high	low (ax)	high	low (sa)
high	**98.8%**	1.2%	**97.6%**	2.4%
low	11.7%	**88.3%**	11.2%	**88.8%**

Table 9. Classification of 2 potency levels

In the third stage, we distinguish between the emotions that differ only in the evaluation dimension: happiness and anger as well as neutral and boredom. The confusion matrix for the third stage is shown in Table 10. Here, for the discrimination of anger and happiness, voice quality parameters bring a large improvement because of the rough voice quality for angry utterances. This 3-stage strategy using an optimized feature set of 25 features leads to the overall confusion matrix shown in Table 11. It corresponds to an overall recognition rate

	high activation		low activation	
evaluation	high (hp)	low (ag)	high (nt)	low (bd)
high	**84.6 %**	15.4%	**91.8%**	8.2%
low	5.9%	**94.1 %**	2.8%	**97.2%**

Table 10. Classification of 2 evaluation levels

emotion	happy	bored	neutral	sad	angry	anxious
happy	**82.2%**	0.0%	0.0%	0.0%	15.0%	2.8%
bored	0.0 %	**94.6%**	2.7%	2.7%	0.0%	0.0%
neutral	1.9%	7.8%	**87.4%**	1.9%	0.0%	1.0%
sad	0.0%	8.3%	2.5%	**86.7%**	0.0%	2.5%
angry	5.9%	0.0%	0.0%	0.0%	**94.1%**	0.0%
anxious	5.3%	0.9%	0.0%	0.9%	6.2%	**86.7%**

Table 11. 3-stage hierarchical classification

of 88.8%. This is an additional improvement of 5.3% in comparison to the 2-stage approach. In particular, the recognition rate of happiness is improved by nearly 15% due to its separately selected feature set containing the so important voice quality information.

5. Conclusion and outlook

5.1 Conclusion

In this study, we presented a novel approach of speaker independent emotion classification. We used a large set of voice quality parameters in addition to standard prosodic features. Altogether we extracted 346 acoustic features from the speech utterances. In all classification studies, we used the SFFS algorithm to reduce the feature number to 25. In a first study, we could show that our voice quality parameters outperform the well known mel frequency cepstral coefficients in the application of speaker independent emotion recognition. Thus, a combined feature set of prosodic and voice quality features led to the best recognition result using an 1-stage classification. Using MFCC and VQP in addition to prosodic features brought no further improvement in classification performance. We further compared a flat 1-stage classification of 6 emotions with a 2-stage respectively 3-stage hierarchical classification approach using only prosodic and a combined feature set. A summary of all the results using the best 25 features is shown in Table 12.

method / features	prosodic features	prosodic + VQP features
1-stage	**72,8%**	75,5%
2-stage	79,3%	83,5%
3-stage	83,2%	**88,8%**

Table 12. Overview of classification strategies

We observed that in general a multi-stage classification performs better than a flat classifier. For the classification of 6 emotions, the best recognition rate could be achieved by using the 3-stage classification consisting of 5 binary subclassifications. This is true for both only prosodic features and the whole feature set. The overall classification rate is raised by 10.4% respectively 13.3% by using the psychological motivated hierarchical classification in comparison to a flat classification. We also showed that parameters of voice quality supply a contribution in addition to the well known prosodic features. They deliver information concerning the different phonation types used by the emotion production of the speaker. This information is not adequately contained in the prosodic features. In the flat classification we could improve the classification rate by 2.7%. Another interesting observation is that the gain of using additional VQP is even higher when using a multi-stage classification. For the 2-stage respectively 3-stage approach, using voice quality features result in a gain of 4.2% respectively 5.6%. With our best method, we achieved an improvement of 16.0% in comparison to a standard flat classification using only prosodic features. This improvement could be even larger by using other emotional databases that make more use of different voice qualities in the production of emotions.

5.2 Outlook

Although all the presented classifications are speaker independent, the results are strongly optimized for the 10 speakers contained in the database. By using the speaker independent classification rate as criterion for the selection algorithm, the features are selected in a way

that is optimizing the classification rate of the unknown speaker. So we can say the classification itself is speaker independent but not the feature selection process. Because of the relatively low number of speakers, the dependency of the results on the speakers is high. That is why an additional study on the robustness of the here presented results is necessary. In this study, the classification data should neither be included in the training data nor in the feature selection process.

Another open question is: How do other multi-stage classification approaches perform? In the pattern recognition literature there exist other multi-stage classification methods as cascaded or parallel classification approaches. Do they significantly differ in the performance? And how about the robustness of these different approaches? Which is the most robust one for speaker independent emotion recognition?

This study is based on a well known German database. We have to mention that the utterances are produced by actors. So the speakers only performed this emotional state in an acoustic manner. They have not necessarily felt this emotion at the moment when they produced the spoken utterance. It would be interesting to test the proposed methods with a more natural database. But larger emotion databases with conversational speech are really rare.

6. References

F. Burkhardt, A. Paeschke, M. Rolfes, and W.F. Sendlmeier. A database of German emotional speech. *Proceedings of Interspeech*, 2005.

Nick Campbell and Parham Mokhtari. Voice quality: the 4th prosodic dimension. *15th International Congress of Phonetic Sciences, 2003*, 2003.

R. Cowie and E. Douglas-Cowie. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, 18(1):32–80, 2001.

P. Ekman. Universal and cultural differences in facial expressions of emotion. *Nebraska Symposium on Motivation*, 19:207–283, 1972.

G. Fant. *Acoustic theory of speech production*. The Hague: Mouton, 1960.

Christer Gobl and Ailbhe Ni Chasasaide. The role of voice quality in communicating emotion, mood and attitude. *Speech communication*, 40:189–212, 2003.

Gudrun Klasmeyer and Walter F. Sendlmeier. *Voice quality measurement*, chapter Voice and emotional states, pages 339–357. Singular Publishing group, 2000.

Chul Min Lee and S. Narayanan. Toward detecting emotions in spoken dialogs. *Transaction on speech and audio processing*, 13(2):293–303, 2005.

Michael Lewis, Jeannette Haviland-Jones, and Lisa Feldman Barrett, editors. *Handbook of emotions*. The Guilford Press, 2008.

Marko Lugger and Bin Yang. Classification of different speaking groups by means of voice quality parameters. *ITG-Sprach-Kommunikation*, 2006.

Marko Lugger and Bin Yang. An incremental analysis of different feature groups in speaker independent emotion recognition. *ICPhS, Saarbrücken*, 2007a.

Marko Lugger and Bin Yang. The relevance of voice quality features in speaker independent emotion recognition. *ICASSP, Hawaii, USA*, 2007b.

Marko Lugger and Bin Yang. Cascaded emotion classification via psychological emotion dimensions using a large set of voice quality parameters. In *IEEE ICASSP, Las Vegas*, 2008.

Marko Lugger, Bin Yang, and Wolfgang Wokurek. Robust estimation of voice quality parameters under real world disturbances. *IEEE ICASSP*, 2006.

S. McGilloway, R. Cowie, S. Gielen, M. Westerdijk, and S. Stroeve. Approaching automatic recognition of emotion from voice: A rough benchmark. *ISCAWorkshop Speech and Emotion*, pages 737–740, 2000.

A. Nogueiras, A. Morena, A. Bonafonte, and JB. Marino. Speech emotion recognition using hidden Markov models. *Eurospeech*, pages 2679–2682, 2001.

T. Nwe, S. Foo, and L. De Silva. Speech emotion recognition using hidden Markov models. *Speech communication*, 41:603–623, 2003.

G.C. Orsak and D.M. Etter. Collaborative SP education using the internet and matlab. *IEEE Signal processing magazine*, 12(6):23–32, 1995.

P. Pudil, F. Ferri, Novovicova J., and J. Kittler. Floating search method for feature selection with nonmonotonic criterion functions. *Pattern Recognition*, 2:279–283, 1994.

H. Schlosberg. Three dimensions of emotions. *Psychological Review*, 61:81–88, 1954.

B. Schuller, D. Arsic, F. Wallhoff, and G. Rigoll. Emotion recognition in the noise applying large acoustic feature sets. *Speech Prosody, Dresden*, 2006.

K. Stevens and H. Hanson. Classification of glottal vibration from acoustic measurements. *Vocal Fold Physiology*, pages 147–170, 1994.

D. Talkin, W. Kleijn, and K. Paliwal. A robust algorithm for pitch tracking (RAPT). *Speech Coding and Synthesis, Elsevier*, pages 495–518, 1995.

David Talkin. Speech formant trajectory estimation using dynamic programming with modulated transition costs. *Technical Report, Bell Labs.*, 1987.

B. Tischer. Die vokale Kommunikation von Gefühlen, Fortschritte der psychologischen Forschung. Psychologie-Verlag-Union, 1993.

D. Ververidis and C. Kotropoulos. A state of the art review on emotional speech databases. *1st Richmedia conference*, pages 109–119, 2003.

Irena Yanushevskaya, Christer Gobl, and Ailbhe Ni Chasaide. Voice quality and loudness in affect perception. *Speech prosody, Campinas*, 2008.

23

A Weighted Discrete KNN Method for Mandarin Speech and Emotion Recognition

Tsang-Long Pao, Wen-Yuan Liao and Yu-Te Chen
Department of Computer Science and Engineering, Tatung University
Taiwan, R.O.C.

1. Introduction

Speech signal is a rich source of information and convey more than spoken words, and can be divided into two main groups: linguistic and nonlinguistic. The linguistic aspects of speech include the properties of the speech signal and word sequence and deal with what is being said. The nonlinguistic properties of speech have more to do with talker attributes such as age, gender, dialect, and emotion and deal with how it is said. Cues to nonlinguistic properties can also be provided in non-speech vocalizations, such as laught or cry.
The main investigated linguistic and nonlinguistic attributes in this article were those of audio-visual speech and emotion speech. In a conversation, the true meaning of the communication is transmitted not only by the linguistic content but also by how something is said, how words are emphasized and by the speaker's emotion and attitude toward what is said. The perception of emotion in the vocal expressions of others is vital for an accurate understanding of emotional messages (Banse & Scherer, 1996). In the following, we will introduce the audio-visual speech recognition and speech emotion recognition, which are the applications of our proposed weighted discrete K-nearest-neighbor (WD-KNN) method for linguistic and nonlinguistic speech, respectively.
The speech recognition consists of two main steps, the feature extraction and the recognition. In this chapter, we will introduce the methods for feature extraction in the recognition system. In the post-processing, the different classifiers and weighting schemes on KNN-based recognitions are discussed for the speech recognition. The overall structure of the proposed system for audio-visual and speech emotion recognition is depicted in Fig. 1. In the following, we will briefly introduce the previous researches on audio-visual and speech emotion recognition.

1.1 Audio-visual speech recognition

For past decades, automatic speech recognition (ASR) by machine has been an attractive research topic. However, in spite of extensive research, the performance of current ASR is far from the performance achieved by humans, especially in noisy condition. Most previous ASR systems make use of the acoustic speech signal only and ignore the visual speech cues. They all ignore the auditory-visual nature of speech.
Although acoustic-only-based ASR systems yield excellent results in the laboratory experiment, the error of the recognition can increase in the real world. Noise robust methods

have been proposed. To overcome this limitation, audio speech-reading system, through the use of visual information into audio information, has been considered (Faraj & Bigun, 2007; Farrell et al, 1994; Kaynak et al, 2004). In addition, there has been growing interested in introducing new modalities into the ASR and human–computer interface. With this motivation, enormous research on multi-model ASR has been carried out.

In recent years, there has been many automatic speech-reading systems proposed, that combine audio and visual speech features. For all such systems, the objective of these audio-visual speech recognizers is to improve recognition accuracy, particularly in difficult condition. They most concentrated on the two problems of visual feature extraction and audio-visual fusion. Thus, the audio-visual speech recognition is a work combining the disciplines of image processing, visual-speech recognition and multi-modal data integration. Recent reviews can be found in Chen (Chen & Rao, 1997; Chen, 2001), Mason (Chibelushi et al., 2002), Luettin (Dupont & Luettin, 2000) and Goldschen (Goldschen, 1993).

As above described, most ASR work on detecting speech states investigated speech data which were recorded in quiet environment. But humans are able to perceive emotions even in noisy background (Chen, 2001). In this article we will compare several classifiers for detecting speech from clean and noisy Mandarin speech.

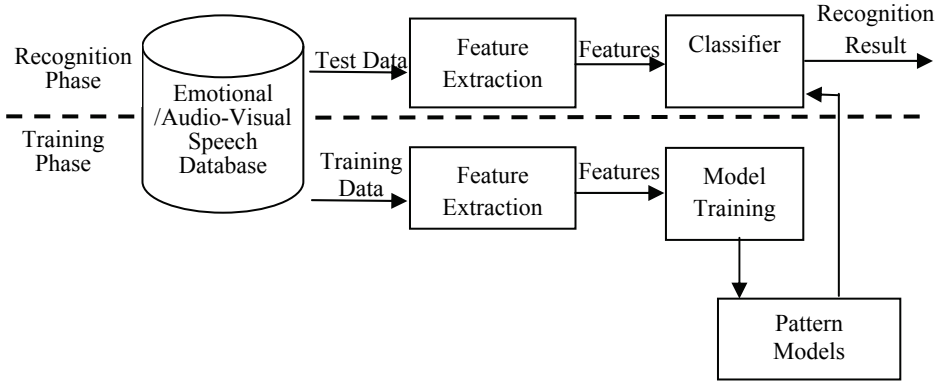


Fig. 1. Overall speech recognition system consisting of the speech extraction and recognition.

1.2 Speech emotion recognition

Besides being described as the audio-visual speech recognition, recognizing emotions from speech has gained increased attention in recent years. There are wide-ranging applications in real word (Huang & Ma, 2006), including health, public safety, education, slogan validation and call center. Taking advantage of the emotional information in speech allows more effective processing of the language information and yields a much more natural human-computer interaction.

Research on understanding and modeling human emotions, a topic that has been predominantly dealt within the fields of psychology and linguistics, is attracting increasing attention within the engineering community. A major motivation comes from the need to improve both the naturalness and efficiency of spoken language human-machine interfaces. Researching emotions, however, is extremely challenging for several reasons. One of the

main difficulties results from the fact that it is difficult to define what emotion means in a precise way. Various explanations of emotions given by scholars are summarized in (Kleinginna & Kleinginna, 1981). Research on the cognitive component focuses on understanding the environmental and attended situations that give rise to emotions; research on the physical components emphasizes the physiological response that co-occurs with an emotion or rapidly follows it. In short, emotions can be considered as communication with oneself and others (Kleinginna & Kleinginna, 1981).

Traditionally, emotions are classified into two main categories: primary (basic) and secondary (derived) emotions (Murray & Arnott, 1993). Primary or basic emotions generally can be experienced by all social mammals (e.g., humans, monkeys, dogs and whales) and have particular manifestations associated with them (e.g., vocal/facial expressions, behavioral tendencies and physiological patterns). Secondary or derived emotions are combinations of or derivations from primary emotions.

Emotional dimensionality is a simplified description of the basic properties of emotional states. According to the theory developed by Osgood, Suci and Tannenbaum (Osgood et al, 1957) and in subsequent psychological research (Mehrabian & Russel, 1974), the computing of emotions is conceptualized as three major dimensions of connotative meaning: arousal, valence and power. In general, the arousal and valence dimensions can be used to distinguish most basic emotions. The locations of emotions in the arousal-valence space are shown in Fig. 2, which provides a representation that is both simple and capable of conforming to a wide range of emotional applications.

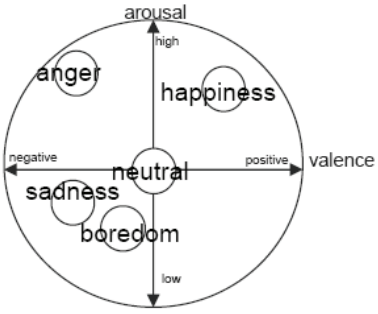


Fig. 2. Graphic representation of the arousal-valence dimension of emotions (Osgood et al. 1957)

1.3 Speech recognition methods

There are various techniques for classification such as K-nearest-neighborhood (KNN), weighted KNN (WKNN) (Dudani, 1976), KNN classification using Categorical Average Patterns (WCAP) (Takigawa *et al*, 2005), Gaussian mixture model (GMM) (Neiberg *et al*, 2006; Reynolds *et al*, 2001; Reynolds & Rose, 1995), Hidden Markov Model (HMM) (Brand et al,1997; Chibelushi *et al*, 2002), Neural Network (NN) and support vector machine (SVM). In our system, the classification was performed using our proposed weighted discrete K-nearest-neighbor (WD-KNN) classifier (Pao *et al*, 2007).

In this chapter we focus on recognition from speech signals and moreover on comparison of different weighting functions applied in various weighted discrete KNN-based classifiers.

The performance is verified by experiments with a Mandarin speech corpus. The baseline performance measure is based on the traditional KNN classifier.
This chapter is organized as follows. In section 2, we introduce the used classifiers and previous researches. In section 3, the feature selection policy and extraction methods for speech and emotion are described. In section 4, an speech emotion recognition system is reviewed and three common weighting functions and the used Fibonacci function are described. Experimental results are given in section 5. In section 6, some conclusions are outlined.

2. Classifiers

The problem of detecting the speech and emotion can be formulated as assignment a decision category to each utterance. Two main types of information can be used to identify the speaker's speech: the semantic content of the utterance and the acoustic features such as variance of the pitch. In the following, we will review various classification and other related literatures.

2.1 KNN

K-nearest neighbor (KNN) classification is a very simple, yet powerful classification method. The key idea behind KNN classification is that similar observations belong to similar classes. Thus, one simply has to look for the class designators of a certain number of the nearest neighbors and sum up their class numbers to assign a class number to the unknown.
In practice, given an instance y, KNN finds the k neighbors nearest to the unlabeled data from the training space based on the selected distance measure. The Euclidean distance is commonly used. Now let the k neighbors nearest to y be $N_k(\mathbf{y})$ and c(z) be the class label of z. The cardinality of $N_k(\mathbf{y})$ is equal to k and the number of classes is l. Then the subset of nearest neighbors within class $j \in \{1, \ldots, l\}$ is

$$N_k^j(\mathbf{y}) = \{\mathbf{z} \in N_k(\mathbf{y}) : c(\mathbf{z}) = j\} \tag{1}$$

The classification result $j^* \in \{1,\ldots,l\}$ is defined as the majority vote:

$$j^* = \arg\max_j \left| N_k^j(\mathbf{y}) \right| \tag{2}$$

2.2 WKNN

Weighted KNN was proposed by Dudani (Dudani, 1976). In WKNN, the k nearest neighbors are assigned different weights. Let w_i be the weight of the ith nearest samples and $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k$ be the k nearest neighbors of test sample $\boldsymbol{y}$ arranging in increasing distance order. So $\boldsymbol{x}_1$ is the first nearest neighbor of $\boldsymbol{y}$. The classification result $j^* \in \{1,\ldots,l\}$ is assigned to the class for which the weights of the representatives among k nearest neighbors sum to the largest value.

$$j^* = \arg\max_j \sum_{p=1}^{k} \begin{cases} w_p, & \text{if } c(\mathbf{x}_p) = j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

2.3 WCAP

The WCAP classification method was proposed by Takigawa for improving performance on handwritten digits recognition (Takigawa et al, 2005). Let w_i^j be the weight of the ith nearest samples of class j. After the k nearest samples of a test sample $\boldsymbol{y}$, denoted as $\boldsymbol{x}_i^j$, i=1,…,k, are extracted from class j by using Euclidean distance measure d_i^j. Then, the weight is calculated and normalized by equations which will be described in next section. Finally, the class of a test sample is determined by the following classification rule:

$$j^* = \arg\min_j \left\{ \left\| \sum_{p=1}^{k} w_p^j \mathbf{x}_p^j - \mathbf{y} \right\|^2 \right\} \tag{4}$$

2.4 HMM

A hidden Markov model (HMM) is a statistical model for sequences of feature vectors that are representative of the input signal (Robert & Granat, 2003; Yamamoto *et al*, 1998). The observed data is assumed to have been generated by an unobservable statistical process of a particular form. This process is such that each observation is coincident with the system being in a particular state. Furthermore it is a first order Markov process: the next state is dependent only on the current state. The model is completely described by the initial state probabilities, the first order Markov chain state-to-state transition probabilities, and the probability distributions of observable outputs associated with each state. HMM has a long history in speech recognition. It has the important advantage that the temporal dynamics of speech features can be caught due to the presence of the state transition matrix. From the experimental results of (Kwon *et al*, 2003), HMM classifiers yielded classification accuracy significantly better than the linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA).

2.5 GMM

Gaussian Mixture Models (GMMs) provide a good approximation of the originally observed feature probability density functions by a mixture of weighted Gaussians. The mixture coefficients were computed by use of an Expectation Maximization algorithm. Each emotion is modeled in one GMM. The decision is made for the maximum likelihood model. From the results of (Reynolds *et al*, 2001; Reynolds & Rose, 1995), the authors concluded that using GMMs on the frame level is a feasible technique for speech classification and the results of two models VQ and GMM are not worse than the performance of the HMM.

2.6 WDKNN

We proposed the WD-KNN method for classifying speech and emotion in previous research (Pao *et al*, 2007). Before presenting the proposed method, we describe unweighted-distance KNN classifier as it is the foundation of the method. Without loss of generality, the collected speech samples are split into data elements $\boldsymbol{x}_1,\ldots, \boldsymbol{x}_t$, where t is the total number of training samples. The space of all possible data elements is defined as the input space $\boldsymbol{X}$. The elements of the input space are mapped into points in a feature space $\boldsymbol{F}$. In our work, a feature space is a real vector space of dimension d, $\Re^d$. Accordingly, each point $\boldsymbol{f}_i$ in $\boldsymbol{F}$ is

represented by a *d*-dimensional feature vector. A feature map is defined to be a function that takes an input element in the input space and maps it to a point in the feature space. We use ϕ to define a feature map

$$\phi : \mathrm{X} \rightarrow \mathrm{F} \tag{5}$$

Let $\boldsymbol{x}_i^j$, $i = 1,...,n_j$, be the *i*-th training sample of class *j*, where n_j is the number of samples belonging to a class *j*, $j \in \{1,\ldots,l\}$ and *l* is the number of classes. The total number of training samples is

$$t = \sum_{j=1}^{l} n_j \tag{6}$$

When a test sample $\boldsymbol{y}$ and Euclidean distance measure are given, we obtain the *k* nearest neighbors belonging to class *j*, $N_{k,l}^j(\boldsymbol{y})$, which can be defined as

$$N_{k,l}^j(\mathbf{y}) = \left\{\mathbf{z} \in N_{k,l}(\mathbf{y}) : c(\mathbf{z}) = j\right\} \tag{7}$$

where the cardinality of the set $N_{k,l}^j(\boldsymbol{y})$ is equal to *k*. Finally the class label of the test sample in unweighted-discrete MKNN classifier is determined by

$$j^* = \arg\min_{j=1,\ldots,l} \sum_{i=1}^{k} dist_i^j \tag{8}$$

where $dist_i^j$ is the Euclidean distance between the *i*th nearest neighbor in $N_{k,l}^j(\mathbf{y})$ and the test sample $\boldsymbol{y}$.

Next, we will describe and formulate the weighted-distance KNN classifier (Dudani, 1976; Pao *et al*, 2007) as follows. Among the *k* nearest neighbors in class *j*, the following relationship is established:

$$dist_1^j \le dist_2^j \le \ldots \le dist_k^j \tag{9}$$

Let w_i be the weight of the *i*th nearest samples. From above, we know that the one having the smallest distance value $dist_1^j$ is the most important. Consequently, we set the constraint $w_1 \ge w_2 \ge \ldots \ge w_k$. Then, the classification result $j^* \in \{1, ..., l\}$ is defined as

$$j^* = \arg\min_{j=1,\ldots,l} \sum_{i=1}^{k} w_i dist_i^j \ . \tag{10}$$

In our proposed system, the selection of weights used in the WD-KNN is an important factor for the recognition rate. After extensive investigations and calculations, we found that the Fibonacci sequence weighting function yields the best result in the WD-KNN classifiers. The Fibonacci weighting function is defined as follows

$$w_i = w_{i+1} + w_{i+2}, \quad w_k = w_{k-1} = 1 \tag{11}$$

The definition is in the reverse order of the ordinary Fabonacci sequence. Why Fibonacci weighting function is used? The Fibonacci weighting function indicates that each weight is the sum of the two latter ones. This implies that when the weighted value of 1st nearest neighbor equals to the sum of the weighted values of the latter two neighbors nearest to test sample, the later two added up has the same importance as the first one. The second reason is that we compared different weighting schemes, including Fibonacci weighting, linear distance weighting, inverse distance weighting, and rank weighting, in KNN based classifiers to recognize speech and emotion sates in Mandarin speech (Pao *et al*, 2007). The experimental results show that the Fibonacci weighting function performs better than others.

3. Features extraction

3.1 Acoustic features for emotion recognition

For speech recognition system, a critical step is the extraction and selection of the feature set. Various features relating to pitch, energy, duration, tunes, spectral, and intensity, etc. have been studied in speech recognition and emotion recognition (Murray & Arnott, 1993; Kwon et al, 2003). Due to the redundant information, the forward feature selection (FFS) and the backward feature selection (BFS) are carried out to extract the most representative feature set based on KNN classifier among energy, pitch, formant (F1, F2 and F3), linear predictive coefficients (LPC), Mel-frequency cesptral coefficients (MFCC), first derivative of MFCC (dMFCC), second derivative of MFCC (ddMFCC), Log frequency power coefficients (LFPC), perceptual linear prediction (PLP).

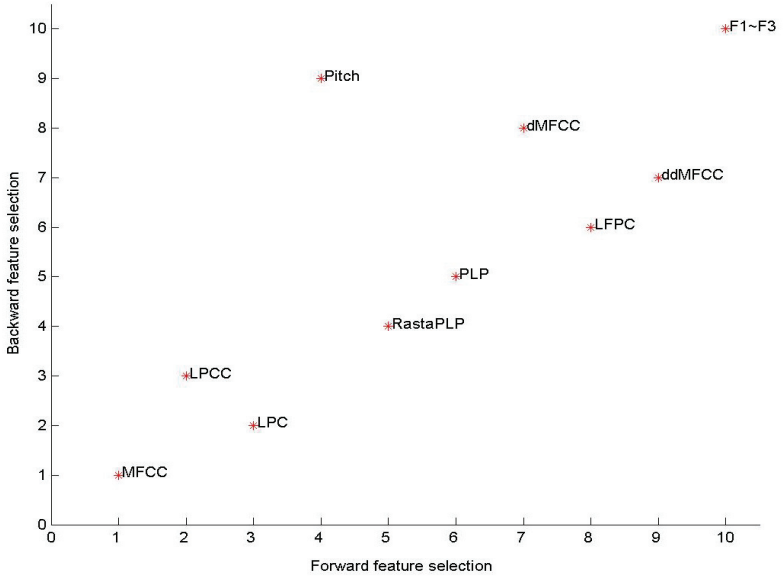


Fig. 3. Features ranking in speech emotion recognition using FFS and BFS by KNN classifier.

For each of these series, mean values are determined to build up the fixed-length feature vector. Besides, emotion was expressed mainly at the utterance level. It is crucial to normalize the feature vector. In this article, we use max-min normalization, to normalize feature vector to the range of [0, 1]. Fig. 3 shows the ranking of these features by KNN classifier. Features near origin are considered to be more important. Finally, we combine MFCC, LPCC and LPC as the best feature set used in the emotion recognition system. The zero-order coefficients of MFCC and LPCC are included as they provide energy information. When we obtain the features from the training and the test data, we can calculate the distance between them to classify the test data.

3.2 Acoustic and visual features for speech recognition

In the previous section, we introduce several features used in speech emotion recognition. For speech recognition, there have been many automatic speech-reading systems proposed recently that combine audio and visual speech features (Poamianos et al, 2003; Zeng et al, 2005; Tang & Li, 2001). For all such systems, the objective of these audio-visual speech recognizers is to improve recognition accuracy, particularly in difficult condition.

Generally speaking, the features for visual speech information extraction from image sequences can be grouped into the following classes: geometric features (Kaynak et al, 2004; Petajan, 1984; Petajan, 1988), model based features (Dupont & Luettin, 2000; Hazen, 2006; Poamianos et al, 2003; Aleksic et al, 2002), visual motion features (DeCarlo & Metaxas, 2000; Faraj & Bigun, 2006), and image based features(Matthews et al, 2002; Zhang et al, 2002). In our system, the visual motion feature is used.

The motion-based feature approach assumes that visual motion during speech production contains relevant speech information. Visual motion information is likely to be robust to different skin reflectance and speakers. Optical flow (DeCarlo & Metaxas, 2000) algorithms are usually used in the estimation of feature point movement. The algorithms usually do not calculate the actual flow field but visual flow field. A further problem consists of the extraction of related features from the flow field. However, recent research about motion-based segmentation got more performance than previous experiments. So the visual motion analysis can improve the performance of recognition.

To estimate the movement of feature point in the image, the estimation method similar to the motion compensation technique used in the video compression is used. The image is partitioned into a set of non-overlapped fixed sized small rectangular blocks. The translation motion within each block is assumed to be uniform. This model only considers translation motion. Other types of motion, such as rotation and zooming, may be approximated by the piecewise translation of these small blocks.

In the audio-visual speech recognition system, the region of interest is first extracted from the original image. The main features, corners or edges of the mouth, are then found as the cues for motion estimation. The motion vectors computation corresponding to the feature points are performed by the block matching based motion estimation algorithm. Finally, the motion vectors for selected feature points are carried out for the feature vectors as the input of the recognition. Fig. 4 shows the example of segmented mouth images and selected feature points, marked by the white dots, and their corresponding motion vectors.

Since speech basically is a non-stationary random process, the characteristics of the acoustic features for audio-visual speech recognition is stochastic and its statistics are time-varying. According to the study of speech production, differences of human speech are generated by

the variation of the mouth and vocal tract. In the audio-visual recognition, we extract the MFCC features, including basic and derived features, for the audio-visual speech recognition.

Fig. 4. Examples of segmented mouth image series for pronunciation "Yi" of Mandarin word "1" and selected feature points in the audio-visual speech recognition system

4. Speech recognition system

4.1 The emotional and audio-visual speech corpus

For the initial stage, an emotional corpus needs to be built up in order to form a base for eliciting emotions from speech signals. In this chapter, five primary emotions, anger, happiness, sadness, boredom, and neutral, are investigated. The emotion corpus database was recorded by 18 males and 16 females who portray 20 prompting Mandarin sentences with the above mentioned emotional states. These sentences are one to six words and are purposely neutral and meaningful so the participants can easily express them with these emotions. Human subjective judgment was conducted to filter out ambiguous emotional utterances for further recognition analysis. Corpora with 80% or higher agreement was kept.
As for the audio-visual database, there are some databases exist for the audio-visual research area. But almost all of them are in English or other language, such as Tulips1, AVLetters, M2VTS (Messer et al, 1999), CUAVE (Patterson et al, 2002), etc. The Mandarin database is rare in comparison with other languages. In our experiment, we recorded and created an audio-visual database of Mandarin speech, including Mandarin digits 0 ~ 9.
Our audio-visual database consists of two major parts, one in English and one in Mandarin. The video in English was recorded from 35 speakers while the video in Mandarin was recorded from 40 speakers. The importance of the database is to allow the comparison of recognition of English speech and Mandarin speech. The video is in color with no visual aids given for lip or facial feature extraction. In both parts of database, each individual

speaker was asked to speak 40 isolated English and Mandarin digits, respectively, facing a DV camera.

The video was recorded at a resolution of 320×240 with the NTSC standard of 29.97 fps, using a 1-mega-pixel DV camera. The on-camera microphone was used to record the speeches. Lighting was controlled and a blue background was used to allow change of different backgrounds for further applications. In order to split the video into the visual part and the audio part, we developed a system to decompose the video format (*.avi) into the visual image files (*.bmp) and speech wave files (*.wav) automatically.

4.2 Weighting schemes

As described in Section 1, speech perception by human is a bimodal process characterized by high recognition accuracy and attractive performance degradation in the presence of distortion. There are many classifiers available for decision making such as support vector machines, Bayesian networks, decision trees, artificial neural networks, and fuzzy neural networks, HMM, GMM and KNN-based classifiers. This section focuses on weighting schemes for KNN-based classifiers, which include traditional KNN, WKNN, and WD-KNN. The details about these classifiers are presented in Section 2. The key idea of WKNN, and WD-KNN classifiers is to assign more weights to closer samples by a weighting function to improve the recognition rate. The common weighting functions are as follow. Dudani has proposed a simple linear distance weighting function (Dudani, 1976)

$$w_i = \begin{cases} 1, & \text{if } d_k = d_1 \\ \dfrac{d_k - d_i}{d_k - d_1}, & \text{if } d_k \neq d_1 \end{cases}, \tag{12}$$

where d_i is the distance to the test sample of the ith nearest neighbor, and d_1 and d_k indicate the distance of the nearest neighbor and the farthest neighbor respectively. Dudani has further proposed an inverse distance weighting function

$$w_i = \frac{1}{d_i} \quad if \quad d_i \neq 0 \tag{13}$$

and a rank weighting function

$$w_i = k - i + 1 . \tag{14}$$

From the experimental results done by Dudani, it is well known that weighted version of KNN can improve error rates by using above weighting functions. In this chapter, we propose to use Fibonacci sequence as the weighting function in these classifiers. The Fibonacci weighting function is defined in Eq. (11).

5. Experimental results

5.1 Experimental results for different weighting schemes

First, the value of k in KNN, WCAP and WD-KNN classifiers must be determined. In our previous experiments (Chang, 2005), the distribution of recognition accuracy from clean speech on different k indicates that k sets to 10 can make an acceptable performance with

relatively simple computation in KNN. Therefore, the value of k in KNN, WCAP and WD-KNN classifiers is set to 10.

Table 1 summarizes the experimental results of different weighting functions in speech emotion recognition using various classifiers. The accuracy ranges from 73.8~76.1%, 73.1%~74.5%, and 78.7%~81.4% in WKN, WCAP, and WD-KNN, respectively. One important finding is that Fibonacci weighting function outperform others in all classifiers. Compared to the baseline attained from KNN method, the largest accuracy improvement of 4.9%, 2.8% and 12.3% can be achieved in these classifiers. The highest recognition rate is 81.4% with WD-KNN classifier weighted by Fibonacci sequence.

		Weighting functions			
		Linear distance	Inverse distance	Rank	Fibonacci
Classifiers	WKNN	75.6%	74.2%	73.8%	76.1%
	WCAP	74.2%	73.6%	73.1%	74.5%
	WD-KNN	78.7%	79.5%	81.2%	81.4%

Table 1. Experimental results of using different weighting functions in speech emotion recognition

In the audio-visual speech recognition, we use our Mandarin speech database as the input data. The database used here contains the Mandarin digits 0 to 9 by 40 speakers. There are a total of 1600 utterances. In the training phase, the 400 utterances of the database containing Mandarin digits 0-9 from all speakers are used as the training set. After we train the model, the other 1200 utterances are used as the testing set in testing phase.

The video stream is a sequence of 17 to 25 images with resolution of 200×120 pixel from the database. Before we compute the visual parameter, some image processing techniques are applied to image in order to make the computation convenient and increase the precision of the visual parameter. In our system, all of image sequences for Mandarin utterance was used in GMM and HMM recognition. In KNN and WD-KNN classifiers, since the distance between the feature vectors is computed, the size of each feature vector must be the same. The images of each utterance used for recognition is selected for a fixed number of images as the fixed-length feature vectors.

Table 2 summarizes the experimental results of different weighting functions in audio-visual speech recognition using various classifiers. The accuracy ranges from 72.8%~79.2%, and 84.6%~98.0% in WKNN and WD-KNN, respectively. The important finding is that

		Weighting functions			
		Linear distance	Inverse distance	Rank	Fibonacci
Classifiers	KNN	74.6%*			
	WKNN	76.4%	74.1%	72.8%	79.2%
	WD-KNN	84.6%	86.3%	88.5%	98.0%

*Weighting function is not used in KNN.

Table 2. Experimental results of using different weighting functions in audio-visual speech recognition

Fibonacci weighting function outperform others in all classifiers. Compared to the baseline attained from KNN method, the largest accuracy improvement of 6.4% and 13.4% can be achieved in these classifiers. The highest recognition rate is 98.0% with WD-KNN classifier weighted by Fibonacci sequence.

5.2 Experimental results using different classifiers on clean and noisy speech

Table 3 demonstrates the emotion recognition accuracy of clean speech and speech interfered by white Gaussian noise from the used classifiers. Accuracy in the table is the average recognition ratio of five emotions. From the results, the proposed WD-KNN is observed outstanding performance at all SNR among the three KNN-based classifiers. Compared with all other methods, the accuracy of WD-KNN is the highest on clean speech and noisy speech from 40dB to 20dB. GMM outperformed others on the 5dB noisy speech. The accuracy of HMM is decreased least among all classifiers from clean speech to 5dB noisy speech.

SNR	KNN	GMM	HMM	WCAP	WD-KNN
Clean	72.2%	70.3%	62.5%	74.5%	81.4%
40dB	65.7%	61.3%	50.2%	51.6%	71.5%
35dB	62.3%	59.7%	49.1%	55.4%	70.3%
30dB	60.8%	60.6%	51.6%	48.0%	67.3%
25dB	61.3%	60.0%	50.5%	44.4%	65.1%
20dB	51.8%	55.7%	39.1%	32.3%	57.1%
15dB	38.7%	47.1%	46.2%	43.6%	45.3%
10dB	32.7%	41.9%	39.6%	36.6%	37.4%
5dB	26.5%	38.7%	35.5%	30.1%	30.6%

Table 3. Comparison of the speech emotion recognition accuracy using five classifiers on clean and noisy speech

Table 4 summarizes the experimental results of audio-visual speech recognition for different classifiers on noise at various SNR values. The accuracy ranges from 78.7%~34.1%, 80.2%~34.3%, 75.1%~42.2%, and 81.8%~45.3% for GMM, HMM, KNN and WD-KNN, respectively. The results show that the WD-KNN with Fibonacci weighting function

SNR	***GMM***	***HMM***	***KNN***	***WD-KNN***
clean	92.5%	99.5%	95.8%	98.0%
30dB	78.7%	80.2%	75.1%	81.8%
25dB	70.4%	75.4%	76.3%	80.3%
20dB	68.5%	70.2%	70.4%	73.4%
15dB	55.2%	62.2%	63.7%	69.2%
10dB	49.2%	57.4%	57.5%	61.5%
5dB	46.5%	52.8%	54.2%	58.4%
0dB	34.1%	34.3%	42.2%	45.3%

Table 4. Comparison of the audio-visual speech recognition rate using different classifiers on clean and noisy speech

outperforms others in most of the cases. Compared to the baseline attained from KNN method, the recognition accuracy improvement of 2.2% to 5.5% at various SNR values can be achieved. In clean condition, the performance of the HMM recognizer seems better than the WD-KNN one. But in the noisy condition, the performance of WD-KNN classifier weighted by Fibonacci sequence is better than other classifiers.

6. Conclusions

In this chapter, we present a speech emotion recognition system to compare several classifiers on the clean speech and noisy speech. Our proposed WD-KNN classifier outperforms the other three KNN-based classifiers at every SNR level and achieves highest accuracy from clean speech to 20dB noisy speech when compared with all other classifiers. Similar to (Neiberg et al, 2006), GMM is a feasible technique for emotion classification on the frame level and the results of GMM are better than performances of the HMM. Although the performance of HMM is the lowest on clean speech, it is robust when the noise increase. The accuracy of KNN dropped rapidly when noise increases from 20dB to 15dB. WCAP performed the same from clean speech to 40dB noisy speech. The accuracy of 10dB noisy speech exceeds 20dB noisy speech in HMM and WCAP classifiers, which are unusual phenomena. In the future, more efforts will be made to investigate these strange results.

Automatic recognition of audio-visual speech aims at building classifiers for classifying audio-visual speech in test audio-visual speech. Until now, several classifiers were adopted independently. Among them, KNN is a very simple but elegant approach to classify various audio-visual speech. Later, some extensions of KNN, such as WKNN and WD-KNN, were proposed to improve the recognition rate.

Moreover, our focus is also to discuss weighting schemes used in different KNN-based classifier, including traditional KNN, weighted KNN and our proposed weighted discrete KNN. The key idea in these classifiers is to find a vector of real-valued weights that would optimize classification accuracy of the classification or recognition system by assigning lower weights to farther neighbors that provide less relevant information for classification and higher weights to closer neighbors that provide more reliable information. Several weighting functions were studied, such as linear distance weighting, inverse distance weighting and rank weighting. In this chapter, we propose to use the Fibonacci sequence as the weighting function. The overall results of the proposed classifier have proved that Fibonacci weighting function in three extended versions of KNN outperform others.

From the experimental results, we can observe that each classifier has their own advantages and disadvantages. How to combine these advantages of each classifier to achieve higher recognition rate requires further study. Besides, how to get an optimal weighting sequence is also deserved to be investigated.

7. References

P. S. Aleksic, J. J. Williams, Z. Wu, and A. K. Katsaggelos (2002), "Audio-Visual Speech Recognition Using MPEG-4 Compliant Visual Features", *EURASIP Journal on Applied Signal Processing*, No. 11, pp. 1213-1227, 2002

R. Banse & K. R. Scherer (1996), "Acoustic Profiles in Vocal Emotion Expression," *Journal of Personality and Social Psychology* 70, pp. 614-636. 1996.

M. Brand, N. Oliver & A. Pentland (1997), "Coupled hidden Markov models for complex action recognition," *Proc. IEEE CCVPR*, pp. 994–999, 1997.

Y. H. Chang (2005), "Emotion Recognition and Evaluation of Mandarin Speech Using Weighted D-KNN Classification", *Conference on Computational Linguistics and Speech Processing XVII* (ROCLING XVII), pp. 96-105, 2005.

T. Chen & R. Rao(1997), "Audiovisual interaction in multimedia communication," *ICASSP*, vol. 1. Munich, pp. 179-182, Apr. 1997.

T. Chen (2001), "Audio-visual speech processing," *IEEE Signal Processing Magazine*, Jan. 2001.

C. C. Chibelushi, F. Deravi & J. S. D. Mason (2002), "A review of speech-based bimodal recognition," *IEEE Trans. Multimedia*, vol. 4, pp. 23-37, Feb. 2002.

D. DeCarlo & D. Metaxas (2000), "Optical Flow Constraints on Deformable Models with Applications to Face Tracking," *Int'l J. Computer Vision*, vol. 38, no. 2, pp. 99-127, 2000.

S. Dupont & J. Luettin (2000), "Audio-visual speech modeling for continuous speech recognition," *IEEE Trans. Multimedia*, vol. 2, pp. 141–151, Sept. 2000.

S. A. Dudani (1976), "The Distance-Weighted K-Nearest-Neighbor Rule," *IEEE Trans Syst. Man Cyber* (1976) 325-327, 1976.

M.I. Faraj & J. Bigun (2006), "Person Verification by Lip-Motion," *Proc. Conf. Computer Vision and Pattern Recognition Workshop* (CVPRW '06), pp. 37-45, 2006.

M. I. Faraj & J. Bigun (2007), "Synergy of Lip-Motion and Acoustic Features in Biometric Speech and Speaker Recognition", *Computers, IEEE Transactions,* Vol. 56, No. 9 pp.1169 – 1175, Sept. 2007.

K. Farrell, R. Mammone & K. Assaleh (1994), "Speaker Recognition Using Neural Networks and Conventional Classifiers," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 1, pp. 194-205, 1994.

T. J. Hazen (2006), "Visual Model Structures and Synchrony Constraints for Audio-Visual Speech Recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 1082-1089, 2006.

R. Huang & C. Ma (2006), "Toward a Speaker-Independent Real-Time Affect Detection System", Vol. 1, *the 18th International Conference on Pattern Recognition*, pp. 1204-1207, 2006.

M. N. Kaynak, Q. Zhi, etc (2004), "Analysis of Lip Geometric Features for Audio-Visual Speech Recognition," *IEEE Transaction on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 34, pp. 564-570, July 2004.

P. R. Kleinginna & A.M. Kleinginna (1981), "A Categorized List of Emotion Definitions with Suggestions for a Consensual Definition," *Motivation and Emotion*, 5(4), pp.345-379, 1981.

O. W. Kwon, K. Chan, J. Hao & T. W. Lee (2003), "Emotion Recognition by Speech Signals", *Proceedings of EUROSPEECH*, pp. 125-128, 2003.

I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox & R. Harvey (2002), "Extraction of visual features for lipreading," *IEEE Trans. pattern analysis and machine intelligence*, vol. 24, pp. 198-213, 2002.

K. Messer, J. Matas, J. Kittler & J. Luettin (1999), "Xm2vtsdb: The Extended M2VTS Database," *Proc. Second Int'l Conf. Audio- and Video-Based Biometric Person Authentication* (AVBPA '99), pp. 72-77, 1999.

I. Murray & J. L.Arnott (1993), "Towards the Simulation of Emotion in Synthetic Speech: A Review of the Literature on Human Vocal Emotion," *Journal of the Acoustic So-ciety of America* 93(2), pp. 1097-1108, 1993.

D. Neiberg, K. Elenius & K. Laskowski (2006), "Emotion Recognition in Spontaneous Speech Using GMMs", *In Proc. of Interspeech 2006*. Pittsburg, pp.809-812, 2006.

C. E.Osgood, J.G. Suci & P. H. Tannenbaum (1957)., The Measurement of Meaning. The University of Illinois Press, Urbana, 1957.

T. L. Pao, Y. M. Cheng, Y. T. Chen & J. H. Yeh (2007), "Performance Evaluation of Different Weighting Schemes on KNN-Based Emotion Recognition in Mandarin Speech," *International Journal of Information Acquisition*, Vol. 4, No. 4, pp. 339-346, Dec. 2007.

E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy (2002), "Moving-Talker, Speaker-Independent Feature Study, and Baseline Results Using the CUAVE Multimodal Speech Corpus," *EURASIP Journal on Applied Signal Processing*, No. 11, pp. 1189-1201, 2002

E. D. Petajan, N. M. Brooke, B. J. Bischoff & D. A. Boddoff (1988), "Experiments in automatic visual speech recognition," in *Proc. 7th FASE Symp.*, Book 4, pp. 1163–1170, 1988.

E. D. Petajan (1984), "Automatic lipreading to enhance speech recognition," in *Proc. IEEE Global Telecommunications Conf.*, Atlanta, GA, pp. 265-272, Nov.1984.

G. Poamianos, etc (2003),"Recent Advances in the Automatic Recognition of Audiovisual Speech" *Proceeding of the IEEE*, Vol. 91, No. 9, September 2003.

D. Reynolds, T. Quatieri & R. B. Dunn (2001), "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, nos. 1-3, pp. 9-21, 2001.

D. Reynolds & R. Rose (1995), "Robust Text-Independent Speaker Identification Using Gaussian Mixture Models," *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 1, pp. 72-83, 1995.

E. Robert & A. Granat (2003), "A Method of Hidden Markov Model Optimization for Use with Geophysical Data Sets", *International Conference on Computational Science*, pp. 892-901, 2003.

Y. Takigawa, S. Hotta, S. Kiyasu & S. Miyahara, (2005), "Pattern Classification Using Weighted Average Patterns of Categorical k-Nearest Neighbors" *Proc. of the 1th International Workshop on Camera-Based Document Analysis and Recognition* 111-118, 2005

X. Tang & X. Li (2001), "Fusion of Audio-Visual Information Integrated Speech Processing," *Proc. Third Int'l Conf. Audio- and Video-Based Biometric Person Authentication* (AVBPA '01), pp. 127-143, 2001.

E. Yamamoto, S. Nakamura & K. Shikano (1998), "Lip Movement Synthesis from Speech Based on Hidden Markov Models," *J. Speech Comm.*, vol. 26, no. 1, pp. 105-115, 1998.

X. Zhang, C. C. Broun, R. M. Mersereau & M. A. Clements (2002), "Automatic Speechreading with Applications to Human-Computer Interface", *EURASIP Journal on Applied Signal Processing*, No. 11, pp. 1228-1247, 2002

Z. Zeng, Z. Zhang, B. Pianfetti, J. Tu & T. S. Huang (2005), "Audio-visual affect recognition in activation-evaluation space", *Proc. IEEE International Conference on Multimedia and Expo*, pp. 828–831, July 2005.

Applications

24

Motion-Tracking and Speech Recognition for Hands-Free Mouse-Pointer Manipulation

Frank Loewenich and Frederic Maire
Queensland University of Technology
Australia

1. Introduction

The design of traditional interfaces relies on the use of mouse and keyboard. For people with certain disabilities, however, using these devices presents a real problem.
In this chapter we describe a graphical user interface navigation utility, similar in functionality to the traditional mouse pointing device. Movement of the pointer is achieved by tracking the motion of the head, while button-actions can be initiated by issuing a voice command. Foremost in our mind was the goal to make our system easy to use and affordable, and provide users with disabilities with a tool that promotes their independence and social interaction.
The chapter is structured as follows. Section 2 provides an overview of related work. Section 3 describes our proposed system architecture, the face detection and feature tracking algorithms, as well as the speech recognition component. Section 4 contains experimental results and Section 5 discusses future work. We conclude the chapter in Section 6.

2. Research motivation and related work

Persons with disabilities are often unable to use computers. This is because they are either unable to find a suitable means of interaction or they simply cannot afford commercial solutions. We also found that available solutions do not promote the individual's sense of independence, as they require a third party to attach markers at various points of their anatomy. Our work addresses these shortcomings by providing a non-intrusive, reliable, inexpensive and robust visual tracking system. It allows persons, who may have disabilities ranging from not being able to use their hands to severe cases where the person is only able to move their head, to navigate and manipulate the graphical user interface using head movements and speech.
Research into assistive technologies is ongoing and in (Evans et al., 2000), the authors describe a head-mounted infrared-emitting control system that is a 'relative' pointing device and acts like a joystick rather than a mouse. In (Chen et al., 1999) a system containing an infrared transmitter was described. The transmitter was mounted on to the user's eyeglasses, along with a set of infrared receiving modules that substitute the keys of a computer keyboard, and a tongue-touch panel to activate the infrared beam. In (Atyabi et al., 2006) the authors describe a system for translating a user's motion to mouse movements.

Their tracking algorithm relies on detecting specific features such as the eyes or nose to follow head movements across multiple frames.
There are also various commercial mouse alternatives available today. NaturalPoint (NaturalPoint, 2006) markets several head-tracking-based mouse alternatives on their web site. While the benefits are real, these devices still require the user to attach markers either to the head or glasses. Other systems use infrared emitters that are attached to the user's glasses, head-band, or cap. Some systems, for example the Quick Glance system by EyeTech Digital Systems (EyeTech, 2006), place the transmitter over the monitor and use an infrared-reflector that is attached to the user's forehead or glasses. Mouse clicks are generated with a physical switch or a software interface.

3. System architecture

We have implemented a prototype of our system that requires only a web camera and microphone, which fully emulates the functionality normally associated with a mouse device. In this section we provide a short overview of the image-processing algorithms and speech-interaction technologies the system is based on.
The system consists of two signal-processing units and interpretation logic (see Figure 1). Image processing algorithms are applied to the video stream to detect the user's face and follow tracking points to determine head movements. The audio stream is analyzed by the speech recognition engine to determine relevant voice commands. The interpretation logic in turn receives relevant parameters from the signal-processing units and translates these into on-screen actions by the mouse pointer.

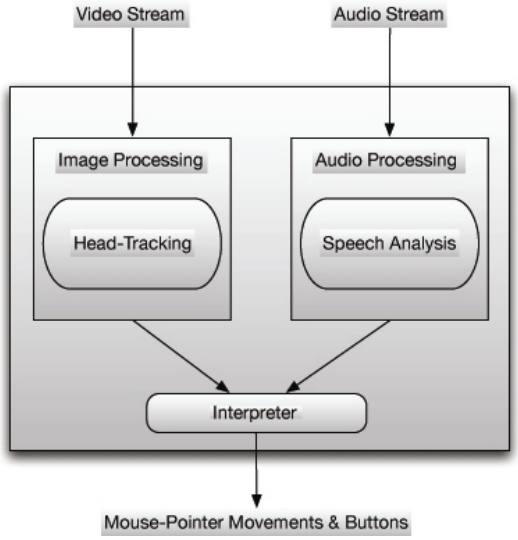


Fig. 1. High-level system overview.

3.1 Face detection

The face-detection component is fundamental to the functioning of our head-tracking system and is based on the Haar-Classifier cascade algorithm. This algorithm was first

introduced by Viola and Jones (Viola & Jones, 2001). It is appearance-based and uses a boosted cascade of simple feature classifiers, providing a robust framework for rapid visual detection of frontal faces in grey scale images. The process of face detection is complicated by a number of factors. Variations in pose (frontal, non-frontal and tilt) result in the partial or full occlusion of facial features. Beards, moustaches and glasses also hide features. Another problem is partial occlusion by other objects. For example, in a group of people some faces may be partially covered by other faces. Finally, the quality of the captured image needs consideration.

The algorithm is based on AdaBoost classification, where a classifier is constructed as a linear combination of many simpler, easily constructible weak classifiers. For AdaBoost learning algorithm to work each weak classifier is only required to perform slightly better than a random guess. For face detection a set of Haar wavelet-like features is utilized. Classification is based on four basic types of scalar features, proposed by Viola and Jones for the purpose of face detection. Each of these features has a scalar value that can be computed efficiently from the integral image, or summed area table. This set of features has recently been extended to deal with head rotation (Lienhart & Maydt, 2002). Weak classifiers are cascaded into a collection of weak classifiers (weak learners) to form a stronger classifier. AdaBoost is an adaptive algorithm to boost a sequence of classifiers, in that the weights are updated dynamically according to the errors in previous learning cycles. The algorithm employed by Viola & Jones (Viola & Jones, 2001) has a face detection cascade of 38 stages with 6000 features. According to Viola & Jones, the algorithm nevertheless achieved fast average detection times. On a difficult dataset, which contained 507 faces and 75 million sub-windows, faces are detected using an average of 10 feature evaluations per sub-window. As a comparison, Viola & Jones show in experiments that their system is 15 times faster than a detection system implemented by Rowley et al. (Rowley et al., 1998). A strong classifier, consisting of a cascade of weak classifiers is shown in Figure 3, where blue icons are non-face and red icons are face images. The aim is to filter out the non-face images, leaving only face images. The cascade provides higher accuracy over a single, weak classifier.

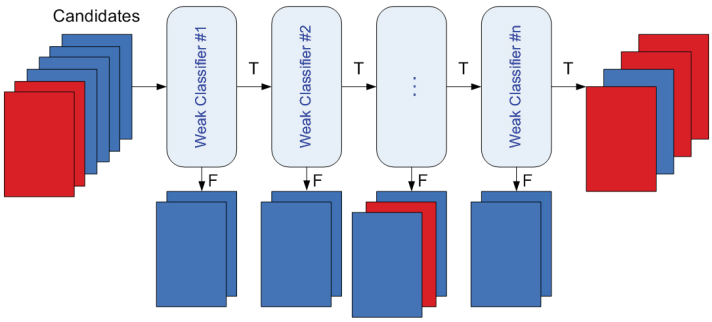


Fig. 3. Weak classifiers are arranged in a cascade. Note that some incorrect classifications may occur, as indicated in the diagram.

3.2 Head-Tracking

Our optical tracking component uses an implementation of the Lucas-Kanade optical flow algorithm (Lucas & Kanade, 1981), which first identifies and then tracks features in an image.

These features are pixels whose spatial gradient matrices have a large enough minimum eigenvalue. When applied to image registration, the Lucas-Kanade method is usually carried out in a coarse-to-fine iterative manner, in such a way that the spatial derivatives are first computed at the coarse scale in the pyramid, one of the images is warped by the computed deformation, and iterative updates are then computed at successively finer scales.

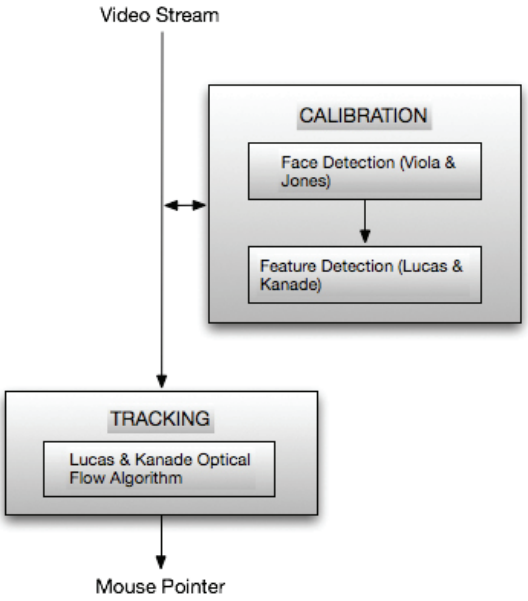


Fig. 4. The head-tracking component. After initial calibration, the video stream is processed in real-time by the tracking algorithm.

Our algorithm restricts the detection of feature points to the face closest to the computer screen to exclude other people in the background. Before tracking is initiated, feature points are marked with a green dot (Figure 5), and are still subject to change.

Fig. 5. The face is detected using the Haar-Classifier cascade algorithm and marked with a yellow square. Green dots mark significant features identified by the Lucas-Kanade algorithm. The image on the right demonstrates performance in poor lighting conditions.

Once tracking has been started, feature points are locked into place and marked with a red dot. It should be noted that the marking of the features and the face is for debugging and demonstration purposes only. Tracking is achieved by calculating the optical flow between two consecutive frames to track the user's head movement, which is translated to on-screen movements of the mouse pointer (see Figure 6).

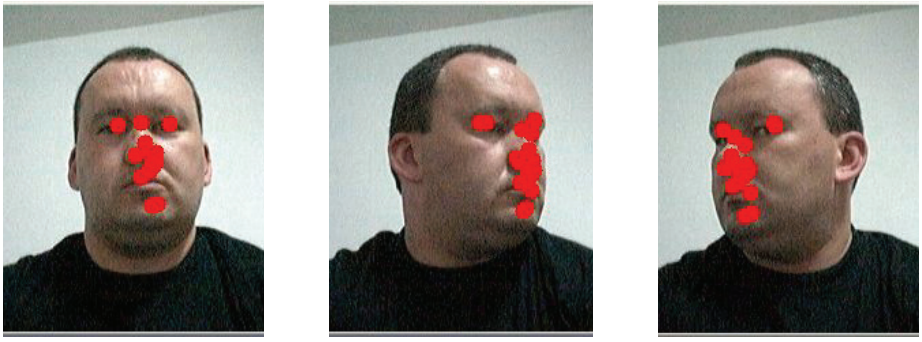


Fig. 6. Feature extraction using modified Lucas-Kanade algorithm.

The user can choose between two operational modes for translating head movements to on-screen mouse pointer movements. Selection of the desired mode may be accomplished in real-time by issuing a voice command. These modes are:

Relative Mode

This mode simulates the joystick control of a mouse pointer. If the system detects deviation of the tracking points from their original position above a certain threshold, the mouse pointer is moved in the given direction by a single pixel. Movement in this direction continues as long as the deviation of the tracking points is maintained. However, the rate of pixels being moved is steadily increased relative to the amount of time elapsed since the movement was initiated. Should the movement be interrupted, the rate of movement is reset to a single pixel. This mode is especially useful where fine control of the mouse pointer is required, and where navigation of the whole screen is still necessary. This makes the system ideal for disabled users who have to make precise on-screen movement, such as artists or engineers.

Absolute Mode

This closely resembles mouse pointer control associated with mouse hardware devices. In this mode, the distance of the tracking points from their original location is translated to the location of the mouse pointer from the center of the screen. The distance the mouse pointer will move away from the center of the screen depends on the resolution setting reported by the operating system.

3.3 Speech recognition

Speech recognition technology has progressed to a level, where users can to some extend control actions on the computer using voice commands. The most popular speech recognition technique is based on hidden Markov models (HMM). HMM is a doubly stochastic model, where the underlying phoneme string and frame-by-frame, surface acoustic realizations are represented as probabilistic Markov processes. HMM systems

classify speech segments using dynamic programming. As an alternative to the frame-by-frame approach, similar speech recognition performance has been achieved by first identifying speech segments, classifying the segments and then using segment scores to recognise words. Speech recognition has matured and with the ability to train the parameters of the model using training data to gain optimal performance, it represents a powerful solution (Rabiner & Juang, 1986).

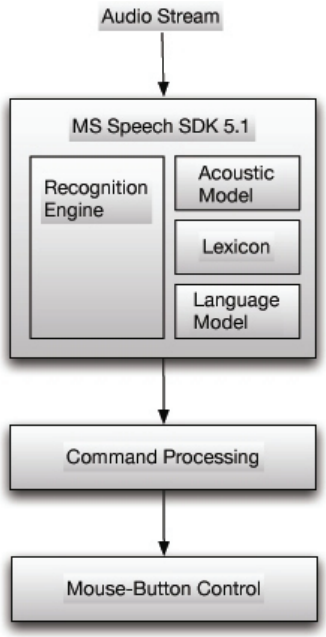


Fig. 7. The speech-recognition components of our head-tracking system.

Our prototype system utilizes the Microsoft Speech SDK 5.1, which is available as a free download from Microsoft (see Figure 7). By issuing the relevant voice command, the user may at any time perform common tasks, such as click-drag and drag-drop operations. The need for double-clicking, for example, represents a great challenge for people with reduced dexterity and motor control in their hands. Our system not only enables the user to execute single- or double-clicks, but also more complex operations. For example, a click-drag operation can be initiated with a vocal command or a file may be picked up and dropped on another folder by issuing a simple sequence of two commands (pick-up and release) in combination with head movements (carrying the object to another location).
Voice commands used to activate buttons are:

- "Click" – single left click
- "Double – double left click
- "Right" – single right click
- "Hold" – single left click and hold (click-drag)
- "Drop" – release the left mouse button after performing a click-drag operation

The commands may be changed to suit the individual user. There is also scope to extend the set of commands for added functionality.

4. Experimental results

In preliminary trials, the effectiveness of our system was tested with a group of 10 volunteers. Each of the subjects had previously used a computer, and was thus able to comment on how our system compares to using a traditional mouse pointing device. Each user was given a brief tutorial on how to use the system, and then allowed to practice moving the cursor for one minute. After the practice period, each user was asked to play a video game (Solitaire). The one minute training time was perceived as sufficient to introduce the features of the system.

Subsequent interviews revealed that users preferred to learn on-the-job, while using our mouse-replacement system to play a computer game. Task completion times were similar to using a traditional mouse device. However, users commented positively on the fact that they were able to control the mouse pointer using head movements, while their hands remained free to perform other tasks. Also positive was the short amount of time users required to become acquainted with the system. They found that using head movements for mouse control quickly become second-nature and issuing voice commands requires no effort at all. All users commented on the possibility of extending the set of voice commands to add functionality to the system.

5. Future work

A future implementation of our system could further extend the speech recognition component, allowing the user, for example, to open applications using vocal commands would eliminate much navigating through menu structures. The speech synthesis component could be extended to provide contextual feedback in the form of reading to the user the names of windows or icons the mouse pointer is currently resting on.

Although the system has been developed and tested on the Microsoft Windows operating system, it is possible to port this technology to Apple and Linux/Unix operating systems. In particular deployment on an open-source platform would open the technology up to a much wider user base. Developers can now develop for Windows, and expect their .NET application to run on Apple and Linux operating systems (Easton & King, 2004). Recent work by the 'Mono Project' and 'Portable .NET'-project has contributed to making .NET code truly portable, presenting the possibility to use the system on these platforms with minimal change.

6. Conclusion

The prototype system has proven to be robust, being able to tolerate strong variations in lighting conditions. We see potential for our approach being integrated in interfaces designed specifically for users with disabilities. Especially attractive should be the fact that it provides a low-cost means of human-computer interaction requiring the most basic of computer hardware and its reliance upon established approaches in the areas of computer vision and speech interaction. It may also prove useful for other applications, for example, where it is necessary to activate controls on a computer interface while at the same time performing precision work using both hands. Another application area could be computer games. Furthermore, the modular architecture of the system allows for ready integration in any number of software projects requiring a low-cost and reliable head-tracking component.

7. References

Atyabi, M., Hosseini, M. S. K., & Mokhtari, M. (2006). The Webcam Mouse: Visual 3D Tracking of Body Features to Provide Computer Access for People with Severe Disabilities. *Proceedings of the Annual India Conference*, pp. 1-6, ISBN: 1-4244-0369-3 , New Delhi, September 2006

Camus, T. A. (1995). Real-time optical flow. *Technical Report CS-94-36*. The University of Rochester. Rochester, New York

Chen Y. L., Tang F. T., Chang W. H., Wong M. K., Shih Y. Y., & Kuo T. S. (1999). The new design of an infrared-controlled human-computer interface for the disabled, *IEEE Trans. Rehab. Eng.*, Vol. 7, No. 4, December 1999, 474–481, ISSN: 1063-6528

Cole, R., Mariani, J., Uszkoreit, H., Varile, G., B., Zaenen, A., & Zampolli, A. (1998). *Survey of the State of the Art in Human Language Technology,* Cambridge University Press, ISBN-10: 0521592771, Cambridge

Easton, M. J. & King, J. (2004). *Cross-Platform .NET Development: Using Mono, Portable .NET, and Microsoft .NET*. Apress Publishers, ISBN: 1590593308, Berkeley

Evans, D. G., Drew, R., & Blenkhorn, P. (2000). Controlling mouse pointer position using an infrared head-operated joystick. *Rehabilitation Engineering, IEEE Transactions on [see also IEEE Trans. on Neural Systems and Rehabilitation],* Vol. 8, No. 1, March 2000, 107-117, ISSN: 1063-6528

EyeTech Digital Systems Inc. (2006). On-line Product Catalog: Retrieved May 29 from http://www.eitetechds.com/products.htm

Lienhart, R. & Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. *Proceedings of the International Conference on Image Processing (ICIP)*, pp. I-900- I-903, September 2002, IEEE, Rochester, New York, USA

Loewenich, F., & Maire, F. (2006). A Head-Tracker Based on the Lucas-Kanade Optical Flow Algorithm. In: *Advances in Intelligent IT - Active Media Technology 2006,* Li, Y., Looi, M. & Zhong, N. (Ed.), Vol. 138, pp. 25-30, IOS Press, ISBN: 1-58603-615-7, Amsterdam, Netherlands

Lucas, B. D., & Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pp. 674—679, Vancouver

NaturalPoint Inc. (2006). On-line Product Catalog: Retrieved May 29 from http://www.naturalpoint.com/trackir/02-products/product-TrackIR-4-PRO.html

Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. ASSP Magazine, IEEE, Vol. 3, No. 1, 4-16, ISSN: 0740-7467

Viola, P., & Jones, J. J. (2001). Robust Real-Time Face Detection. *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pp. 122-130, ISBN:0-7803-7965-9, July 2003, IEEE Computer Society, Washington, DC, USA

Viola, P., & Jones, J. J. (2004) Robust Real-Time Face Detection. *International Journal of Computer Vision*, Vol. 57, No. 2, May 2004, 137-154, ISSN: 0920-5691

25

Arabic Dialectical Speech Recognition in Mobile Communication Services

Qiru Zhou[1] and Imed Zitouni[2]
[1]*Bell Labs, Alcatel-Lucent, NJ,*
[2]*IBM T.J. Watson Research Center*
USA

1. Introduction

We present in this chapter a practical approach in building Arabic automatic speech recognition (ASR) system for mobile telecommunication service applications. We also present a procedure in conducting acoustic modelling adaptation to better take into account the pronunciation variation across the Arabic speaking countries.

Modern Standard Arabic (MSA) is the common spoken and written language for all the Arab countries, ranging from Morocco in the west to Syria in the East, including Egypt, and Tunisia. However, the pronunciation varies significantly from one country to another to a degree that two persons from different countries may not be able understand each other. This is because Arabic speaking countries are characterized by a large number of dialects that differ to an extent that they are no longer mutually intelligible and could almost be described as different languages. Arabic dialects are often spoken rather than written varieties. MSA is common across the Arab countries, but it is often influenced by the dialect of the speaker. This particularity of the Arabic countries constitutes a practical problem in the development of a speech-based application in this region; suppose a speech application system is built for one country influenced by one dialect, what does it take to adapt the system to serve another country with a different dialect region? This is particularly challenging since resource to build accurate speaker independent Arabic ASR system for mobile telecommunication service applications are limited for most of the Arabic dialects and countries.

Recent advances in speaker independent automatic speech recognition (SI-ASR) have demonstrated that highly accurate recognition can be achieved, if enough training data is available. However, the amount of available speech data that take into account the dialectal variation of each Arabic country is limited, making it challenging to build a high performance SI-ASR system, especially when we target specific applications. Another big challenge when building an SI-ASR is to handle speaker variations in spoken language. These variations can be due to age, gender, educational level as well as the dialectical variants of Arabic language. Usually an ASR system trained in one regional variation exhibits poorer performance when applied to another regional variation. Three problems may arise when a SI-ASR system built for one dialect but applied to target users with a different dialect: (1) Acoustic model mismatch, (2) Pronunciation lexicon mismatch and (3) Language model mismatch.

In the following we show how to build an Arabic speech recognition system for mobile telecommunication service applications. Furthermore we show how to adapt the acoustic model and the pronunciation lexicon of ASR systems for the Modern Standard Arabic to better take into account the pronunciation variation across the Arabic countries. This is a challenging problem especially when not enough data is available for every country. One of the topics we address is the dialect adaptation across the region. We investigate how a Modern Standard Arabic ASR system trained on one variation of the language in one country A can be adapted to perform better in a different Arab speaking country B, especially where only small amount of data related to country B is available. We show in this chapter how we take into account the pronunciation variation and how we adapt the acoustic model.

Experiments are conducted on Orientel (Oriental, 2001) database covering, in addition to Modern Standard Arabic, Arabic dialect spoken in Morocco, Tunisia, Egypt, Jordan, and United Arab Emirate. Results show an interesting improvement is achieved by using our adaptation technique. In this work, we experiment dialect adaptation from Tunisian (Maghreb Arabic) dialect to Jordan (Levantine Arabic) MSA dialect.

In section 2 and section 3, we discuss SI-ASR model training and adaptation techniques that are language neural. In section 4, we demonstrate a real-world practise on building Arabic speech recognition systems with both model estimation techniques and adaptation techniques. We then conclude the paper in section 5.

2. Hidden Markov Model parameter estimation for speaker independent speech recognition

Although other pattern recognition methods have been developed in the history of speech recognition research and development, the hidden Markov model (HMM) method is by far the most successful method used in speech recognition. Almost all modern speech recognition research and commercial systems are using some form of HMM to model the spectral and temporal variations of basic speech units. HMM is a very powerful statistical method of characterizing the observed data samples of a discrete-time series. For speaker independent speech recognition, HMM provides an efficient way to build parametric models on large amount of observation samples (e.g., speech data collection from many speakers.). Incorporate with the dynamic programming principle, it can be used for pattern segmentation and classification of a time-varying sequence. HMM and dynamic programming are the two key technologies for most of the modern SI-ASR systems today.

2.1 Hidden Markov Model

The basic HMM theory for pattern classification was developed by Baum and his colleagues during 1960s and 70s (Baum 1966~1970). The hidden Markov model is a statistical model that uses a finite number of states where the output observation is a random variable X generated according to a output probabilistic function associated with each state. It can be viewed as a double-embedded stochastic process with an underlying stochastic process (the state sequence) not directly observable (hence "hidden"). A hidden Markov model is defined as:

1. $\boldsymbol{O} = \{o_1, o_2, \cdots, o_M\}$ −An output observation alphabet.
2. $\boldsymbol{S} = \{s_1, s_2, \cdots, s_N\}$ − A set of states representing the state space of the model.

3. $A = \{a_{ij}\}$ −A transition probability matrix, where a_{ij} is the probability of taking a transition from state i to state j:

$$a_{ij} = P(s_t = j \mid s_{t-1} = i) \tag{1}$$

4. $\boldsymbol{B} = \{b_i(k)\}$ −An output probability matrix, where $b_i(k)$ is the probability of emitting symbol o_k when state I is entered. Let $\boldsymbol{X} = X_1, X_2, \cdots, X_t, \cdots$ be the observed output of the HMM. The state sequence $S = s_1, s_2, \cdots, s_t, \cdots$ is not observed (hidden). Therefore, $b_i(k)$ can written as:

$$b_i(k) = P(X_t = o_k \mid s_t = i) \tag{2}$$

5. $\boldsymbol{\pi} = \{\pi_i\}$ −An output initial state distribution, where

$$\pi_i = P(s_0 = i), \qquad 1 \le j \le N \tag{3}$$

All probabilities must satisfy the following properties

$$a_{ij} \ge 0,\ b_i(k) \ge 0, \pi_i \ge 0,\ \forall \text{ all } i, j, k \tag{4}$$

$$\sum_{k=1}^{N} a_{jk} = 1,\ \sum_{k=1}^{M} b_j(k) = 1,\ \sum_{i=1}^{N} \pi_i = 1 \tag{5}$$

In continuous observation density HMMs, the observations are continuous signals (vectors), the observation probabilities then often be replaced by finite mixture probability density functions (pdf):

$$b_j(o) = \sum_{k=1}^{M} c_{jk} N(o, \mu_{jk}, \sigma_{jk}), \qquad 1 \le j \le N \tag{6}$$

where o is the observation vector being modelled. c_{jk} is the mixture coefficient for the kth mixture in state j and N is any log-concave or elliptically symmetric density (in speech recognition, Gaussian pdf is commonly used). Without loss of generality, we can assume that N is Gaussian in (3) with mean vector $\boldsymbol{\mu_{jk}}$ and covariance matrix $\boldsymbol{\sigma_{jk}}$ for the kth mixture component in state j. The mixture gain c_{jk} satisfy

$$\sum_{k=1}^{M} c_{jk} = 1, 1 \le j \le N \tag{7}$$

$$c_{jk} \ge 0, 1 \le j \le N, 1 \le k \le M \tag{8}$$

so that the the pdf is properly normalized, i.e.,

$$\int_{\infty}^{\infty} b_j(o) do = 1, \qquad 1 \le j \le N \tag{9}$$

In practical acoustic speech signal modelling, we generally assume that HMM is in left to right format with equal transitional probability

$$a_{ij} = a_{ii} = 1/2, \qquad 1 \le i < N, i < j \tag{10}$$

$$a_{ij} = 0, \qquad j > i + \Delta i \tag{11}$$

$$a_{NN} = 1, a_{Ni} = 0, \; i < N \tag{12}$$

and π is uniform distribution to simply the model topology. Hence the model parameter estimation becomes estimate probability matrix in equation (2) for discrete HMMs, and Gaussian pdf parameters in equation (6) for continuous density HMMs, given training speech data set and model topology. A typical 3-state left to right HMM phoneme model topology is shown in Figure 1.

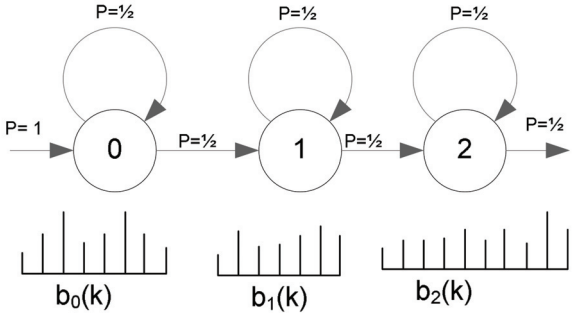


Fig. 1. A 3-state hidden Markov phoneme model topology example

2.2 HMM parameter estimation

Given the above form HMM, there are three basic problems of interests must be solved to for the HMM to be useful in real-world applications (Rabiner, 1989)

1. **The evaluation problem**
   Given the observation sequence $\boldsymbol{O} = (o_1, o_2, \ldots, o_T)$, and a model $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, how do we compute $P(\boldsymbol{O} \mid \lambda)$, the probability of the observation sequence?
2. The decoding problem
   Given the observation sequence $\boldsymbol{O} = (o_1, o_2, \ldots, o_T)$ and the model λ, what is the most likely state sequence $\boldsymbol{q} = \{q_1, q_2 \ldots q_T\}$ in the model that produces the observations?
3. The learning problem
   How do we adjust the model parameters $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ to maximize $P(\boldsymbol{O} \mid \lambda)$?

In this chapter, we concentrate on the third problem which is the problem on how to train HMMs, given a set of speech training data and model topology. Combine with the forward and backward procedure designate to solve Problem 1 with Baum-Welsh EM (expectation-maximization) method (Dempster, 1977) using maximum likelihood (ML) approach, an iterative procedure is developed to estimate and re-estimate continuous density HMM model parameters efficiently (Rabiner, 1989):

$$\overline{c}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)} \tag{13}$$

$$\overline{\mu}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \bullet \boldsymbol{O}_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{14}$$

$$\overline{\sigma}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \bullet (\boldsymbol{O}_t - \mu_{jk})(\boldsymbol{O}_t - \mu_{jk})'}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{15}$$

where $\gamma_t(j, k)$ is the probability of being in state j at time t with the k^{th} mixture component account for o_t:

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \right] \left[\frac{c_{jk} N(o, \mu_{jk}, \sigma_{jk})}{\sum_{m=1}^{M} c_{jm} N(o, \mu_{jk}, \sigma_{jk})} \right] \tag{16}$$

Forward variable

$$\alpha_t(i) = P(o_1 o_2 ... o_t, q_t = i \mid \lambda) \tag{17}$$

Can be calculated efficiently using inductive algorithm:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(o_1)\,,\ 1 \le \mathrm{i} \le \mathrm{N} \tag{18}$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(o_{t+1}), \begin{matrix} 1 \le t \le T-1 \\ 1 \le j \le N \end{matrix} \tag{19}$$

3. Termination

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{20}$$

Similarly, the backward variable

$$\beta_t(i) = P(o_{t+1} o_{t+2} ... o_T \mid q_t = i, \lambda) \tag{21}$$

Can be calculated:

1. Initialization

$$\beta_T(i) = 1\,,\ 1 \le \mathrm{i} \le \mathrm{N} \tag{22}$$

2. Induction

$$\beta_t(j) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \text{ , t = T-1, T-2, …,1, } 1 \le i \le N \tag{23}$$

Another commonly used approach is segmental ML training (also referred as Viterbi training)(Rabiner, 1986) which using K-means model parameter estimation and Viterbi algorithm re-segment the state sequence iteratively until model convergence.
In following sections, we discuss issues to train real-world application continuous density HMMs. Similar procedure applies to HMM training with tied mixture density functions and tied-state HMMs, with moderate algorithm modifications.

2.3 HMM topology

In HMM speech modelling, we assume each of the HMM state capture part of the speech as a quasi-stationary segment (20 ~ 30 milliseconds). For left-to-right HMM, all we need to decide is the number of states. This decision is depend on the word pronunciation and duration of the speech the HMM to model. For a long word or a phrase model, a HMM to model it may need 10 or more states. For a sub word or a phoneme model, 3 to 5 states are commonly used. Since silence and pause in speech is stationary, 1 to 3 states will be sufficient. Also for training purpose, available speed data to train the model will affect the number of states decision since there must be sufficient data (speech feature vectors) to train each of the states.

2.4 Initial estimation

A key question is that how to choose initial estimates of the HMM parameters so that the local maximum is equal to or as close as possible to the global maximum of the likelihood function? Experimentally, we observed that for raw training data, random or uniform initial estimates of $\boldsymbol{A}$ (Equation (1)) and $\boldsymbol{\pi}$ (Equation (3)) are adequate but a good initial estimation of $\boldsymbol{B}$ is required for continuous density HMM estimation. Here the most difficult problem is model state level segmentation that is hard to do by human labelling process. Here are a few practices in real-world application cases:

1. If there is a pre-built HMM that closes to the language acoustics of the target HMM, the pre-built HMM can be used to perform initial segmentation of the data to get better word, phoneme, and state boundaries. In case of acoustic feature mismatch, a signal and model conversion and rescaling is required.
   In this chapter, we will show how to use HMM trained on studio recorded, wideband speech to bootstrap HMM to be trained using narrow band telephony and mobile speech.
2. If there is human labelled speech data at speech element level (phoneme for sub word system, word for whole word recognition system.), we can use a random or uniform segmentation approach estimated the state segmentation to initialize the training. Then K-mean clustering ((MacQueen, 1967), (Hartigan & Wong, 1979)) can be used to create initial model. We can further refine the model using segmental kmeans iteration method (Rabiner 1986) or EM algorithms described above.
3. Furthermore, model adaptation and cross dialect and/or language phoneme mapping may be used for bootstrapping as well.

2.5 Effects of insufficient training data

In general, there are always finite amount of data to train HMM for a given target language and/or dialect. Hence we find always an inadequate number of occurrences of low probability events to give good estimates of the model parameters. There are literature discusses on effects of inadequate data and solutions on this issue ((Jelinek, 1980), (Rabiner & Juang 1992)). Here are some practices to address this issue:

1. Try to increase the training data to fill the data gap. But this is not always possible due to data collection availability or associated cost.
2. Decrease the model complexity (reduce model set size to remove rare seen acoustic event models, or reduce the mixture number for HMM with low count of available acoustic event). But this may produce model set with inadequate coverage of a language.
3. Using a floor value to replace the observation probability or density function.
4. For context dependent HMM, using tied state and furthermore tied mixture techniques to cover certain acoustic context events with low count.

Other smoothing techniques such as deleted interpolation (Jelinek, 1980) and Bayesian smoothing have been suggested to compensate the sparse data difficulties.

2.6 Duration modeling

Although HMM is a powerful statistical modelling tool, there are limitations of HMM to model real-world speech signals. One of the well known limitations is the duration model, which is the exponential decrease as represented as

$$d_i(t) = a_{ii}^t(1 - a_{ii}) \tag{24}$$

To improve duration modelling, an explicit time duration distribution can be built for each state. This duration distribution function parameters can be estimated from training data. Or a simple histogram distribution can be created which limit to finite number of duration time length.

3. Adaptation and corrective training yechniques

In real-world speech applications, it often requires a speech recognition system to adapt its acoustic and language models to new situations. Most of the speaker dependent system do need adapt to it's user in order to be produce satisfactory performance. Even a well trained robust speaker independent system that can accommodate wide range of speakers and environment may always have the situation that there is a mismatch between the model and the operating condition. If there is adequate training data and computing resource, the SI-ASR system may be re-trained to the new environment and users. But in most of the real-world situations, there is neither enough data nor resource to retrain the system. An alternative solution is to apply adaptation algorithms using limited data and computing resource to reduce the mismatches mentioned above to improve system performance. Therefore, a well built real-world speech recognition should have adaptation capability to minimize the possible mismatch in short time and minimum calibration data (e.g., a few utterances from a speaker can adapt the system in favour of the speaker using it.).

Many effective adaptation techniques have been developed to improve real-world speech applications. Basically, these adaptation techniques can be divided to two categories: model

adaptations and channel adaptations. The former adaptation changes acoustic and/or language model parameters (linear or non-linear transformations) to improve recognition accuracy. It is more suitable for speaker variations, unseen language situations and accents not covered in model training data. In most of the case, the system apply model adaptation are tuned to a specific speaker or a new dialect group of speakers. The latter is mainly addressing acoustic channel environment situation changes and improves recognition by tuning the system to be more environments robust. Channel adaptation (or front end adaptation) algorithms such as dynamic cepstral mean subtraction and signal bias removal (Rahim, 1994) are become a standard component to most of the modern speech recognition systems.

In this chapter, we are focusing on the former adaptation to address situation changes require model parameter changes for new dialects and vast acoustic environment changes. The most common adaptation techniques are maximum a posterior (MAP) and maximum likelihood linear regression (MLLR) algorithms. We briefly describe these two adaptation methods in the following sections.

3.1 MAP model adaptation

Maximum a posterior (MAP) estimation uses Bayesian learning framework to obtain estimation of random HMM parameter vector $\boldsymbol{\lambda}$ (Lee,1996). For a given set training / adaptation data $\mathbf{x}$, the conventional ML estimation assumes that $\boldsymbol{\lambda}$ is fixed but unknown and solves the equation

$$\lambda_{ML} = \arg\max_{\lambda} f(x \mid \lambda) \tag{25}$$

where f ($\mathbf{x} \mid \boldsymbol{\lambda}$) is the likelihood of observation $\mathbf{x}$. i.e., MAP formulation assumes that the parameter $\boldsymbol{\lambda}$ to be a random vector with a know distribution f. Furthermore, we assume there is a correlation between the observation vectors and the parameters so that a statistical inference of $\boldsymbol{\lambda}$ can be made using a small set of adaptation data $\mathbf{x}$. Before making any new observations, $\boldsymbol{\lambda}$ is assumed to have a prior density g($\boldsymbol{\lambda}$) and new data are incorporated, $\boldsymbol{\lambda}$ is characterized by a posterior density g($\boldsymbol{\lambda} \mid$x). The MAP estimate maximizes the posterior density

$$\lambda_{MAP} = \arg\max_{\lambda} g(\lambda \mid x) = \arg\max_{\lambda} f(x \mid \lambda) g(\lambda) \tag{26}$$

Since the parameters of a prior density can also be estimated from an existing HMM $\boldsymbol{\lambda}_0$, this framework provides a way to combine with newly acquired data $\mathbf{x}$ in an optimal way.

Let $\mathbf{x} = (x_1, \ldots, x_N)$ be a set of scalar observations that are independent and identical distributed (i.i.d.) Gaussian distribution with mean m and variance σ^2. Here assume that the mean m is a random variable and the variance σ^2 is fixed. I can be shown that the conjugate prior for m is also Gaussian with mean μ and variance κ^2. If we use the conjugate prior for the mean to perform MAP adaptation, then the MAP estimation for the parameter $\mathbf{m}$ is

$$\tilde{m} = \frac{N\kappa^2}{\sigma^2 + N\kappa^2}\bar{x} + \frac{\sigma^2}{\sigma^2 + N\kappa^2}\mu \tag{27}$$

where N is the total number of training samples and $\bar{x}$ is the sample mean.

Using a prior density

$$g(\sigma^2) = \begin{cases} cons\tan t, if\sigma^2 \geq \sigma^2_{\min} \\ 0, otherwise \end{cases} \tag{28}$$

The MAP estimate of the variance is

$$\tilde{\sigma}^2 = \begin{cases} S_x, ifS_x \geq \sigma^2_{\min} \\ \sigma^2_{\min}, otherwise \end{cases} \tag{29}$$

where S_x is the sample variance of **x**.
From (27), we can see that the MAP estimation of the Gaussian mean is a weighted average of the prior mean μ and the sample mean.
The MAP training can be iterative as well. This requires an initial estimate of model parameters.

3.2 MMLR model adaptation

We can use a set of linear regression transformation functions to map both mean and covariance (variance for diagonal covariance matrix) in order to maximize the likelihood of the adaptation data (Leggetter, 1995). Since the transformation parameters can be estimated from relatively small amount of adaptation data, it is very effective for rapid adaptation. The maximum likelihood linear regression (MLLR) has been widely used to obtain adapt models for either a new speaker or a new environment condition.
Specifically, MLLR is a model adaptation method that estimates a set of linear transformations for the mean and variance parameters of a Gaussian mixture HMM system. The transformation matrix used to give a new estimate of the adapted mean is given by

$$\tilde{\mu} = W\xi \tag{30}$$

Where W is the n x (n+1) transformation matrix (n is the dimensionality of the data) and ξ is the extended mean vector

$$\xi\hat{} = (\mathrm{w}\ \mu_1 \mu_2 \ldots \mu_\mathrm{n})^\mathrm{T} \tag{31}$$

where w is a bias offset (normally a constant). It has been show that W can be estimated as

$$w_i = k^{(i)} G^{(i)-1} \tag{32}$$

where w_i is the i^{th} row of W. and

$$G^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma^2_{mi}} \xi_m \xi_m^T \sum_{t=1}^{T} L_m(t) \tag{33}$$

and

$$k^{(i)} = \sum_{m=1}^{M} \sum_{t=1}^{T} L_m(t) \frac{1}{\sigma^2_{mi}} o_i(t) \xi_m^T \tag{34}$$

where $L_m(t)$ is the occupancy probability for the mixture component m at time t.
Similarly, variance transformation matrix can be calculated iteratively.

3.3 Comparison of MAP and MLLR

In general, MLLR performs better when adaptation data is limited. But when more adaptation data are available, MAP becomes more accurate. Also MLLR can combine with MAP to improve performance for both less and more adaptation data situations. When more adaptation data is available, MAP is more computation efficient. For more details, readers may refer to (Wang et al., 2003).

4. Experiments

In this section, we present and discuss our experiment on building SI-ASR for Arabic speech recognition for dialect variations. In this work, we use phoneme level sub word HMM models for the SI-ASR. Therefore the generic techniques on model estimation and adaptation all can apply to this work. The main purposes in this work is to bootstrap SIASR systems for telecommunication network applications, from a SI-ASR system built on studio recorded speech. Obviously, there is a significant acoustic mismatch and dialect mismatch. Another problem we need to solve is the phonetic mapping of the different systems. First, we introduce the speech corpora used in the experiment.

4.1 Speech corpora in experiments

The following speech copora are used in our experiments

4.1.1 West Point speech corpus

Since we don't have an initial bootstrap wireless telephony acoustic model, we bootstrap the experiment using an existing Arabic acoustic model trained on West Point Speech Corpus (WPA) (LaRocca, 2002) which is a collection of wideband studio speech recording.
The West Point Arabic Speech Corpus (WPA) contains MSA speech data that was collected by members of the Department of Foreign languages at the United States Military Academy at West Point and the Center for Technology Enhanced Language Learning (CTELL). The original purpose of this corpus was to train acoustic models for automatic speech recognition that could be used as an aid in teaching Arabic language. The corpus consists of 8,516 speech files, total 11.42 hours of speech data. Each speech file represents one person reciting one prompt from one of four prompt scripts. The utterances were recorded using a Shure SM10A microphone and a RANE Model MS1 pre-amplifier. The files were recorded as 16-bit PCM audio files, with a sampling rate of 22.05 KHz. Approximately 7200 of the recordings are from native informants and 1200 files are from non-native informants. Overall, there are about 30% of non-native speakers contributed about 10% of the total speech recording. There is no information given on the origin countries/regions of the native speakers in the database. There is 1128 distinct orthographic word in the WPA lexicon. The WPA phoneme symbol set is shown in Table 1.

4.1.2 Orientel corpora

OrienTel (OrienTel, 2001) is a data collection and research project driven by an international industrial and academic consortium. The goal of OrienTel is to enable the project's

Symbol	Description		Symbol	Description
C	voiced pharyngeal fricative		iy	high front tense vowel
D	velarized voiced alveolar stop		j	voiced palato-alveolar fricative
G	voiced velar fricative		k	voiceless velar stop
H	voiceless pharyngeal fricative		l	voiced alveolar lateral
Q	voiceless glottal stop		m	voiced bilabial nasal
S	velarized voiceless alveolar fricative		n	voiced alveolar nasal
T	velarized voiceless alveolar stop		q	voiceless uvular stop
TH	velarized voiced interdental fricative		r	voiced alveolar flap
Z	voiced interdental fricative		s	voiceless alveolar fricative
ae	low front vowel		sh	voiceless palato-alveolar fricative
ah	low back vowel		sil	silence
aw	back upgliding diphthong		sp	short pause
ay	front upgliding diphthong		t	voiceless alveolar stop
b	bilabial voiced stop		th	voiceless interdental fricative
d	voiced alveolar stop		uw	high back rounded vowel
ey	upper mid front vowel		w	voiced bilabial approximant
f	voiceless labiodental fricative		x	voiceless velar fricative
g	voiced velar stop		y	voiced palatal approximant
h	voiceless glottal fricative		z	voiced alveolar fricative
ih	high front lax vowel			

Table 1. West Point Arabic Speech Database Phoneme Set

participants to design and develop speech-based multilingual interactive communication services for the Mediterranean and the Middle East, ranging from Morocco in the West to the Gulf States in the east, including Turkey and Cyprus. These applications will typically be implemented on mobile and multi-modal platforms such as GSM or UMTS phones, personal digital assistants (PDAs) or combinations of the two.

Examples of applications are unified messaging, information retrieval, customer care, banking, and service portals. To achieve this goal, the consortium conducts various surveys of the OrienTel region, compiles a set of linguistic databases, conducts research into ASR-related problems the OrienTel languages hold and develops demonstrator applications bearing evidence of OrienTel's multilingual orientation.

During OrienTel project, 21 databases (Zitouni et. al., 2002) were collected to cover the four broad Arabic dialect regions, namely Mahgreb Arabic, Egyptian Arabic, Levantine Arabic and Gulf Arabic. Additional languages of commercial interest were also collected in the OrienTel region are English, French, German, Cypriote Greek, Turkish and Hebrew. Three

Country	1st linguistic variety	2nd linguistic variety	3rd linguistic variety	Patner
United Arab Emirates	Colloquial Gulf Arabic as spoken in the UAE	Standard Arabic as spoken in the UAE	English	Scansoft (Now Nuance)
Jordan	Colloquial Levantine Arabic as spoken in Jordan	Standard Arabic as spoken in Jordan	English	Lucent Technologies (Now Alcatel-Lucent)
Egypt	Colloquial Egyptian Arabic	Standard Arabic as spoken in Egypt	English	IBM
Morocco	Colloquial Maghreb Arabic spoken in Morocco	Standard Arabic as spoken in Morocco	French	ELDA/UPC
Tunisia	Colloquial Maghreb Arabic spoken in Tunisia	Standard Arabic as spoken in Tunisia	French	UPC/ELDA
Israel, Palestine	Colloquial Levantine Arabic as spoken in Israel and Palestine	Hebrew as spoken in Israel		NSC
Cyprus	Greek as spoken in Cyprus	English as spoken in Cyprus		Knowledge /Univ. of Patras
Turkey, Germany	Turkish as spoken in Turkey	German spoken by Turks in Germany		Siemens

Table 2. Collected linguistic varieties in the OrienTel region and patners

languages were collected for each of the Arabic countries: 500 speaker collection of the Modern Standard Arabic (MSA), 1000 speakers' collection of the Modern Colloquial Arabic (MCA), and 500 speakers of the 3rd language used in business in this country (English, French, German, etc.). The databases produced are shown in Table 2. The general speech contents classes of the databases are defined in Table 3. All OrienTel collections are phonetically labelled with SAMPA (SAMPA, 2005) phoneme system and symbol set (c.f. Table 5).

Two of the Orientel speech databases used in our study are MSA Tunisia and MSA Jordan (c.f. Table 4). The common features for the data collections are as the following: Each of the speakers recorded 51 utterances in his/her session, within the contents defined in Table 1. The total speech duration approximates to 45 hours for each of the databases. The recordings were performed from offices, homes, public places and on streets. The speech data is sufficient for acoustic training and testing for various application domains. About 80% of the data is defined as the training set and 20% is defined as the testing set. The division of the training set and testing set is based on recording sessions. Therefore no

speaker is in both sets. Some of the content classes suitable for training and other linguistic research (the A, W, S, X contents in Table 3) are excluded from testing set. Also training and testing sets were divided as even as possible per gender and age group distributions. The speaker ages are in the range of 16 to 60. The lexicon size for both of the databases is around 26,000.

Code	# of Uttr.	Utterance Description
I, B	3	isolated digits
C	8	digit/number strings
N	2	natural number
M	2	currency amount (local and foreign)
Q	2	yes/no questions
D	3	Dates
T	1	Times
A	6	application keyword/keyphrases
E	1	word spotting phrases
O	3	directory assistance names
L	2	Spellings
W	4	phonetically rich words
S	9	phonetically rich sentences
X	5	spontaneous items (for control use)

Table 3. OrienTel Database Content Definitions

MSA database	Speakers (Males/Females)	Dialect Region	Mobile Network (%)
Tunisia	598 (359/239)	Maghreb	70.0
Jordan	556 (288/268)	Levantine	70.5

Table 4. Orientel Tunisia and Jordan MSA Databases

All OrienTel speech corpora are collected mainly on mobile telephone networks with smaller portion of speech collected on wired line telephone networks. All the telephone networks are digital.

4.2 Experimental results

Our experiment has the following steps:

1. Speech data feature extraction and acoustic model structure selection.
2. Bootstrap using existing, wide band MSA ASR system to a narrow band ASR system.
3. Retrain and global MLLR adaptation to Tunisian MSA ASR system.
4. Dialect adaptation to Jordan MSA ASR system.

We illustrate in Figure 3 the different steps we follow in the next few sub-sections in order to build an accurate SI-ASR system for Jordanian MSA. We remind the reader that our goal is to show how we can adapt an Arabic MSA SI-ASR system built on speech data spoken in one country (e.g., Tunisia) to be more effective when used by speakers in another country (e.g., Jordan). Here we assume that we don't have enough data in the target language (e.g., Jordan) to train a complete system from scratch. This is why we consider the Tunisian MSA

SAMPA Symbol	WPA Symbol	SAMPA Keyword	Arabic Orthography		SAMPA Symbol	WPA Symbol	SAMPA Keyword	Arabic Orthography
Consonants: Plosives					**Nasals**			
b	b	ba:b	باب		m	m	ma:l	مال
t	t	tis?`	تسع		n	n	nu:r	نور
d	d	da:r	دار		**Trill**			
t`	T	t`a:bi?`	طابع		r	r	rima:l	رمال
d`	D	d`arab	ضرب		**Lateral**			
k	k	kabi:r	كبير		l	l	la:	لا
g	g	gami:l	جميل		l`	l	?al`l`ah	الله
?	Q	?akl	أكل		**Semivowels**			
p	-	paris	برس		w	w	wa:hid	واحد
Consonants: Fricatives					j	y	jawm	يوم
f	f	fi:l	فيل		**Vowels**			
v	-	nivi:n	نفين		i	ih	D`il	ظل
T	th	Tala:T	ثلاث		a	ah	X\al	حل
D	TH	Dakar	ذكر		u	uw	?`umr	عمر
D`	Z	D`ala:m	ظلام		i:	iy	?`i:d	عيد
s	s	sa?`i:d	سعيد		a:	ae	ma:l	مال
z	z	zami:l	زميل		u:	uw	fu:l	فول
s`	S	s`aGi:r	صغير		-	aw		
S	sh	Sams	شمس		-	ay		
Z	j	Zami:l	جميل		-	ey		
x	x	xit`a:b	خطاب					
G	G	Garb	غرب					
X\	H	X\ilm	حلم					
?` (?\)	C	?`alam	علم					
h	h	hawa:?	هواء					

Table 5. SAMPA to WPA phone set mapping

SI-ASR system as our baseline. The third step described earlier in this section shows how to built the Tunisian MSA SI-ASR system and the fourth step describes how the adaptation is conducted to build a more accurate SI-ASR system used by speakers from the target language (e.g. Jordanian).

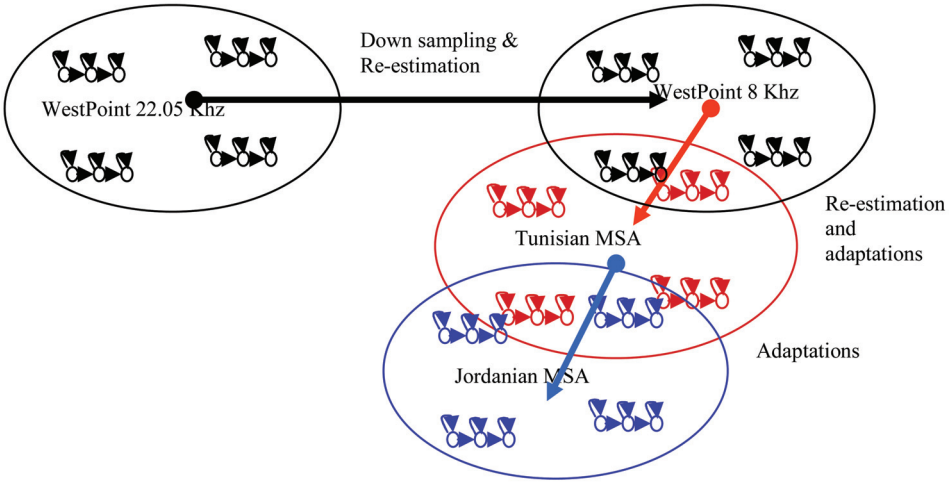


Fig. 3. Arabic Acoustic Model Dialect Adaptation

4.2.1 Feature extraction and model structure selection

Our experiment speech data feature extraction and model structure selections are as the following:

- We select 39 MFCC feature vector for speech feature extraction (12 c, 12 Δc, 12 ΔΔc, e, Δe, and ΔΔe) (Young, 2006).
- The models are continuous density HMMs. All the phone models are 3-state and 16 mixtures. The leading and ending silence/noise model is 3-state and 16 mixtures. The inter-word silence model is 1-state and 16 mixtures.
- Only mono-phone models are used in our preliminary research. We want to use this simple acoustic mapping to identify the basic acoustic and language features before moving to advanced acoustic modeling (e.g., context dependent models) in the next step.

4.2.2 Acoustic model bootstrap

In order to use the wide band recorded WPA speech data to bootstrap a 8 KHz narrow band speech recognition system for mobile voice communication applications, we down sampled the 22050 Hz training data to 8 KHz sampling, along with a 300-3400 Hz band pass filtering to approximate the characteristics of the typical mobile voice communication channels. Along with the WPA speech corpus, the WPA authors provided a pre-built HTK ASR HMM model from WPA speech data to make it easy for us to segment the speech data automatically to the phoneme level. Using the down sampled data, the following two steps shown in Figure 2 are applied to build our bootstrap MSA ASR:

- The segmental K-means and maximum likelihood HMM training algorithms were used for WPA 8 KHz model bootstrap, followed by multiple iterations of Baum- Welch model parameter re-estimation to refine the model until it converged to the best performance by testing it on WPA test set, which produced the WPA 8 KHz ASR system WP_8K_0.

- The WPA to SAMPA phone mapping was applied to WP_8K_0 to create WP_8K_1, which uses SAMPA phoneme set.

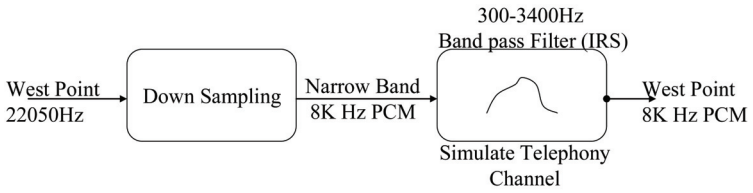


Fig. 2. West Point speech down sampling

4.2.3 SAMPA to WPA phone set mapping

After analyzing of the WPA database, we concluded that the WPA speech data was sufficient for the creation of all the MSA allophone models that can be mapped to the target OrienTel MSA speech recognition using SAMPA phoneme set. Table 5 provides a mapping between them. Since two SAMPA Arabic phones "l'" and "u:" are missing in the WPA system, we used two close phones "l" and "u" to clone them for bootstrap. Then the phone models are rebuilt on the whole Orientel Tunisia MSA training set. Two foreign language SAMPA phones "p" and "V" are not presented in either WPA data or in OrienTel Tunisian database. We fill this gap by borrowing the phoneme models from an English ASR system. Besides, three vowels "aw", "ay", and "ey" appeared in WPA are not used in SAMPA.

4.2.4 Tunisian MSA speech recognition system from WPA recognition system

Once we built the WP_8K_1 ASR system, we experiment both model re-estimation and global MLLR model adaptation to produce MSA ASR systems using OrienTel Tunisian training data. The four MSA speech recognition systems built are:

- WP_8K_1: The baseline telephone SI-ASR system by WPA down sampling.
- BW_6: The retrained on Tunisian MSA training set, with 6 iterations of Baum- Welch parameter re-estimation, bootstrapped by WP_8K_1 model.
- MLLR_1: MLLR model mean adaptation on WP_8K_1, using Tunisian MSA training set.
- MAP_1: MAP adaptation on MLLR_1 on both mean and variances, using Tunisian MSA training set.

Since we are using supervised training in our experiment, we only select a subset of speech recording with good labelling. We also excluded incomplete recorded speech and other exceptions. The total number of retraining and adaptation utterances used in our experiment is 7554.

We test the MSA systems described above by using selected classes of Tunisian MSA data. The test result is shown in Table 6. All the tests are on complete OrienTel Tunisian test set with some excluded data classes described above. We use one pass Viterbi search algorithm with bean control to reduce search space to improve recognition speed. For language model, we use a manual written context free grammar to cover all the content classes described in Table 3, exclude class A, W, S, X not included in test. We also excluded imcomplete recorded and improper labeled speech in Tunisian data collection test set. The total number of utterances we used in test is 1899.

From Table 6, we observed the following

- Acoustic and speaker mismatch can cause significant SI-ASR performance degradation when we use WPA down sampled system (WP_8K_1) to do test on OrienTel Tunisian MSA test set.
- Re-trained OrienTel Tunisian achieves the best performance at combined error rate reduction of 67%, compare to the original system WP_8K_1.
- A simple MLLR global adaptation on miss matched models can achieve 50% error reduction combined, given there are sufficient adaptation data.
- Further MAP adaptation achieves another 2.1% error reduction from MLLR_1 system. It is obvious that MLLR can achieve such good result is due to the main resource of errors from the initial WP_8K_1 is the acoustic channel miss match.

We observed poor results on yes/no (class Q) recognitions. And MAP even make it worse, oppose to overall improvement compare to other classes. From our initial analysis, this is due to pronunciation variations and poorly formed a prior density estimation described in section 2.

Tunisian	MSA	ASR	Systems	
Contents Code	WP_8K_1	BW_6	MLLR_1	MAP_1
I,B	31.25	7.18	9.72	10.19
C	34.13	6.34	14.58	10.84
N	7.33	2.81	3.12	3.64
M	15.42	4.48	7.71	9.95
Q	92.05	37.78	35.56	70.00
D	15.63	9.09	10.39	11.31
T	23.32	11.66	14.13	15.55
E	25.38	12.69	16.24	9.64
O	59.81	25.69	43.58	43.06
L	45.17	18.40	28.35	26.81
Total Error rate	28.01	9.27	14.24	13.94

Table 6. Word error rates (%) on Tunisian MSA ASR systems, bootstrap from WPA ASR

4.2.5 Tunisian to Jordan dialect adaptation

In this section, we experiment adaptation from Tunisian dialect to Jordan dialect. To establish a baseline for comparison, we use the best model built on Tunisian training data, the six iteration of embedded Baum-Welsh trained model BM_6 described in previous section. We name it T1_BW_6 in this section.

Tunisian	MSA	ASR	Systems
ASR Systems	T1_BW_6	G_MLLR	MAP
Error rate	16.85	15.65	16.75

Table 7. Word error rates (%) on Jordan MSA ASR systems, adapted from Tunisian ASR

From Orientel Jordan data collection, we selected 1275 utterances from its test set which constitute about 80 speakers. Table 7 listed our testing results. From column 1, we observed that dialect mismatch degraded ASR performance. Instead of retrain the ASR system, we

randomly selected 600 utterances from Jordan training set to adapt a Tunisian ASR system to Jordan ASR system. This is about 2.5% of total Jordan training set of 23,289 utterances. Using global MLLR adaptation, we saw ASR accuracy improves by 7%. Furthermore, we tested MAP adaptation using the same adaptation set on T1_BW_6 model; we saw slightly accuracy improvement of less than 1%. We believe that for MAP adaptation, more data are needed since it adapts more parameters than produce a global transformation in MLLR.
In this experiment, we used the same recognition algorithm and slightly modified lexicon and context-free grammar as the last section experiment. The lexicon change is based on pronunciation we observed between Tunis and Jordan data collection. Using a PC with 2.4 Ghz Intel processor (Core 2 quad core, but only one is used since our software only use one thread), the 600 utterances adaptation only takes less than 10 minutes.

5. Conclusion

In this chapter, we studied several approaches in building Arabic speaker independent speech recognition for real-world communication service applications. In order to find out practical and efficient methods to build search a system using limited data resource, we study both traditional acoustic model re-estimations algorithms and adaptation methods, which require much less data to improve SI-ASR performance from an existing SI-ASR system with dialect mismatch. Also adaptation methods are more practical to implement as online system to improve SI-ASR at runtime, without restart the system. This is an important feature required by communication service applications, since we need high availability and a little room for down time.
In this work, we only study acoustic model re-estimation and adaptation aspects to improve SI-ASR in mismatched dialect environment. We also observed that there are significant pronunciation variations in different Arabic dialects that need lexicon changes to improve SI-ASR performance. We made lexicon modification when we experiment Tunisia to Jordan dialect adaptation as described above. Also we realize that there are language model variations between different dialects as well.

6. Acknowledgements

The authors wish to thank Col. Stephen A. LaRocca, Mr. Rajaa Chouairi, and Mr. John J. Morgan for their help and fruitful discussion on WPA database for Arabic speech recognition and provide us a bootstrap HMM used in our research.
This work is partially funded by European Commission, under Orientel as an R&D project under the 5th Framework Programme (Contract IST-2000-28373).

7. References

Afify, M., Sarikaya, R Kuo, ., H-K. J., Besacier, L., and Gao Y-Q. (2006). On the use of morphological analysis for dialectal Arabic speech recognition, ICSLP 2006.

Baum, L. E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains, Ann Math Stat. 37, 1554-1563.

Baum, L. E. and Eagon, J. A. (1967). An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology, Bull. Amer. Math. Soc. 73, 360-363.

Baum, L. E., Petrie, T., Soules, G. and Weiss N. (1970). A Maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, Ann. Math. Statist. Volume 41, Number 1, 164-171.

Billa, J., Noamany, M., Srivastava, A., Makhoul, J., and Kubala, F. (2002). Arabic speech and text in TIDES OnTAP, Proceedings of HLT 2002, 2nd International Conference on Human Language Technology Research, San Francisco.

Chou, W., Lee, C.-H., Juang, B.-H., and Soong, F. K. (1994) A minimum error rate pattern recognition approach to speech recognition, Journal of Pattern Recognition, Vol. 8, No. 1, pp. 5-31.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, 1977.

Diehl, F., Gales, M.J.F. Tomalin, , M., and Woodland, P.C. (2008). PHONETIC PRONUNCIATIONS FOR ARABIC SPEECH-TO-TEXT SYSTEMS, ICASSP 2008.

Ephraim, Y.,and Rabiner, L. R. (1990) On the relations between modeling approaches for speech recognition, IEEE Trans. on Information Theory 36(2): 372-380.

Hartigan, J. A. and Wong, M. A. (1979). A K-Means Clustering Algorithm, Applied Statistics 28 (1): 100−108.

Huang, X.-D., Acero, A., and Hon, H.-W. (2001). Spoken Language Processing: A Guide to Theory, Algorithm, and System Development, Prentice Hall.

Gauvain, L. and Lee, C.-H. (1994). Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, IEEE Trans. on Speech and Audio Proc., Vol. 2, No. 2, pp. 291-298.

Kirchhoff, K. Vergyri, D. (2004). Cross-dialectal acoustic data sharing for Arabic speech recognition, ICASSP 04.

Kirchhof, K., et. al. (2003). NOVEL APPROACHES TO ARABIC SPEECH RECOGNITION: REPORT FROM THE 2002 JOHNS-HOPKINS SUMMER WORKSHOP, ICASSP 2003.

Jelinek, F. and Mercer, R.L. (1980) Interpolated estimation of Markov source parameters from sparse data, in Pattern Recognition and Practice, Gelsma, E.S. and Kanal, L.N. Eds. Amsterdam, The Netherlands: North Holland, 381-397.

Juang, B.− H., (1985). Maximum likelihood estimation for mixture multivariate observations of Markov chains, AT&T Technical Journal.

Juang, B.− H., Chou, W. and Lee, C.-H. (1996). Statistical and Discriminative Methods for Speech Recognition, in *Automatic Speech and Speaker Recognition: Advanced Topics*, Kluwer Academic Publishers, pp 83-108.

Lee, C.-H. and Gauvain, J. L. (1996). Bayesian adaptive learning and MAP estimation of HMM, in *Automatic Speech and Speaker Recognition: Advanced Topics*, Kluwer Academic Publishers, pp 83-108.

Lee, C.-H. and Huo Q. (2000). On Adaptive decision rules and decision parameter adaptation for automatic speech recognition, Proceedings of the IEEE, Vol. 88, No. 8, pp. 1241-1269.

Lee, C.-H., Gauvain, J.-L., Pieraccini, R., and Rabiner, L. R. (1993) Large vocabulary speech recognition using subword units, Speech Communication, Vol. 13, Nos. 3-4, pp. 263-280.

LaRocca, S. A., Chouairi, R. (2002). West point Arabic speech corpus, LDC2002S02, Linguistic Data Consortium, http://www.ldc.upenn.edu.

Leggetter C. J. and Woodland P. C. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models, Computer Speech & Language, vol. 9, no. 2, 171-185 (15).

MacQueen, J. B., (1967). Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297

Normandin, Y. (1996). Maximum Mutual Information Estimation of Hidden Markov Models, in *Automatic Speech and Speaker Recognition: Advanced Topics*, Kluwer Academic Publishers, pp 57-82.

OrienTel, (2001) Multilingual access to interactive communication services for the Mediterranean & the Middle East, http://www.speechdat.org/ORIENTEL/index.html

Rabiner, L. R (1989). A Tutorial on hidden Markov models and selected applications in speech recognition, PROCEEDINGS OF THE IEEE, VOL. 77, NO. 2.

Rabiner, L. R., Juang, B.-H. (1993). Fundamental of Speech Recognition, Prentice Hall, Englewood Cliffs.

Rabiner, L. R., Juang, B.-H., and Lee, C.-H. (1996) An overview of automatic speech Recognition, in *Automatic Speech and Speaker Recognition: Advanced Topics*, Kluwer Academic Publishers, pp 1-30.

Rabiner, L. R., Wilpon, J. G., Juang, B.-H. (1986). A Segmental k-means training procedure for connected word recognition, AT&T technical journal 65:33, 21-31.

Rahim, M and Juang, B.-H., (1994) Signal bias removal for robust speech recognition, Proceedings ICASSP-94, Adelaide, Australia.

Rambow, O., et. al. (2006). Parsing Arabic dialects, final report version 1, Johns Hopkins Summer Workshop 2005.

SAMPA, (2005). Speech Assessment Methods Phonetic Alphabet): Computer readable phonetic alphabet, http://www.phon.ucl.ac.uk/home/sampa/home.htm

Schultz, T., and Black, A. W. (2006). Challenges with rapid adaptation of speech translation systems to new language pairs. ICASSP 2006.

Siemund, R. et. al. (2002). OrienTel − Arabic speech resource for the IT Market, LREC 2002 Arabic Workshop.

Wang, Z., Schultz, T., and Waibel, A. (2003). COMPARISON OF ACOUSTIC MODEL ADAPTATION TECHNIQUES ON NON-NATIVE SPEECH, ICASSP 2003.

Young, S. et. al. (2006). The HTK Book, http://htk.eng.cam.ac.uk.

Yu, K. , Gales, M.J.F., and Woodland, P.C. (2008). UNSUPERVISED DISCRIMINATIVE ADAPTATION USING DISCRIMINATIVE MAPPING TRANSFORMS, ICASSP 2008.

Zitouni, I. et. al. (2002). OrienTel: speech-based interactive communication applications for the mediterranean and the Middle East, ICSLP 2002.

26

Ultimate Trends in Integrated Systems to Enhance Automatic Speech Recognition Performance

C. Durán
University of Pamplona
Colombia

1. Introduction

An automatic speech recognition (ASR) system can be defined as a mechanism capable of decoding the signal produced in the vocal and nasal tracts of a human speaker into the sequence of linguistic units contained in the message that the speaker wants to communicate (Peinado & Segura, 2006**)**. The final goal of ASR is the man–machine communication. This natural way of interaction has found many applications because of the fast development of different hardware and software technologies. The most relevant are the access to information systems; an aid to the handicapped, automatic translation or oral system control. ASR technology has made enormous advances in the last 20 years, and now large vocabulary systems can be produced that have sufficient performance to be usefully employed in a variety of tasks (Benzeghiba et al., 2007; Coy & Barker, 2007; Wald, 2006; Leitch & Bain, 2000). However, the technology is surprisingly brittle and, in particular, does not exhibit the robustness to environmental noise that is characteristic of humans. Speech recognition applications that have emerged over the last few years include voice dialing (e.g., "Call home"), call routing (e.g., "I would like to make a collect call"), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g., a radiology report), domotic appliances control (e.g., "Turn on Lights" or "Turn off lights"), content-based spoken audio search (e.g., find a podcast where particular words were spoken), isolated words with a pattern recognition, etc. With the advances in VLSI technology, and high performance compilers, it has become possible to incorporate different algorithms into hardware. In the last few years, various systems have been developed to serve a variety of applications. There are many solutions which offer small-sized, high performance systems; however, these suffer from low flexibility and longer cycle-designed times. A complete software-based solution is attractive for a desktop application, but fails to provide an embedded portable and integrated solution.

Nowadays, High-end Digital Signal Processors (DSP's) from companies, such as; Texas Instruments (TI) or Analog Devices and High-performance systems like Field Programmable Gate Array (FPGA) from companies, such as; Xilinx or Altera, that provide an ideal platform for developing and testing algorithms in hardware.

The Digital signal processor (DSP) is one of the most popular embedded systems in which computational intensive algorithms can be applied. It provides good development flexibility

and requires a relatively short application development cycle; therefore, the Automatic Speech Recognition on DSP Technology will continue as an active area of research for many years.

Speech recognition is a related process that attempts to identify the person speaking, as opposed to what is being said. The general-purpose speech recognition systems are generally based on Hidden Markov Models (HMM's). This is a statistical model which outputs a sequence of symbols or quantities. One possible reason why these models are used in speech recognition is that a speech signal could be viewed as a piece-wise stationary or short-time signal (Rabiner, 1989; Jou et al., 2001). Another of the most powerful speech analyses techniques is Linear Predictive Coding (LPC). The covariance analysis of linear predictive coding has wide applications, especially in speech recognition and speech signal processing (Schroeder & Atal, 1985; Kwong & Man, 1995; Tang et al., 1994). Real-time applications demand very high-speed processing, for example, for linear predictive coding analysis. An approach to acoustic modeling is the use of Artificial Neural Networks (ANN) (Lim et al., 2000). They are capable of solving much more complicated recognition tasks, but do not scale as well as LPC or HMM's when it comes to amplified vocabulary. Rather than being used in general purpose speech recognition applications, they can handle low quality, noisy data and speaker independence. Such systems can achieve greater accuracy like LPC or HMM's based systems, as long as there is training data and the vocabulary is limited. A more general approach using neural networks is phoneme recognition. This is an active field of research, but generally the results have been better than for LPC or HMM's. There are also LPC-ANN and HMM-ANN hybrid systems that use the neural network.

This chapter provides an overview of some applications with integrated systems, in order to improve the performance of the ASR systems. In the last section of this chapter the use of LPC-ANN hybrid system as an alternative in the identification of speech command and the implementation using Matlab® Released Software is described coupled with the DSP Hardware (DSK6416T Started Kit) developed by Texas Instruments.

2. VLSI technology

To learn the concept of integrated or embedded systems and their importance for ASR it is necessary to explain what Very Large Scale Integration (VLSI) technology is.

VLSI is the process of creating integrated circuits by combining thousands of transistor-based circuits into a single chip (S. Kung, 1985; Barazesh et al., 1988; Cheng et al, 1991). VLSI began in the 70's when complex semiconductors were being developed. The processor is a VLSI device; however, the term is no longer as common as it once was since chips have increased into a complexity of millions of transistors. Today, in 2008, billions of transistor processors are commercially available; an example of which is the dual core processor called Montecito Itanium. This is expected to become more commonplace as a semiconductor.

The ASR has been an active area of research for many years; for this reason, with the advances in VLSI technology and high performance compilers, it has become possible to incorporate algorithms in hardware with great improvements in performance. In the last few years, various systems have been developed to cater to a variety of applications (Phadke et al., 2004; Melnikoff et al., 2002). Figure 1 shows an example of VLSI technology which possesses properties of low-cost, high-speed and massive computing capability; therefore, it is a suitable candidate in integrated Systems (e.g. The DSP) to enhance Automatic Speech Recognition Performance**.** Due to the fast progress of VLSI, algorithm-oriented architectural

array appears to be effective, feasible, and economical. Many algorithms can be efficiently implemented by array processors.

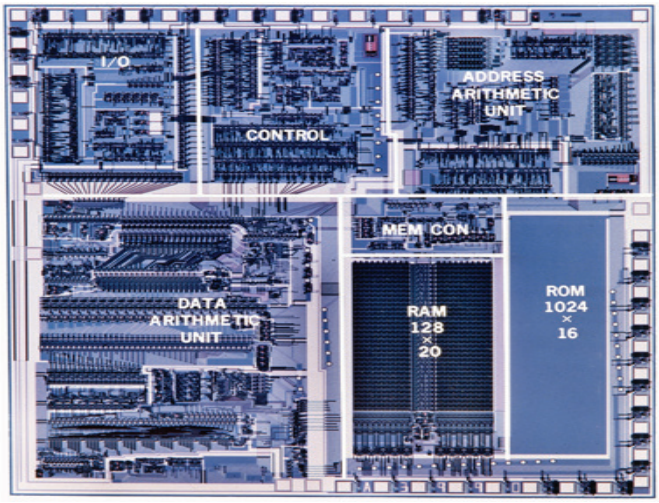


Fig. 1. DSP chip from VLSI (Bell Labs, device layout)

2.1 DSP chip and typical applications

This section describes the concepts of the DSP's device which can be applied to ASR. Digital signal processing chips (DSP's) were introduced in the early 80's and have caused a revolution in product design. Current major DSP manufacturers include Texas Instruments (TI), Motorola, Lucent/Agere, Analog Devices, NEC, SGS-Thomson, and Conexant (formerly Rockwell Semiconductor).

DSP's are specifically designed to rapidly perform the sum of products' operation required in many discrete-time signal processing algorithms. They contain hardware parallel multipliers and functions implemented by microcoding in ordinary microprocessors that are implemented by high-speed hardware in DSP's. Since they do not have to perform any of the functions of a high end microprocessor, like an Intel Pentium Dual-Core, a DSP can be streamlined to have a smaller size, use less power, and have a lower cost (Tretter, 2008). The velocity of the latest generations of DSP's have increased to the point where they are being used in high speed applications like DSL, wireless base stations and hand sets. Other applications include the implementation of video coding motion (Louro et al., 2003), Real-time audio transmission (Pasero & Montuori, 2003), e-commerce security (Jiankun et al., 2001), digital signal Processing system design (Kehtarnavaz et al., 2004) and implementation of facial recognition algorithms (Batur et al., 2003). Some of the advantages resulting are the integrated digital circuits that are very reliable and can be automatically inserted in boards easily.

DSP's can implement complicated linear and nonlinear algorithms and easily switch functions by jumping to different sections of the program code, such as; the implementation of an Artificial Neural Network, or on the "On-Chip DSP" for the ASR. The complexity of the algorithms is only limited by the imagination of the programmer and the processing speed of the DSP. For the implementation of a DSP in speech recognition applications, it is

essential to define the following parameters, such as; DSP technology, size of the data set, number of samples, computing speed and pattern recognition methods. Analog circuits are designed to perform specific functions and lack the flexibility of the programmable DSP approach. Another advantage is that small changes in the DSP function can be made by varying a few lines of code in a ROM or EPROM, while similar changes may be very difficult with a hard-wired analog circuit. Digital signal processing algorithms were used long before the advent of DSP chips.

DSP's have continually evolved since they were first introduced as VLSI improved technology since users requested additional functionality and as competition arose. Additional functions have been incorporated like hardware bit-reversed addressing for Fast Fourier Transform (FFT) and Artificial Neural Networks, hardware circular buffer addressing, serial ports, timers, Direct-Memory-Access (DMA) controllers, and sophisticated interrupt systems including shadow registers for low overhead context switching. Analog Devices has included switched capacitor filters and sigma-delta A/D and D/A converters on some DSP chips. Instruction rates have increased dramatically; the state-of-the-art DSP's, like the TMS320C5000 series are available with devices that can operate at clock rates of 200 MHz. Figure 2 presents a photograph of the first TMS 320 programmable DSP from Texas Instruments.

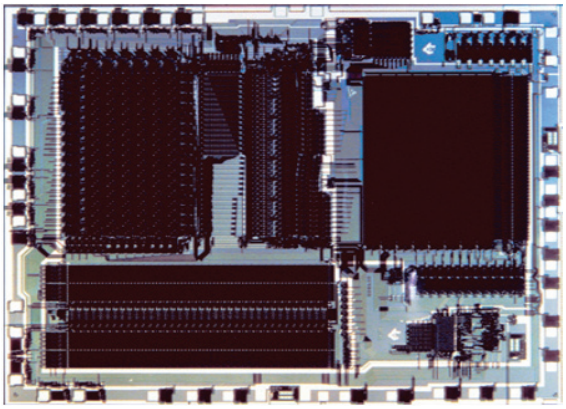


Fig. 2. First TMS 320 programmable DSP device, Texas Instruments, Inc.

The TMS320C6000 family has Very Long Instruction Word (VLIW) architecture, which has devices with clock rates up to 1 GHz (e.g. DSP TMS320C6416T of TI) the speed increase is largely a result of reduced geometries and improved CMOS technology.

In the last years, DSP manufacturers have been developing chips with multiple DSP cores and shared memory for use in high-end commercial applications like network access servers handling many voice and data channels. DSP chips with special purpose accelerators like Viterbi decoders, turbo code decoders, multimedia functions, and encryption/decryption functions are appearing. The rapid emergence of broadband wireless applications is pushing DSP manufacturers to rapidly increase DSP speeds and capabilities so they do not have a disadvantage with respect to FPGA's.

In 1988, TI shipped initial samples of the TMS320C30 to begin its first generation TMS320C3x DSP family. This processor has a 32-bit word length. The TMS320C30 family has a device that can run at 25 million instructions per second (MIPs). The TMS320C31 has a

component that can perform 40 Million Instructions per second (MIPs). TI started its second generation DSP family with the TMS320C40 which contains extensive support for parallel processing. The last years ago TI introduced the TMS320C67x series of floating point and the TMS320C64x series of Fixed point DSP's, such as the TMS320C6713 and the TMS320C6416T. They were implemented on large main-frame computers and, later, on expensive "high-speed" mini-computers. DSP's of Floating-point and Fixed point are used in different applications because of their ease in programming. Some reasons for their applications are because they are smaller, cheaper, faster, and use less power. However, this is not much of a problem in speech recognition applications where the Levels of voice signal can be processed with any type of DSP.
Knowing that DSP's work well in the voice processing, these are used in a wide variety of offline and real-time applications. They are described in the following table:

Application	Description
Telecommunications	Telephone line modems, FAX, cellular telephones, speaker phones, ADPCM transcoders, digital speech interpolation, broadband wireless systems, and answering machines
Voice/Speech	Speech digitization and compression, voice mail, speaker verification, and automatic speech recognition (ASR).
Automotive	Engine control, antilock brakes, active suspension, airbag control, and system diagnosis.
Control Systems	Head positioning servo systems in disk drives, laser printer control, robot control, engine and motor control, and numerical control of automatic machine tools.
Military	Radar and sonar signal processing, navigation systems, missile guidance, HF radio frequency modems, secure spread spectrum radios, and secure voice
Medical	Hearing aids, MRI imaging, ultrasound imaging, and patient monitoring.
Instrumentation	Spectrum analysis, transient analysis, signal generators.
Image Processing	HDTV, pattern recognition, image enhancement, image compression and transmission, 3-D rotation, and animation

Table 1. Typical applications with TMS320C6000 family

Table 2 shows the following three categories of Texas Instruments DSP's. As can be seen for sound processing applications and speech recognition, it would be best to use a 32-bit DSP.

DSP's Categories	Characteristic	Applications
TMS320 C1s, C2x, C24x	Low Cost, Fixed-Point, 16-Bit Word Length	Motor control, disk head positioning, control
TMS320 C5x, C54x, C55x	Power Efficient, Fixed-Point, 16 Bit Words	Wireless phones, modems
TMS320 C62x (16-bit fixed-point) C3x, C4x, C64x, C67x (32–bit)	High Performance DSP's (32-bit floating-point and Fixed-Point)	Communications infrastructure, xDSL, imaging, sound, video

Table 2. Categories of Texas Instruments DSP's

3. ASR applications with integrated systems

ASR with integrated systems are employed in applications, such as; to enhance education for all students (Wald, 2005), methods to learn proper name pronunciations from audio samples (Beaufays et al., 2003), speech translation (Paulik et al., 2005) and classification techniques of speech parameters (Acevedo & Nieves, 2007). Many important results have been obtained through robust automatic speech recognition in car noise environments (Ding et al., 2006) and Statistical voice activity detection (Ramirez et al, 2005), while other applications employ powerful computers to handle complex recognition algorithms. There is clearly a demand for an effective solution on integrated systems like portable communication and various low-cost consumer electronic devices. Digital signal processor (DSP) is one of the most popular embedded platforms on which computationally intensive algorithms can be realized (Yuan et al., 2006).
This section shows a brief description of integrated systems and shows some important results using DSP technology, for example, to identify and classify voice commands or isolated words.

3.1 Isolated word recognition

The automatic speech recognition with integrated systems has a great performance for solving certain problems and limitations in human health service.
Nowadays, one of the problems that affects certain individuals is the lack of acoustic feedback (Deaf Speakers); thereby hindering their ability to communicate effectively (Kota et al., 1993). There is a practical need for the development of devices that can perform recognition of deaf speech in real time (The integrated systems can be a possible solution). Such devices could serve the communication needs of deaf speakers by correctly and reliably recognizing their speech and converting it into printed displays/synthetic speech and used as voice input for a communication system. However, limitations on previously developed automatic speech recognition systems include limited vocabulary sets, intolerance to even slight variations in speech, and inability to operate in real time. Previous researchers have used a combination of dynamic time warping, template matching and HMM for recognition of deaf speech (Abdelhamied et al., 1990; Deller et al., 1988).
Reported recognition rates for isolated word recognition tasks have been in the range of 20% to 99.2%, and are highly dependent on the vocabulary, the extent of hearing loss of the speaker(s) and the performance of the recognition system itself. It is possible to select consistent acoustic features in deaf speech.
Artificial neural networks have been shown to perform pattern recognition, handle incomplete data and variability very well. It would seem appropriate that their use could enhance the performance of deaf speech recognizers by providing hybrid approaches of conventional signal processing and neural systems. In particular, the salient features of neural networks make them a useful tool in building better recognition systems for deaf speech. In this application three hundred utterances were recorded for each subject (age group 20-60 years old) in one session. Speech records were obtained from 6 deaf subjects. Speech intelligibility ratings were then obtained for each deaf speaker. One word list was selected from repetitions and randomized to avoid learning biases.
The entire ASR system including the preprocessor, data set, feature extractor and the neural network are implemented on the Texas Instruments' processor (TI) TMS320C30 DSP EVM (See Figure 3). The software for pre-processing and recognition tasks was written in SPOX

MI and C languages for maintaining modularity and portability. A time delay neural network model built for isolated digital recognition of normal speech was modified to incorporate the additional feature inputs to the network (Castro & Casacuberta, 1991). Time delays were built into the network structure to evolve time invariant feature extractors and feature integrators. The network was trained using general purpose backpropagation control strategy. A preliminary set of experiments were conducted with a vocabulary size of 20 words and dedicated networks for 2 speakers who had the highest and the lowest intelligibility in the speaker set. Six separate training schedules were developed each with a varying number of training tokens in the range 5-3. Each separate network was then tested for recognition rates with testing parameters.

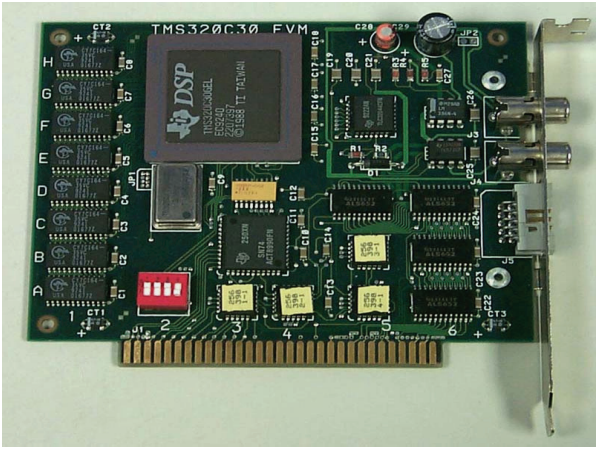


Fig. 3. Texas Instruments (TI) TMS320C30GEL Digital Signal Processor (DSP)

In this application the best results were obtained with the average intelligibility ratings of the two deaf speakers that were 75 % and 38 %. The neural network learned specific features of both speakers very well. A perfect recognition - l00% - of the training set was obtained within 4-5 training sessions (epochs).

3.2 Voice pattern recognition for statistical methods

This application describes the implementation of a pattern recognition algorithm in a Blackfin DSP ADSPBF533 EZ-KIT of Analog Devices, such as; figure 4 presents, which can identify voice patterns from speakers in real time using statistical methods. This system responds only to voice commands based on a statistical analysis of a spectrogram generated by the voice command (Gómez, 2007). The goal of the generation of the spectrogram is characterized by a group of objects with measurements or qualities, where the values tend to be similar. For objects in the same class the differences are minimal and therefore the characteristics are unchanged and irrelevant to those changes in the data provided.

To extract the characteristics of a sound signal it is important not to take it all as a single pattern, since each segment of sound in the time domain has a different parameter and the sum of these is necessary to give the difference at the moment of applying the pattern recognition method. The signal is segmented in parts and each division generates an FFT, where the matrix is gotten with values of volume, frequency and the image obtained in three dimensions which is called spectrogram-time or frequency sonogram (Proakis &

Manolakis, 1998). The features extraction for the reflection spectrum is applied at the moment a voice command is acquired or any word with three different magnitudes of time, level and frequency. The three magnitudes generated by the speaker are affected depending on their moods. The speaker pronounces the command and then the spectrum changes in spite of being the same command.

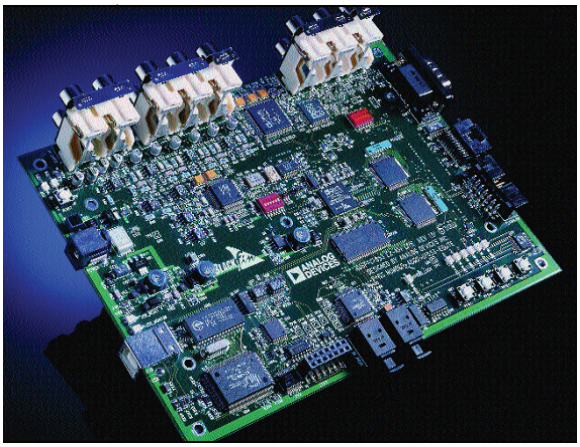


Fig. 4. ADSP-BF533 EZ-KIT Lite®, Analog Devices (ADI) evaluation system for Blackfin® embedded media processors.

Figure 5 shows the graphic of the spectrum reflecting the components in frequency and level. There are points in the categorization of the voice signal, where 9 points were extracted in the frequencies of 300Hz, 500Hz, 800Hz, 1.000Hz, 2.500Hz, 3.600Hz, 5.000Hz 7.000Hz and 8.000Hz. The magnitudes of these frequencies were carried to a vector, creating a database for each command voice.

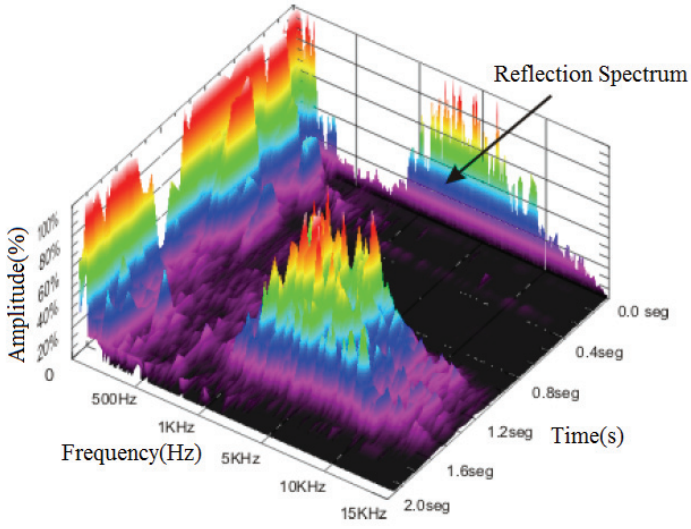


Fig.5. Spectrogram of voice command "Forward" showing the reflective spectrum.

After obtaining a characteristic vector from the dataset a linear regression is generated as a pattern recognition method. The linear regression analysis is a statistical technique for modeling of two or more variables. The regression can be used to relate a signal with another (Montgomery & Runger, 2006). The linear regression between two voice commands give as the results, a new vector of mean values or Mean Square Errors (MSE), and comparing two similar voice commands as the MSE goes to 0; see equation 1.
The residues of the MSE are determinate by:

$$MSE = \left(\frac{1}{n}\right)\sum_{i=1}^{n}\left(y_i - \hat{L}_i\right)^2 \tag{1}$$

Where, $\boldsymbol{L}$ is the regression estimated between two commands, $\boldsymbol{y}$ is the vector generated in the database with the voice commands and $\boldsymbol{n}$ is the length of the vector.
For the implementation of pattern recognition algorithm the development Kit ADSPBF533 EZ-KIT Lite of Analog Devices was utilized, which is composed of several peripherals, such as; ADC (Analog Digital Converters), DAC (Digital Analog Converters), RAM, EEPROM memory, etc. For the start-up of the device it is important to configure all ports and peripherals through the following functions: *i*) Configuration of general peripherals: This stage is composed of memory banks, the filter coefficients and the Fourier transform. *ii*) Configuration of the audio codec: The kit DSP includes an audio codec AD1836 designed for applications of high-fidelity audio; they are used in audio formats to 16 bits at a rate of frequency of 44.100Hz.
With the DSP system tests were made to determine their behavior and their effectiveness. The environmental conditions refer to external noise, location of the speaker regarding the microphone, types of microphones, parameters, compression and volume. Each voice command was assigned a binary code. The success rate of classification of each command was obtained with 50 words (10 each one) to verify coherence between the words pronounced and recognized (see table 3). The DSP acquires the voice command; the signals are filtered and the FFT is generating and calculating the matrix spectrogram, where the time, linear regression and MSE are calculated in milliseconds. The pattern recognition method with the spectrogram reflection technique applied on a DSP, delays 0.1 seconds in processing the information while a personal computer delays 2.5 seconds.

Command	Binary Code	Success rate (%)
Forward	000001	100
Stop	000010	95
Back	000100	100
Left	001000	98
Right	010000	100

Table 3. Success rate of the classification with the DSP Blackfin, to identify different voice commands with a single speaker.

4. Experimental framework

The following section describes an experimental setup where a LPC-ANN hybrid system was used as an alternative in the identification of voice commands from a speaker, and the implementation using Matlab Released 7.1 Software coupled with the DSP Hardware (i.e. The DSK6416T) developed by Texas Instruments.

Figure 6 shows the Automatic speech recognition; it is constituted by two principal phases. The first phase is the training stage where each word or voice signal is acquired with the purpose to obtain a descriptive model from all the words used to build the model and train the network. As can be seen, in the recognition phase a new voice sample is acquired and is then projected onto the model to identify and classify the voice signal using the already trained network. The signal acquisition is obtained with the help of a high gain microphone and then the time is digitized by means of a computer audio card; in this process the voice signals obtained through the feature extraction techniques. With the feature extraction the spectral measurements become a set of parameters that describes the acoustic properties of phonetic units. These parameters can be: Cepstral coefficients, or the energy of the signal (i.e. extracting the energy from LPC), etc. Once the basic parameters are obtained, the aim is to identify the voice signal, applying the methods and algorithms that are translated into numerical values. For this, Neural Networks are used, such as; the Backpropagation or multilayer neural network specifically. Backpropagation was created by generalizing learning rules to multiple-layer networks and nonlinear transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can be approximate as a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as well as can be defined. A Backpropagation consists of three types of layers, namely: the input layer, a number of hidden layers; and the output layer. Only the units in the hidden and output layers are neurons and so it has two layers of weights (Cong et al., 2000; Kostepen & Kumar, 2000). In this experiment a Multilayer Perceptron (MLP) neural network is used, which has a supervised learning phase and employs a set of training vectors, followed by the prediction or recall of unknown input vectors.

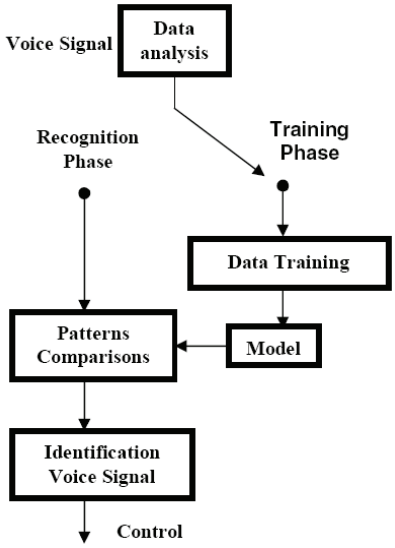


Fig.6. Block Diagram of Automatic Speech Recognition (ASR)

4.1 Data acquisition

Each one of the stages of the Automatic Speech Recognition is developed with the help of Matlab 7.1. The data acquisition control and signal processing are done with an audio digital

card, LPC and Neural Networks which are accomplished by a written-in-house software through of a Graphic User Interface (GUI). This software allows the voice signal to be acquired quickly in real-time. The software allows a model to obtain the training data under platform of Matlab-Simulink and the implementation on the DSP Hardware. To acquire the signal from the auxiliary input of the computer audio card, the function, "wavrecord" is used which corresponds to the acquisition time (i.e .in seconds), the sampling frequency (Fs) in Hz (e.g. 8000, 11025, 22050 and 44100) and the channel is obtained (i.e. Mono is Ch_1 and Ch_2 is Stereo). For example, to acquire a signal in mono-stereo with a period of one second of duration and the sampling frequency of 8000 Hz, it is possible to use the following command from the Workspace of Matlab:

```
>>Fs=8000
>>Y=wavrecord (1*Fs, Fs, 1)
```

To keep a signal in audio format (e.g. wav) the function "wavwrite" is used, where wavwrite (i.e. Y,Fs,NBITS,WAVEFILE) writes the data "Y" to a Windows file specified by the file name "WAVEFILE", with a sample rate of "Fs" in Hz and with "NBITS" number of bits. NBITS must be 8, 16, 24, or 32. Stereo data should be specified as a matrix with two columns. For NBITS < 32, amplitude values outside the range [-1, +1] are clipped. For example, in order to keep the previous sound, the following command will be used:

```
>> wavwrite (Y, Fs, 16,' close.wav')
```

Figure 7 shows the typical signal of the 'close' command, at the moment to acquire the voice signal from the auxiliary input. A total of 12,000 samples was obtained for the first 30 measurements and were processed later.

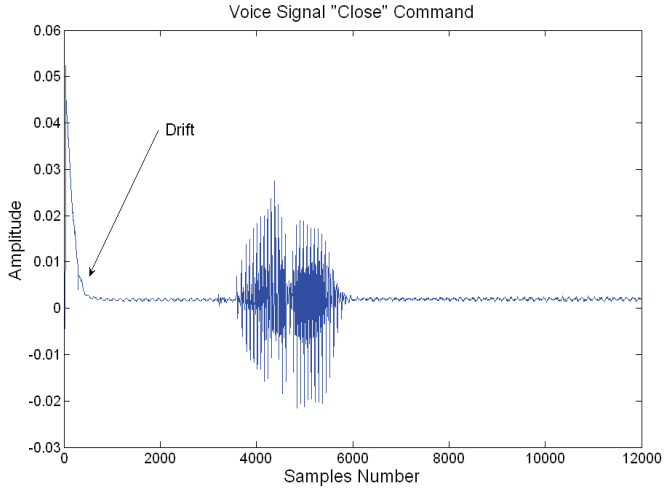


Fig. 7. Voice Signal of the "Close" Command

4.2 Signal processing

The next stage consists of acquiring the energy of the signals from LPC, with the possibility to acquire the principle parameters for network training. The signals processed were made

with algorithms of Matlab. The first samples of the signals acquired were reduced due to the drift that was generating from the audio board; therefore, the "baseline" of the signal was acquired (see figure 8).

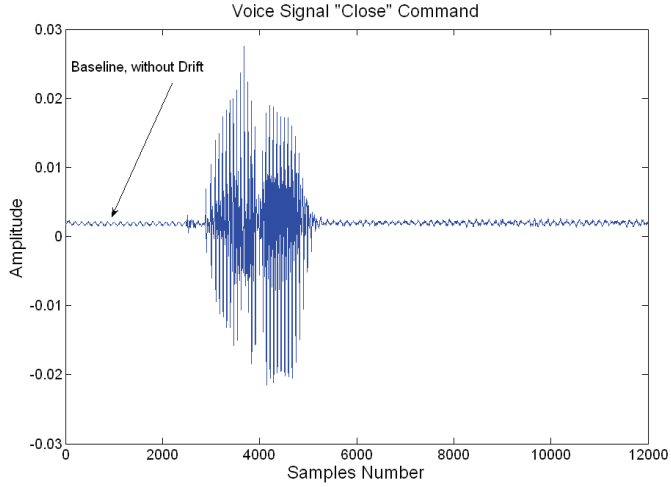


Fig. 8. Voice Signal without "drift" and with "baseline"

The figure below illustrates the acquisition of the first samples which do not contribute important information to the recognition process; therefore, this part was eliminated. In the second step the signal is pre-processed (i.e. Normalized) for 2,000 samples. The figure illustrates this process; where it represents the reduced graph with regard to the samples. This method was used with the aim so that the signal is always located at the same point and the acquisition of measurements must be repetitive.

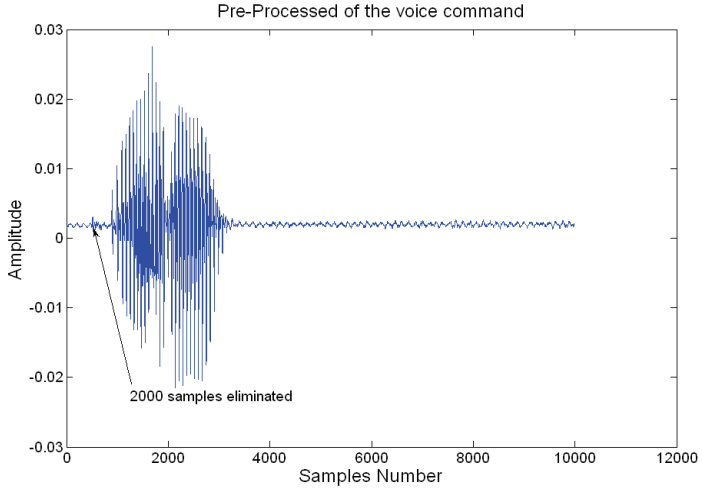


Fig. 9. Normalized Signal

In the next stage one proceeds to find the energy of the signal in the time domain. Two processes with the energy extraction algorithm were developed; in the first, the energy was obtained according to equation 2: In the second, the energy was normalized in as much in amplitude as in duration. 'E' is the energy and 'X' is the voice signal. Figure 10 illustrates the energy for the signal of the "open" command. This process is the same for the open, close, lights and television commands.

$$E(t) = 10.\log \sum_{i=1}^{L} |X(i)|^2 \tag{2}$$

After finding the energy, data analysis techniques of one-dimension for discrete signals in time were applied. The characteristics of each one of the words that form the data set are a faculty of the speaker. A total of 120 measurements were acquired from 4 words (open, close, lights and television), from which 30 measurements correspond to each of them. In order to train the system, the neural network toolbox of Matlab was used to create the model. In this case two-layers of feed-forward network were created. The network's input corresponds to one data set of 120 measurements, from which the first layer has ten neurons and the second layer has one neuron. The network is simulated and the output is acquired through targets; finally the network was trained for 1000 epochs and the output acquired.

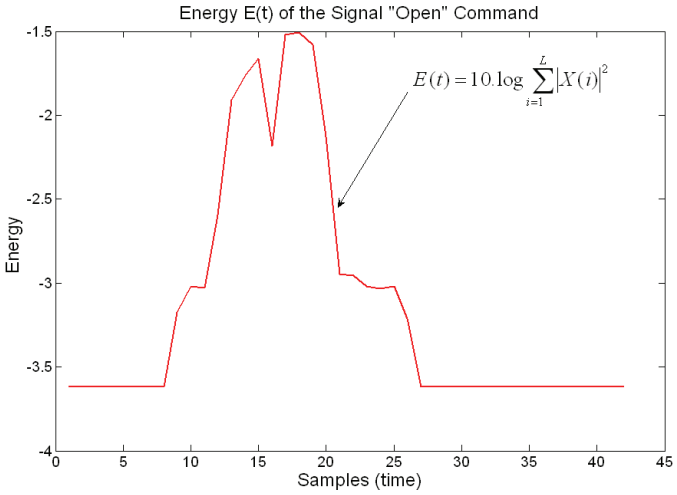


Fig. 10. Energy signal

Figure 11 shows three blocks in the recognition phase, where the first block acquires the voice signal, and afterwards the energy of the signal is obtained to be entered into the neural network. The function "wavrecord" is used to acquire the voice signal; subsequently, the energy of this command is obtained and it will be stored in the vector L, so that from Simulink of Matlab the network takes the value of this vector, making the pattern recognition.
Figure 12 shows the recognition algorithm; the first block takes the value of the vector L (i.e. Energy of the Signal) from the Workspace, the neural network finds the best success rate, while the third block is a function of the input value. The output network displays a number that can be a decimal value (i.e. Binary digits or LED (light emitting diodes) displays) for each of the commands, for example: 1, 2, 3, or 4, depending on the command input.

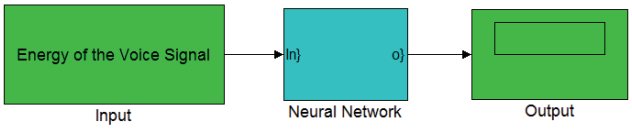


Fig. 11. Simulation of the Network

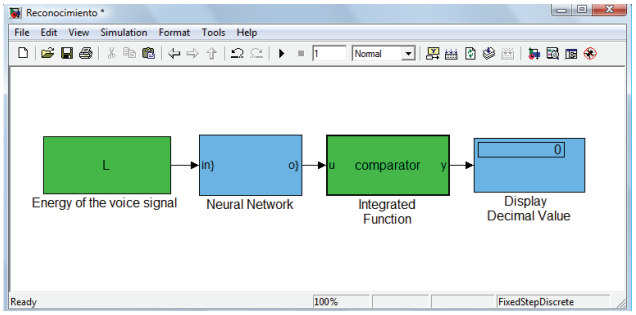


Fig. 12. Pattern Recognition System

4.3 DSP hardware (DSK TMS320C6416T of fixed point)

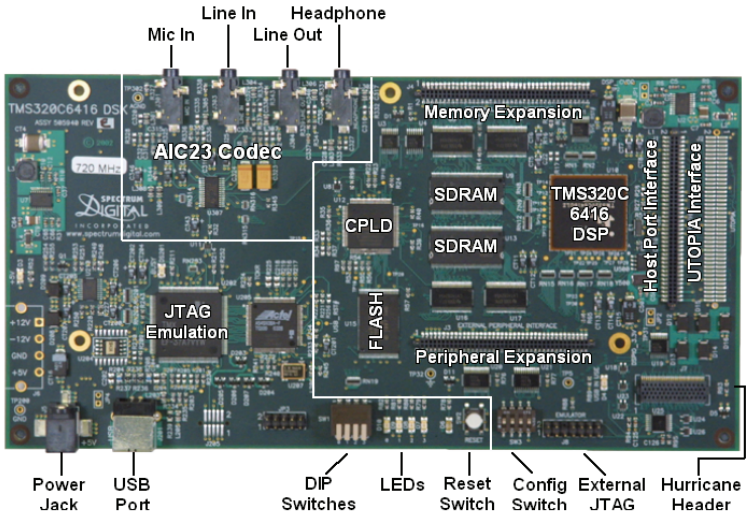


Fig. 13. C6416 DSK Board

The following section is important for the reader, for the reason that describes the DSP board which was used for the voice recognition in real time. The 6416 DSP Starter Kit (DSK) is a low-cost platform which lets customers evaluate and develop applications for the Texas Instruments C64x DSP family. The primary features of the DSK are: Speed 1 GHz, AIC23 Stereo Codec, four Position User DIP Switch and Four User LED's, On-board Flash and SDRAM. The figure 13 shows the main sections of the board.

The TMS320C6416 processor is the heart of the system. It is a core member of Texas Instruments'C64X line of "fixed point" DSP's whose distinguishing features are an extremely high performance 1 GHz VLIW DSP core and a large amount of fast on-chip memory (1Mbyte). On-chip peripherals include two independent external memory interfaces (EMIFs), 3 multi-channel buffered serial ports (McBSPs), three on-board timers and an enhanced DMA controller (EDMA). The 6416 represents the high end of TI's C6000 integral DSP line both in terms of computational performance and on-chip resources. The 6416 has a significant amount of internal memory so typical applications will have all code and data on-chip, especially designed for applications of speech recognition. External accesses are done through one of the EMIFs, either the 64-bit wide EMIFA or the 16-bit EMIFB. EMIFA is used for high bandwidth memories, such as; the SDRAM while EMIFB is used for non-performance critical devices, such as, the Flash memory that is loaded at boot time. A 32-bit subset of EMIFA is brought out to standard TI expansion bus connectors so additional functionality can be added on daughtercard modules.

DSPs are frequently used in audio processing applications so the DSK includes an on-board codec called the AIC23. Codec stands for coder/decoder. The job of the AIC23 is to code analog input samples into a digital format for the DSP to process; then decoded data comes out of the DSP to generate the processed analog output. Digital data is sent to and from the codec on McBSP2. The DSK has 4 LED´s and 4 DIP switches that allow users to interact with programs through simple LED displays and user input on the switches. Many of the included examples make use of these user interface options. The DSK implements the logic necessary to tie board components together in a programmable logical device called a CPLD. In addition to random glue logic, the CPLD implements a set of 4 software programmable registers that can be used to access the on-board LEDs and DIP switches as well as control the daughtercard interface.

The DSK uses a Texas Instruments' AIC23 (i.e. part #TLV320AIC23) stereo codec for input and output of audio signals, as shown in the next figure:

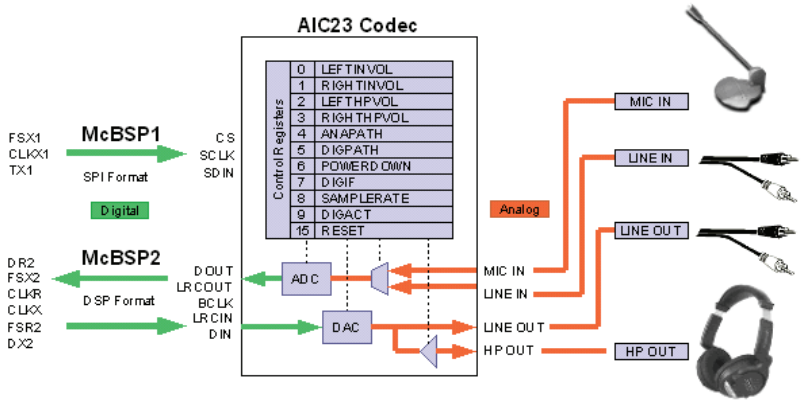


Fig. 14. (TLV320AIC23) audio stereo codec

The codec samples analog signals on the microphone or line inputs and converts them into digital data that can be processed by the DSP. When the DSP is finished with the data, it uses the codec to convert the samples back into analog signals on the line and headphone outputs so the user can hear the output.

The codec communicates using two serial channels; one to control the codec's internal configuration that registers, and one to send and receive digital audio samples. The AIC23 supports a variety of configurations that affect the data formats of the control and data channels.

4.4 Development tool

The following is the tool that provides the code needed to run the DSP Board. The development tool adapted to the DSP is the Code Composer Studio TI.

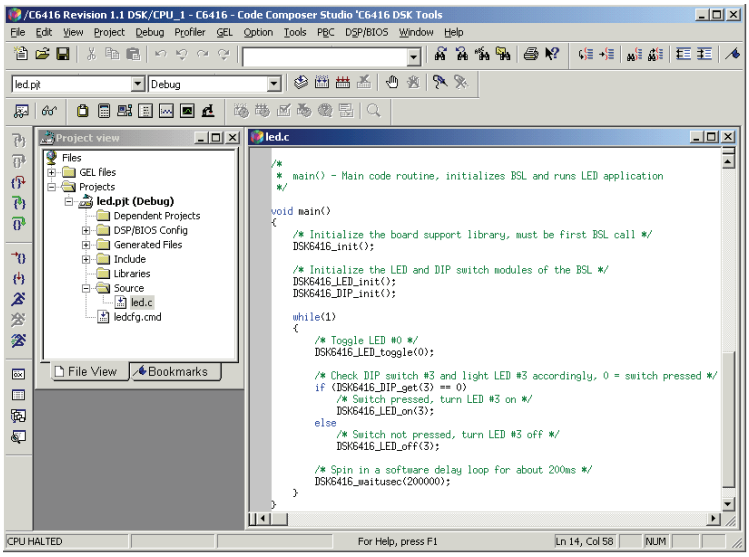


Fig. 15. Development Environment Tool: Code Composer Studio

It consists of an assembler, a C compiler, an integrated development environment (IDE, the graphical interface for the tools) and numerous support utilities like a hex format conversion tool. The DSK includes a special version of the Code Composer especially tailored to features on the 6416 DSK board. Other versions of the Code Composer are available that fully support each of TI's processor families on a wide variety of hardware targets.

The Code Composer IDE is the piece you see when you run the Code Composer. It consists of an editor for creating the source code, a project manager to identify the source files and options necessary for your programs and an integrated source level debugger that lets you examine the behavior of your program while it is running. The IDE is responsible for calling other components, like the compiler and assembler so that developers do not have to hassle running each tool manually.

The 6416 DSK includes a special device called a Joint Test Action Group (JTAG) emulator on-board that can directly access the register and memory state of the 6416 chip through a standardized JTAG interface port. When a user wants to monitor the progress of his program, the Code Composer sends commands to the emulator through its USB host interface to check on any data the user is interested in. This method is extremely powerful because programs can be debugged unobtrusively on real hardware targets without making any special provisions for debug-like external probes, software monitors or simulated

hardware. When designing your own hardware around the 6416, you can debug your application with the same wide functionality of the DSK simply by using the Code Composer with an external emulator and including a header for the JTAG interface signals. You should always be aware that the DSK is a different system from your PC; when you recompile a program with the Code Composer on your PC, you must specifically load it onto the 6416 on the DSK. Other things to be aware of are: 1) when you tell the Code Composer to run, it simply starts executing at the current program counter. If you want to restart the program, you must reset the program counter by using 'Debug' and 'Restart' or re-loading the program that implicitly sets the program counter. 2) After you have started a program running, it continues running on the DSP indefinitely. To stop it, you need to halt it with 'Debug' and 'Halt'.

4.5 Implementation on the DSP

Figure 16 shows a Graphic User Interface (GUI), where each stage of the process is visualized from the acquiring of the signal until the pattern recognition phase. The GUI consists of three-buttons; the first one to capture the signal from the microphone. The next button to obtain the energy and the recognition of the voice command of what has been pronounced before. The energy that was obtained is entered onto the net for its recognition.

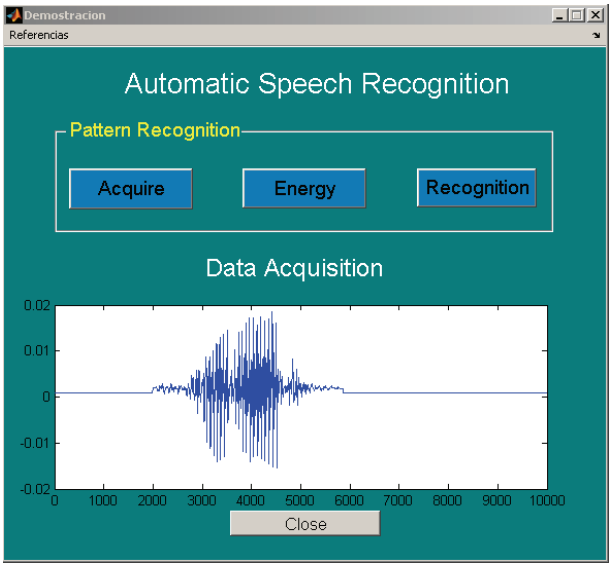


Fig. 16. ASR Interface

The last button makes the decision of the speech recognition command. After having carried out the two previous stages, it then, proceeds to identify the command with the DSP board; this started and builds the model.

Figure 17 shows that the code automatically generates, creates the model with respect to the new voice command through "The Code Composer Studio (CCS)". To achieve this, the tool TLC was used (i.e. Target Language Compiler) from Simulink, to translate the Simulink Block to language C or assembler. Figures 18 and 19 illustrate the experimental way to implement the ASR System using the DSK6416 from Texas Instruments.

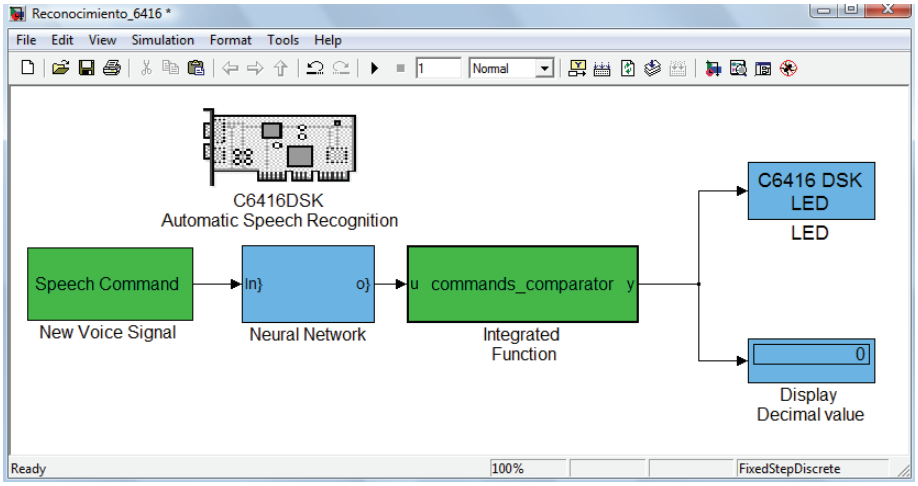


Fig. 17. Recognition model for the DSK6416

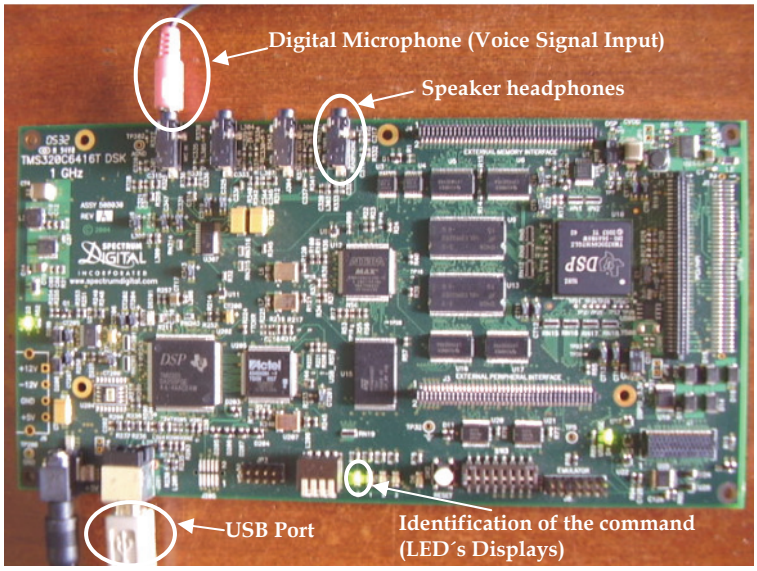


Fig.18. Connection of the DSP Board

The recognition system was subjected to a group of 4 habitual words (Open, Close, Lights and Television). The experiments were carried out pronouncing each one of the words 30 times and writing the successes and mistakes down obtaining 98% of a success rate of classification with only one error, which was detected with the "Ligths" command and located on the command "Close"; see Table 4. It is necessary to highlight that the tests were carried out for a single speaker and under conditions of absence of background noise and the pronunciation of the words that were made with the same characteristics with which it is possible to train the system.

Fig. 19. ASR with the DSP Hardware (DSK6416 TI)

Speech Command	1	2	3	4
Open	**30**	**-**	**-**	**-**
Close	**-**	**30**	**Error**	**-**
Lights	**-**	**-**	**29**	**-**
Television	**-**	**-**	**-**	**30**

Table 4. Results of the Classification with 4 speech commands

The results of this experiment concluded that is possible to implement a speech recognition algorithm (e.g. neural network) in a DSP, as a powerful tool or integrated system for automatic speech recognition.

5. Conclusions

This chapter has shown a review of the state of the art with some applications of the integrated systems, such as DPS´s to improve the performance of the ASR Systems. The chapter has summarized the VLSI technology and algorithm-oriented array architecture which appears to be effective, feasible, and economical in applications of speech recognition. DSP's are frequently used in a number of applications including telecommunications, automotive industries, control systems, medical-science, image processing, and now, in speech recognition. The potential of using statistical techniques and neural networks in speech recognition tasks (e.g. Isolated word recognition, solving limitations in human health, etc.) has been reviewed and preliminary results indicate that they have the potential to improve the performance of speech recognition systems. Dealing with the experimental

framework it is important to bring out that although initially tested with few words, in future work, it will be possible to make tests with a wider data set of voice commands for training.

6. References

Abdelhamied, K.; Waldron, M.; Fox, R.A. (1990). Automatic Recognition of Deaf Speech, *Volh Review*, volume: 2 pp. 121-30, Apr 1990.

Acevedo, C.M.D.; Nieves, M.G. (2007). Integrated System Approach for the Automatic Speech Recognition using Linear predict Coding and Neural Networks, *Electronics, Robotics and Automotive Mechanics Conference, 2007. CERMA 2007*, pp. 207-212, 25-28 Sept. 2007, ISBN: 978-0-7695-2974-5, Cuernavaca.

Barazesh, B.; Michalina, J.C.; Picco, A. (1988). A VLSI signal processor with complex arithmetic capability, *Circuits and Systems, IEEE Transactions on*, Volume: 35, Issue: 5, May 1988, pp. 495-505, ISSN: 0098-4094.

Batur, A.U.; Flinchbaugh, B.E.; Hayes, M.H. (2003). A DSP-based approach for the implementation of face recognition algorithms, *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, Volume: 2, pp. 253-256, ISSN: 1520-6149.

Beaufays, F.; Sankar, A.; Williams, S.; Weintraub, M. (2003). Learning name pronunciations in automatic speech recognition systems, *Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on*, pp. 233- 240, 3-5 Nov. 2003, ISSN: 1082-3409.

Benzeghiba, M.; Mori, D.; Deroo, O.; Dupont, S.; Erbes, T.; Jouvet, D.; Fissore, L.; Laface, P.; Mertins, A.; Ris, C.; Rose, R.; Tyagi, V.; Wellekens, C. (2007). Automatic speech recognition and speech variability: A review, *Speech Communication*, Volume 49, Issues 10-11, October-November 2007, pp. 763-786, ISSN: 0167-6393.

Castro, M.J.; Casacuberta, F. (1991). The use of multilayer perceptrons in isolated word recognition, *Artificial Neural Networks*, *Springer Berlin / Heidelberg*, pp. 469-476, ISBN: 978-3-540-54537-8.

Cheng, H.D.; Tang, Y.Y.; Suen, C.Y.; (1991), VLSI architecture for size-orientation-invariant pattern recognition, CompEuro '91. *Advanced Computer Technology, Reliable Systems and Applications. 5th Annual European Computer Conference. Proceedings*, pp. 63-67, ISBN: 0-8186-2141-9, 13-16 May 1991, Bologna, Italy.

Cong, L.; Asghar, S.; Cong, B. (2000). Robust speech recognition using neural networks and hidden Markov models, *Information Technology: Coding and Computing*, *Proceedings. International Conference on*, pp. 350-354, ISBN: 0-7695-0540-6.

Coy, A.; Barker, J. (2007). An automatic speech recognition system based on the scene analysis account of auditory perception, *Speech Communication*, Volume 49 , Issue 5, May 2007, pp. 384-401, ISSN:0167-6393.

Deller, J.R.; Hsu, D.; Ferrier, L.J. (1988). Encouraging results in the automated recognition of cerebral palsyspeech, *Biomedical Engineering, IEEE Transactions on*, Volume: 35, Issue: 3, pp.218-220, Mar 1988, ISSN: 0018-9294.

Ding, P.; He, L.; Yan, X.; Zhao, R.; Hao, J. (2006). Robust Technologies towards Automatic Speech Recognition in Car Noise Environments, *Signal Processing, 2006 8th International Conference on*, pp. 16-20, ISBN: 0-7803-9737-1, Beijing.

Gómez,A.J. (2007). Diseño e Implementación de un Sistema de Reconocimiento de Patrones de Voz Basado en un DSP Blackfin, *XII simposio de tratamiento de señales, imágenes y visión artificial. STSIVA*, Barranquilla (Colombia).

Jiankun, H.Z.X.; Jennings, A.; Lee, H.Y.J.; Wahyudi, D. (2001). DSP application in e-commerce security, *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, Volume: 2, pp. 1005-1008, ISBN: 0-7803-7041-4, Salt Lake City, UT, USA.

Jou, J.M.; Shiau, Y.H.; Huang, C.J. (2001). An efficient VLSI architecture for HMM-based speech recognition, Electronics, *Circuits and Systems, 2001. ICECS 2001 the 8th IEEE International Conference on*, Volume 1, pp. 469 – 472, ISBN: 0-7803-7057-0, 2-5 Sept. 2001.

Kehtarnavaz, N.; Kim, N.; Panahi, I. (2004). Digital signal processing system design: using LabVIEW and TMS320C6000, *Digital Signal Processing Workshop, 2004 and the 3rd IEEE Signal Processing Education Workshop. 2004 IEEE 11th*, pp. 10-14, ISBN: 0-7803-8434-2.

Kota, R.; Abdelhamied, K.A.; Goshorn, E.L. (1993). Isolated word recognition of deaf speech using artificial neural networks, *Biomedical Engineering Conference, 1993., Proceedings of the Twelfth Southern*, pp. 108-110, ISBN: 0-7803-0976-6, New Orleans.

Kostepen, M.H.; Kumar, G. (1991). Speech Recognition using Back-Propagation Neural Networks, *TENCON '91.1991 IEEE Region 10 International Conference on EC3-Energy, Computer, Communication and Control Systems*, Volume: 2, pp. 144-148, ISBN: 0-7803-0538-8, 28-30 Aug.

Kung, S. (1985). VLSI Array Processors, *ASSP Magazine, IEEE Signal Processing Magazine Inc*, Volume: 2, issue: 3, Part 1, July 1985, pp. 4-22, ISSN: 0740-7467.

Kwong, S.; Man, K.F. (1995). A Speech Coding Algorithm based on Predictive Coding, *Data Compression Conference, 1995. DCC '95. Proceedings*, pp. 455, ISBN: 0-8186-7012-6, 28-30 Mar Hong Kong.

Leitch, D.; Bain, K. (2000). Improving Access for Persons with Disabilities in Higher Education Using Speech Recognition Technology. *AVIOS Proceedings of The Speech Technology & Applications Expo*, pp. 83-86.

Lim, C.P.; Woo, S.C.; Loh, A.S.; Osman, R. (2000). Speech Recognition Using Artificial Neural Networks, wise, *First International Conference on Web Information Systems Engineering (WISE'00)*, Volume 1, pp. 419, ISBN: 0-7695-0577-5-1, 2000, IEEE Computer Society.

Louro, L.; Santos, P.; Rodrigues, N.; Silva, V.; Faria, S. (2003). DSP performance evaluation for motion estimation, *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on*, Volume: 2, pp. 137- 140, 1-4 July 2003, ISBN: 0-7803-7946-2.

Melnikoff, S.J.; Quigley, S.F.; Russell, M.J. (2002), Speech Recognition on an FPGA Using Discrete and Continuous Hidden Markov Models, *Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream, Springer Berlin / Heidelberg*, pp. 89-114, ISBN: 978-3-540-44108-3.

Montgomery, D.C.; Runger, G.C. (2006). Applied Statistics and Probability for Engineers, *John Wiley & Sons*, ISBN-13: 978-0-471-74589-1.

Pasero, E.; Montuori, A. (2003). Neural network based arithmetic coding for real-time audio transmission on the TMS320C6000 DSP platform, *Acoustics, Speech, and Signal*

Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on, Volume: 2, pp. 761-764, 6-10 April 2003, ISBN: 0-7803-7663-3.

Paulik, M.; Stuker, S.; Fugen, C.; Schultz, T.; Schaaf, T.; Waibel, A. (2005). Speech translation enhanced automatic speech recognition, *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, pp. 121 – 126, 27 Nov- 1 Dec 2005, ISBN: 0-7803-9478-X.

Peinado, A.; Segura, J.C. (2006). Speech Recognition Over digital Channels Robustness and standards, *John Wiley & Sons Ltd*, ISBN-13: 978-0-470-02400-3, ISBN-10: 0-470-02400-3, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England.

Phadke, S.; Limaye, R.; Verma, S.; Subramanian, K, (2004), On Design and Implementation of an Embedded Automatic Speech Recognition System, *VLSI Design, 2004. Proceedings. 17th International Conference on*, pp. 127-132, ISBN: 0-7695-2072-3.

Proakis, J.G., Manolakis, D.K. (2006). Digital Signal processing (4th Edition), *Prentice Hall*; 4 edition (April 7, 2006), ISBN-13: 978-0131873742.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, Volume 77, Issue 2, pp. 257-286, ISSN: 0018-9219.

Ramirez, J.; Segura, J.C.; Benitez, C.; Garcia, L.; Rubio, A.J. (2005). Statistical voice activity detection using a multiple observation likelihood ratio test, *IEEE Signal Processing Letters 12 (10)*, pp. 689–692.

Schroeder, M.R.; Atal, B.S. (1985). Code-excited Linear Prediction (CELP): High quality speech at very low bit rates, *IEEE International Conference on ICASSP '85, Acoustics, Speech, and Signal Processing*, Volume 10 pp.937-940, April 1985.

Tang, Y.Y.; Tao, L; Suen, C.Y. (1994). VLSI Arrays for Speech Processing with Linear Predictive, *Pattern Recognition, Conference C: Signal Processing, Proceedings of the 12th IAPR International Conference on*, pp. 357 – 359, ISBN: 0-8186-6275-1, University Chongqing, Oct 1994.

Tretter, S.A. (2008). Communication System Design Using DSP Algorithms with Laboratory Experiments for the TMS320C6713™ DSK, *Springer Science Business Media, LLC*, ISBN: 978-0-387-74885-6.

Wald, M.; (2005).Using Automatic Speech Recognition to Enhance Education for All Students: Turning a Vision into Reality, *Frontiers in Education, FIE '05. Proceedings 35th Annual Conference*, pp. S3G-22- S3G-25, 19-22 Oct. 2005, ISBN: 0-7803-9077-6.

Wald, M. (2006). Captioning for Deaf and Hard of Hearing People by Editing Automatic Speech Recognition in Real Time, *Proceedings of 10th International Conference on Computers Helping People with Special Needs ICCHP 2006, LNCS 4061*, pp. 683-690.

Yuan, M.; Lee, T.; Ching, P.C.; Zhu, Y. (2006). Speech recognition on DSP: issues on computational efficiency and performance analysis, *Microprocessors and Microsystems*, Volume 30, Issue 3, 5 May 2006, pp. 155-164, ISSN: 0141-9331.

27

Speech Recognition for Smart Homes

Ian McLoughlin and Hamid Reza Sharifzadeh
Nanyang Technological University
Singapore

1. Introduction

When Christopher Sholes created the QWERTY keyboard layout in the 1860s (often assumed to be for slowing down fast typists), few would have imagined that his invention would become the dominant input device of the 20th century. In the early years of the 21st century (the so called 'speed and information' century), its use remains dominant, despite many, arguably better, input devices having been invented. Surely it is time to consider alternatives, in particular the most natural method of human communications – spoken language.

Spoken language is not only natural, but in many cases is faster than typed, or mouse-driven input, and is accessible at times and in locations where keyboard, mouse and monitor (KMM) may not be convenient to use. In particular, in a world with growing penetration of embedded computers, the so-called 'smart home' may well see the first mass-market deployment of vocal interaction (VI) systems.

What is necessary in order to make VI a reality within the smart home? In fact much of the underlying technology already exists – many home appliances, electrical devices, infotainment systems, sensors and so on are sufficiently intelligent to be networked. Wireless home networks are fast, and very common. Speech synthesis technology can generate natural sounding speech. Microphone and loudspeaker technology is well-established. Modern computers are highly capable, relatively inexpensive, and – as embedded systems – have already penetrated almost all parts of a modern home. However the bottleneck in the realisation of smart home systems appears to have been the automatic speech recognition (ASR) and natural language understanding aspects.

In this chapter, we establish the case for automatic speech recognition (ASR) as part of VI within the home. We then overview appropriate ASR technology to present an analysis of the environment and operational conditions within the home related to ASR, in particular the argument of restricting vocabulary size to improve recognition accuracy. Finally, the discussion concludes with details on modifications to the widely used Sphinx ASR system for smart home deployment on embedded computers. We will demonstrate that such deployments are sensible, possible, and in fact will be coming to homes soon.

2. Smart Homes

The ongoing incorporation of modern digital technology into day to day living, is likely to see smart homes joining the next wave of computational technology penetration (McLoughlin & Sharifzadeh, 2007). This is an inevitable step in the increasing convenience

and user satisfaction in a world where users expect to be surrounded and served by many kinds of computers and digital consumer electronics products.
In parallel to this, advancements in networking have led to computer networks becoming common in everyday life (Tanenbaum, 1996) – driven primarily by the Internet. This has spawned new services, and new concepts of cost-effective and convenient connectivity, in particular wireless local-area networks. Such connectivity has in turn promoted the adoption of digital infotainment.

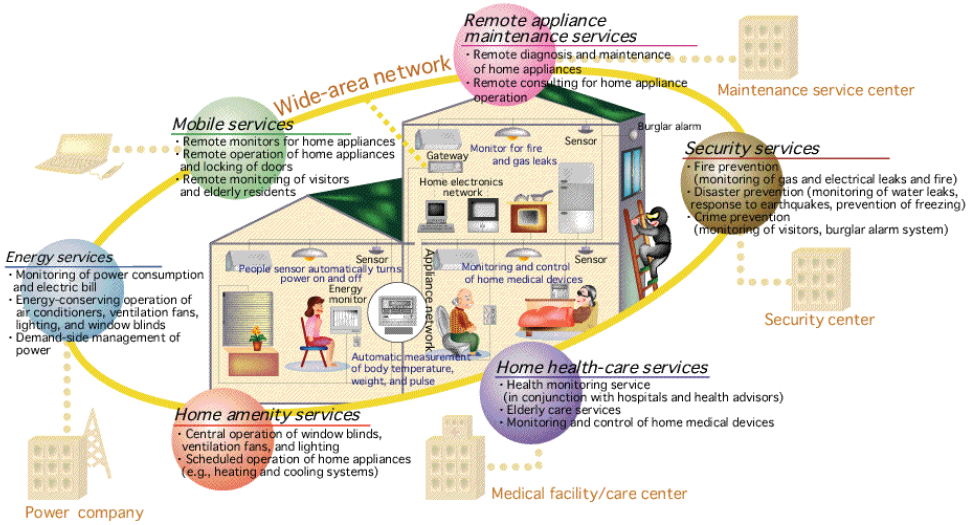


Fig. 1. An illustration of the range and scope of potential smart home services, reproduced by permission of ECHONET Consortium, Japan (ECHONET, 2008).

Recently trends reveal that consumers are more often buying bundles of services in the area of utilities and entertainment, while technical studies in the field of connected appliances (Lahrmann, 1998; Kango et al., 2002b) and home networking (Roy, 1999) are showing increasing promise, and increasing convergence in those areas. Figure 1 illustrates many of the services that can be provided for various activities within a house (ECHONET, 2008). An appliance can be defined as smart when it is 'an appliance whose data is available to all concerned at all times throughout its life cycle' (Kango et al., 2002). As a matter of fact, smart appliances often use emerging technologies and communications methods (Wang et al., 2000) to enable various services for both consumer and producer.
Here we define smart homes as those having characteristics such as central control of home appliances, networking ability, interaction with users through intelligent interfaces and so on. When considering natural interaction with users, one of the most user-friendly methods would be vocal interaction (VI). Most importantly, VI matches well the physical environment of the smart home. A VI system that can be accessed in the garage, bathroom, bedroom and kitchen would require at least a distributed set of microphones and loudspeakers, along with a centralised processing unit. A similar KMM solution will by contrast require keyboard, mouse and monitor in each room, or require the user to walk to a centralised location to perform input and control. The former solution is impractical for cost

and environmental reasons (imagine using KMM whilst in the shower), the latter solution is not user-friendly.
Practical VI presupposes a viable two way communications channel between user and machine that frees the user from a position in front of KMM. It does not totally replace a monitor – viewing holiday photographs is still more enjoyable with a monitor than through a loudspeaker – and in some instances a keyboard or mouse will still be necessary: such as entering or navigating complex technical documents. However a user-friendly VI system can augment the other access methods, and be more ubiquitous in accessibility, answering queries and allowing control when in the shower, whilst walking up stairs, in the dark and even during the messy process of stuffing a turkey.
The following sections focus on ASR issues as an enabling technology for VI in smart home computing, beginning with an overview of ASR evolution and state-of-the art.

3. ASR development

Half a century of ASR research has seen progressive improvements, from a simple machine responding to small set of sounds to advanced systems able to respond to fluently spoken natural language. To provide a technological perspective, some major highlights in the research and development of ASR systems are outlined:
The earliest attempts in ASR research, in the 1950s, exploited fundamental ideas of acoustic-phonetics, to try to devise systems for recognizing phonemes (Fry & Denes, 1959) and recognition of isolated digits from a single speaker (Davis et al., 1952). These attempts continued in the 1960s by the entry of several Japanese laboratories such as Radio Research Lab, NEC and Kyoto University to the arena. In the late 1960s, Martin and his colleagues at RCA Laboratories developed a set of elementary time-normalisation methods, based on the ability to reliably detect the presence of speech (Martin et al., 1964). Ultimately he founded one of the first companies which built, marketed and sold speech recognition products.
During the 1970s, speech recognition research achieved a number of significant milestones, firstly in the area of isolated word or discrete utterance recognition based on fundamental studies by in Russia (Velichko & Zagoruyko, 1970), Japan (Sakoe & Chiba, 1978), and in the United States (Itakura, 1975). Another milestone was the genesis of a longstanding group effort toward large vocabulary speech recognition at IBM. Finally, researchers in AT&T Bell Laboratories initiated a series of experiments aimed at making speech recognition systems that were truly speaker independent (Rabiner et al., 1979). To achieve this goal, sophisticated clustering algorithms were employed to determine the number of distinct patterns required to represent all variations of different words across a wide population of users. Over several years, this latter approach was progressed to the point at which the techniques for handling speaker independent patterns are now well understood and widely used.
Actually isolated word recognition was a key research focus in the 1970s, leading to continuous speech recognition research in the 1980s. During this decade, a shift in technology was observed from template-based approaches to statistical modelling, including the hidden Markov model (HMM) approach (Rabiner et al., 1989). Another new technology, reintroduced in the late 1980s, was the application of neural networks to speech recognition. Several system implementations based on neural networks were proposed (Weibel et al., 1989).

The 1980s was characterised by a major impetus to large vocabulary, continuous speech recognition systems led by the US Defense Advanced Research Projects Agency (DARPA) community, which sponsored a research programme to achieve high word accuracy for a thousand word continuous speech recognition database management task. Major research contributions included Carnegie-Mellon University (CMU), inventors of the well known Sphinx system (Lee et al., 1990), BBN with the BYBLOS system (Chow et al., 1987), Lincoln Labs (Paul, 1989), MIT (Zue et al., 1989), and AT&T Bell Labs (Lee et al., 1990).
The support of DARPA has continued since then, promoting speech recognition technology for a wide range of tasks. DARPA targets, and performance evaluations, have mostly been based on the measurement of word (or sentence) error rates as the system figure of merit. Such evaluations are conducted systematically over carefully designed tasks with progressive degrees of difficulty, ranging from the recognition of continuous speech spoken with stylized grammatical structure (as routinely used in military tasks, e.g., the Naval Resource Management task) to transcriptions of live (off-the-air) news broadcasts (e.g. NAB, involving a fairly large vocabulary over 20K words) and conversational speech.
In recent years, major attempts were focused on developing machines able communicate naturally with humans. Having dialogue management features in which speech applications are able to reach some desired state of understanding by making queries and confirmations (like human-to-human speech communications), are the main characteristics of these recent steps. Among such systems, Pegasus and Jupiter developed at MIT, have been particularly noteworthy demonstrators (Glass & Weinstein, 2001), and the How May I Help You (HMIHY) system at AT&T has been an equally noteworthy service first introduced as part of AT&T Customer Care for their Consumer Communications Services in 2000 (Gorin, 1996).
Finally, we can say after almost five decades of research and many valuable achievements along the way (Minker & Bennacef, 2004), the challenge of designing a machine that truly understands speech as well as an intelligent human, still remains. However, the accuracy of contemporary systems for specific tasks has gradually increased to the point where successful real-world deployment is perfectly feasible.

4. ASR in smart homes

Speech recognition applications can be classified into three broad groups (Rabiner, 1994) of isolated word recognition systems (each word is spoken with pauses before and afterwards, such as in bank or airport telephony services), small-vocabulary command-and-control applications, and large vocabulary continuous speech systems.
From an ASR point of view, a smart home system would aim to be a mixture of the second and third classes: predefined commands and menu navigation can be performed through a grammar-constrained command-and-control vocabulary, while email dictation and similar applications would involve large vocabulary continuous speech recognition. By and large, we can classify an ASR system in a smart home through its vocal interaction in two main categories: first are specific control applications which form the essence of smart homes, and second are general vocal applications which any ASR systems can carry out. We can summarize these categories and their characteristics as the following:

4.1 Smart home control

Probably the main feature of a smart home is the ability to vocally command the functions of the home and its appliances. We refer to this as the command-and-control function.

Most command-and-control applications have a small vocabulary size (0 to 50 words), reflecting the operations required to control the equipment. For example, commands for controlling the lights might include 'on', 'off', the location, and perhaps a few more words, depending on what additional operations are available. In addition the device being controlled should be identified (for example 'please turn on the bathroom light' is made up of a framing word 'please', an operation 'turn on', a location 'bathroom' and a device 'light'). Usually there is a direct mapping between the word or phrase and its semantics, i.e. the action to be carried out or the meaning to be associated with the words. However, more complex commands can be managed through a set of alternatives, where the vocabulary is restricted and known, such as week days or times of the day. As the number of alternative wordings increases, the task of listing all possible combinations and associating them with a given set of actions become unmanageable and so a grammar syntax is required that specifies, in a more abstract way, the words and phrases along with their permissible combinations.

4.2 Vocal interaction for information access

It is the view of the authors that for now, and some time to come, the Internet is likely to constitute the most common route for information access, alongside the stored files and archive of particular users. However it is accessing the vast and diverse World Wide Web (WWW) that is likely to post the most technically challenging tasks for VI. It is feasible to assume that the predominant graphical/textual nature of the current WWW is a natural consequence of the graphical/textual bias of HTML, which is most of all due to the way in which users are conditioned to interact with computers through KMM. If users commonly interacted with the WWW in a vocal fashion then it is quite possible that voice-enabled 'pages' would appear. To date this has not been the case.

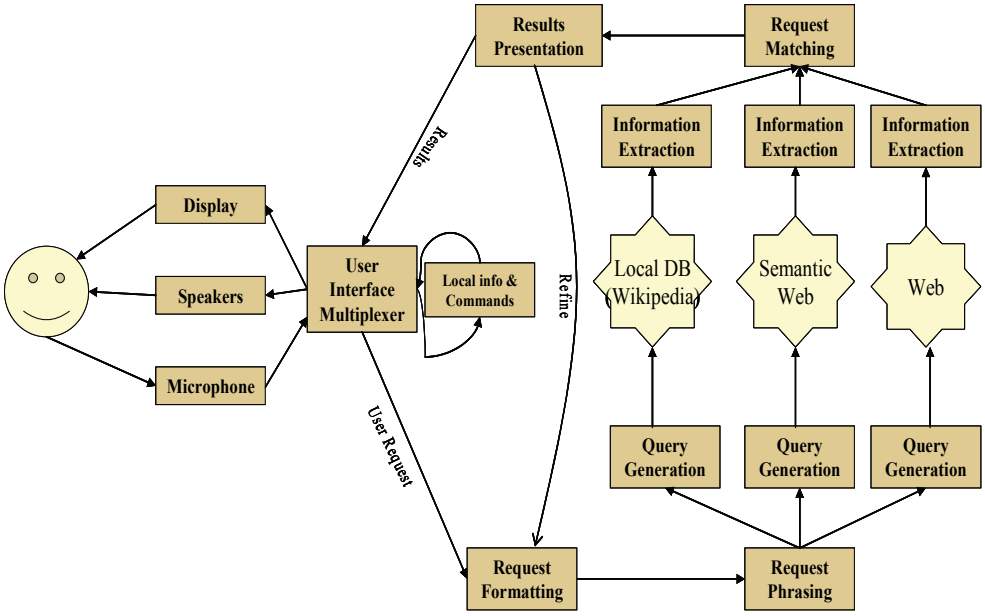


Fig. 2. Overall structure of the WWW vocal query access system.

The semantic web currently being promoted and researched by Tim Berners-Lee and others (see Wikipedia 2008), goes a long way towards providing a solution: it divorces the graphical/textual nature of web pages from their information content. In the semantic web, pages are based around information. This information can then be marked up and displayed *graphically* if required. When designing smart home services benefiting from vocal interactions of the semantic web, the same information could be marked up and presented vocally, where the nature of the information warrants a vocal response (or the user requires a vocal response).

There are three alternative methods of VI relating to the WWW resource:

- The few semantic web pages (with information extracted and then, either as specified in the page, or using local preferences, converted to speech), and then presented vocally.
- HTML web pages, with information extracted, refined then presented vocally.
- Vocally-marked up web pages, presented vocally.

Figure 2 shows the overall structure proposed by the authors for vocal access to the WWW. On the left is the core vocal response system handling information transfer to and from the user. A user interface and multiplexer allow different forms of information to be combined together. Local information and commands relate to system operation: asking the computer to repeat itself, take and replay messages, give the time, update status, increase volume and so on. For the current discussion, it is the ASR aspects of the VI system which are most interesting:

User requests are formatted into queries, which are then phrased as required and issued simultaneously to the web, the semantic web and a local Wikipedia database. The semantic web is preferred, followed by Wikipedia and then the WWW.

WWW responses can then be refined by the local Wikipedia database. For example too many unrelated hits in Wikipedia indicate that query adjustments may be required. Refinement may also involve asking the user to choose between several options, or may simply require rephrasing the question presented to the information sources. Since the database is local, search time is almost instantaneous, allowing a very rapid request for refinement of queries to be put to the user if required before the WWW search may have completed.

Finally, results are obtained as either Wikipedia information, web pages or semantic information. These are analysed, formatted, and presented to the user. Depending on the context, information type and amount, the answer is either given vocally, graphically or textually. A query cache and learning system (not shown) can be used to improve query processing and matching based on the results of previous queries.

4.3 Dictation

Dictation involves the automatic translation of speech into written form, and is differentiated from other speech recognition functions mostly because user input does not need to be interpreted (although doing so may well aid recognition accuracy), and usually there is little or no dialogue between user and machine.

Dictation systems imply large vocabularies and, in some cases, an application will include an additional specialist vocabulary for the application in question (McTear, 2004). Domain-specific systems can lead to increased accuracy.

4.4 Open-ended commands

Imagine a computer-based butler. This would need to be able to interpret, understand, and carry out complex commands. For example 'please book me a night flight to Tokyo next Thursday and reserve a suitable suite at the Grand Hyatt Hotel', or perhaps, 'please check my address book and make a dental appointment at an appropriate time over the next few days'. There is not only a significant vocabulary size implied in such commands, but also the understanding of the meaning of the command, and an appreciation of related factors: for example the words 'suitable' and 'appropriate' demand value judgments.

Quite clearly these commands impinge on the area of natural language processing (NLP) research, but do represent the ultimate destination of research into smart home systems – the home which is able to conveniently cater to our needs. It is likely to be several years before such open-ended commands can be handled successfully by automated systems.

4.5 General ASR applications

Since the advent of computers, the need to collect and neatly present documents has required textual data entry. The potential of ASR systems is that much of this process can be performed through VI. Many data entry applications involve predefined items such as name and address, for example form completion, package sorting, equipment maintenance, and traffic accident reports.

Data entry applications usually have limited vocabulary size including numbers, name and address details, and several additional control words. However, there may also be a requirement for a significant number of application-specific words, depending on the application type (McTear, 2004).

Similarly, computer games will likely form potential future applications which can only be guessed at currently. Educational use also had great potential – much teaching is conveyed vocally in schools and universities worldwide, and an individual education for children, adapting to their pace and needs, may well become a reality with advanced VI systems.

Finally, most societies contain people battling with loneliness, nobody to talk to, nobody who understands them. The thought of a machine companion may seem far-fetched today, but in the opinion of the authors it is only a matter of time before a sufficiently powerful and responsive computer system could become a best friend to some in society. As with the entire smart home project, what is needed is a sufficiently natural and accurate VI system coupled with a computer system that could pass the Turing test, at least when conducted by their potential clients.

5. Vocabulary size and performance

The ability of a system to recognise captured speech, to cater for intra- and inter-speaker variability, and the processing time allowable for recognising utterances are three main usability issues related to VI systems. Other issues include training requirements, robustness, linguistic flexibility and dialogue interaction.

Many factors in ASR for VI can be controlled. For example the variability of speech is mostly confined to a limited set of uses: linguistic flexibility can, and should, be constrained through appropriate grammar design (which is focus of section 6) and so on. The ability to accurately recognize captured speech that has been constrained in the ways discussed above will then depend primarily upon vocabulary size and speech-to-noise ratio. Thus we can

improve recognition firstly by restricting vocabulary size, and secondly by improving signal-to-noise ratio. The former task, constraint of vocabulary size, is the role of constructed grammar in VI systems.

It is well known that vocabulary restrictions can lead to recognition improvements whether these are domain based (Chevalier et al., 1995) or simply involve search-size restriction (Kamm et al., 1994). Similarly the quality of captured speech obviously affects recognition accuracy (Sun et al., 2004). Real-time response is also a desirable characteristic in many cases.

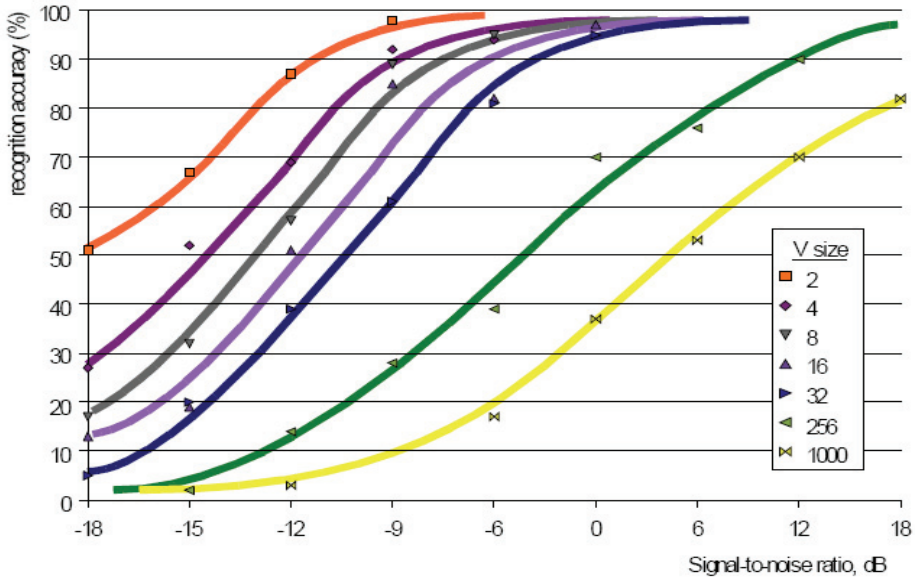


Fig. 3. Effect of vocabulary size and SNR on word recognition by humans, after data obtained in (Miller et al, 1951).

Actually the three aspects of performance: recognition speed, memory resource requirements, and recognition accuracy, are in mutual conflict, since it is relatively easy to improve recognition speed and reduce memory requirements at the expense of reduction in accuracy (Ravishankar, 1996). The task for designing a vocal response system is thus to restrict vocabulary size as much as practicable at each point in a conversation. However, in order to determine how much the vocabulary should be restricted, it is useful to relate vocabulary size to recognition accuracy at a given noise level.

Automatic speech recognition systems often use domain-specific and application-specific customisations to improve performance, but vocabulary size is important in any generic ASR system regardless of techniques used for their implementation.

Some systems have been designed from the ground-up to allow for examination of the effects of vocabulary restrictions, such as the Bellcore system (Kamm et al., 1994) which provided comparative performance figures against vocabulary size: it sported a very large but variable vocabulary of up to 1.5 million individual names. Recognition accuracy decreased linearly with logarithmic increase in directory size (Kamm et al., 1994).

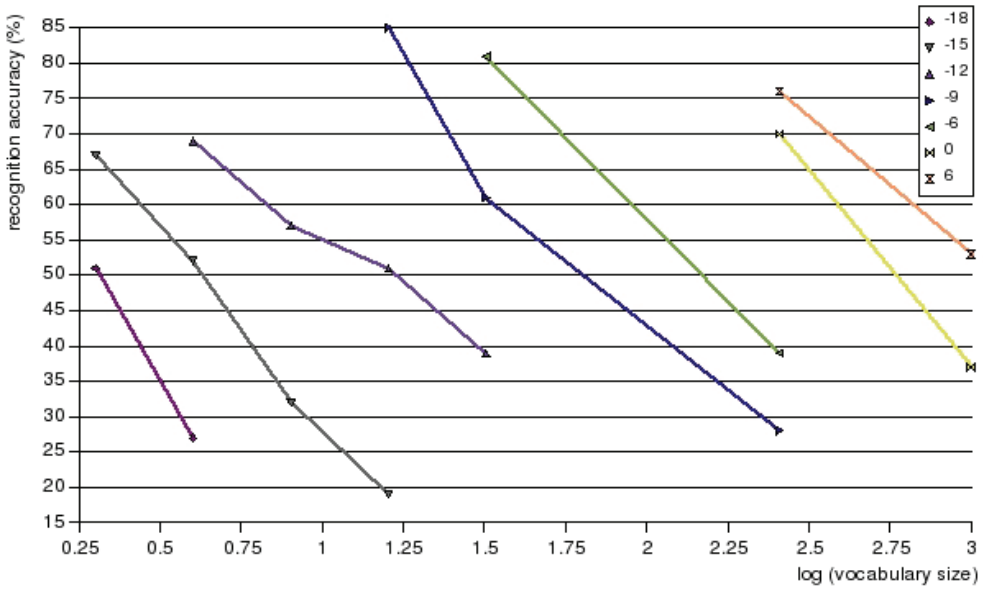


Fig. 4. Plot of speech recognition accuracy results showing the linear decrease in recognition accuracy with logarithmic increase in vocabulary size in the presence of various levels of SNR (McLoughlin, 2009).

To obtain a metric capable of identifying voice recognition performance, we can conjecture that, in the absence of noise and distortion, recognition by human beings describes an approximate upper limit on machine recognition capabilities: the human brain and hearing system is undoubtedly designed to match closely with the human speech creation apparatus. In addition, healthy humans grow up from infanthood with an in-built feedback loop to match the two.

While digital signal processing systems may well perform better at handling additive noise and distortion than the human brain, to date computers have not demonstrated better recognition accuracy in the real world than humans. As an upper limit it is thus instructive to consider results such as those from Miller et al. (Miller et al, 1951) in which human recognition accuracy was measured against word vocabulary size in various noise levels.

The graph of figure 3 plots several of Miller's tabulated results (Kryter, 1995), to show percentage recognition accuracy against an SNR range of between -18 and +18dB$_{SNR}$ with results fit to a sigmoid curve tapering off at approximately 100% accuracy and 0% accuracy at either extreme of SNR. Note that the centre region of each line is straight so that, irrespective of the vocabulary, a logarithmic relationship exists between SNR and recognition accuracy. Excluding the sigmoid endpoints and plotting recognition accuracy against the logarithm of vocabulary size, as in figure 4, clarifies this relationship (McLoughlin, 2009).

Considering that the published evidence discussed above for both human and computer recognition of speech shows a similar relationship, we state that in the presence of moderate levels of SNR, recognition accuracy (A) reduces in line with logarithmic increase in

vocabulary size (V), related by some system dependent scaling factor which will represent by γ:

$$A^{-1} = \gamma \log (V) \quad (1)$$

6. Grammar structures for smart homes dialogues

As seen in section 4, VI for smart homes involves different levels of complexity ranging from an isolated word recognizer to an unconstrained ASR dictation system. For isolated word recognition, a speech recognizer attempts to detect commands by focusing on keywords in segmented natural sentences from the talker. A more complicated system, often referred to as a system driven dialogue (Nakano et al., 2006), can allow a user to complete information fields. Large vocabulary continuous speech recognition (LVCSR), by contrast, requires language models or grammars to select the most likely word sequence from the relatively large number of alternative word hypotheses produced during the search process at each stage in a conversation. Simple recognition tasks can use rule-based regular or context free grammars, where the system only recognizes a limited vocabulary.

In a speech-based smart home system, a VI session begins with a configurable spoken identifier phrase to 'attract the attention' of the ASR computer. From this point on, the adoption of grammar and syntactic structures, although these must be learnt by the user and thus reduce user friendliness, are crucial in maintaining the recognition accuracy of such systems.

Following attention, a tree of vocabulary options is possible. The aim at each stage of is to maximise recognition accuracy for given SNR, by reducing the vocabulary search space. An example sentence for one speaker might be:

ACTION – argument – (optional MODIFIER- (optional argument)) – PAUSE/REPEAT

In general, each of the action, argument and modifier sub phrases will have differing vocabulary characteristics, but it is a common feature of such systems that vocabulary size be minimised through syntactic structure. Let us proposed that each sub-phrase is classified by the following arguments:

- *Vocabulary consisting of V elements*
- *Accuracy requirement, R*
- *Length constraint, L*
- *Interruptibility type, T*

The characteristics of each sub-phrase are known in advance, having been defined by the grammar syntax, with recall prompted by traversal to the current branch. The length constraint is used to help detect end-of-phrase overrun conditions, and perhaps a missed MODIFIER phrase.

One or more phrases by each party in the communication comprise a conversation (or dialogue). An overall conversation, and indeed a phrase made of sub phrases has its own accuracy requirement. However this level of end-to-end link assurance is unwieldy for anything but confirming that an important action should or should not occur as the result of a conversation.

Sub-phrase vocabulary may occasionally need to be unconstrained (such as when performing a web search or dictating an email), but at other times could be limited (in this mode, we can use keywords or embedded phrases in natural sentences to command the

system). Potentially these two classes of recognition task could be performed with two different types of ASR software employing isolated word and continuous speech recognition respectively (Nakano et al., 2006). However both are catered for in Sphinx2 using two density acoustic models, namely semi-continuous and continuous. Some other flexible speech recognition systems have been introduced.

One system by Furui (Furui, 2001) uses a method of automatically summarizing speech, sentence by sentence. This is quite domain-specific (closed-domain) with limited vocabulary size (designed for news transcription), and may not be directly applicable to a continuously variable vocabulary size smart home system. However it does perform very well, and provides an indication of the customisations available in such systems.

Vocabulary size, V, impacts recognition accuracy, and needs to be related to accuracy requirement, R. Since 100% accuracy is unlikely, smart home systems need to be able to cope with inaccuracies through careful design of the interaction and confirmation processes. In particular, a speech recognizer that provides a confidence level, C can tie in with sub-phrase arguments to determine requests-for-clarification (RFC) which are themselves serviced through examination of the interruptibility type, T.

So given a recognition confidence level C, an RFC will be triggered if:

$$\frac{C\gamma}{\log(V)} < R \tag{2}$$

Where γ is system and scale dependent, determined through system training. Interruptibility type includes two super-classes of 'immediate' and 'end-of-phrase'. Immediate interrupts may be verbal or non-verbal (a light, a tone, a gesture such as a raised hand, or a perplexed look on a listeners' face based on the designed interface). An immediate interrupt would be useful either when the utterance is expected to be so long that it is inconvenient to wait to the end, or when the meaning requires clarification up-front. An example of an immediate interrupt would be during an email dictation, where the meaning of an uncertain word needs to be checked as soon as the uncertainty is discovered – reviewing a long sentence that has just been spoken in order to correct a single mistaken word is both time consuming and clumsy in computer dialogue terms.

An end-of-phrase interrupt is located at a natural reply juncture, and could be entirely natural to the speaker as in "did you ask me to turn on the light?"

7. Embedded speech recognition

Nowadays, embedded speech technology as an active research area attracts not only researchers from academia but also industrial groups interested to invest in this promising new market. Thus, more and more companies have launched embedded speech systems. These provide alternative control interfaces for consumer appliances to replace knobs, switches, buttons and so on. In specific niche applications, with limited vocabulary size, the success of such niche products may well advance the public acceptance of speech technology. Current examples include voice dialling for GSM telephones, and media players.

As consumer devices become increasingly complex, naturally the range of features increases, and thus it has become more and more difficult for users to produce the appropriate sequences of key presses to set a control. A typical example is the inability of

most people to use a remote control to set the timer on their video recorder to record forthcoming broadcasts. In addition, as devices decrease in size, and average users increase in age, manual manipulation has similarly become more difficult. From a system architecture point of view, embedded speech recognition is now becoming considered a simple approach to user interfacing. Adoption in the embedded sphere contrasts with the more sluggish adoption of larger distributed system approaches (Tan & Varga, 2008).

However there is a price to be paid for such architectural simplicity: complex speech recognition algorithms must run on under-resourced consumer devices. In fact, this forces the development of special techniques to cope with limited resources in terms of computing speed and memory on such system.

Resource scarcity limits the available applications: on the other hand it forces algorithm designers to optimise techniques in order to guarantee sufficient recognition performance even in adverse conditions, on limited platforms, and with significant memory constraints (Tan & Varga, 2008). Of course, ongoing advances in semiconductor technologies mean that such constraints will naturally become less significant over time.

In fact, increased computing resources coupled with more sophisticated software methods may be expected to narrow the performance differential between embedded and server-based recognition applications: the border between applications realized by these techniques will narrow, allowing for advanced features such as natural language understanding to become possible in an embedded context rather than simple command-and-control systems. At this point there will no longer be significant technological barriers to use of embedded systems to create a smart VI-enabled home.

However at present, embedded devices typically have relatively slow memory access, and a scarcity of system resources, so it is necessary to employ a fast and lightweight speech recognition engine in such contexts. Several such embedded ASR systems have been introduced in (Hataoka et al., 2002), (Levy et al., 2004), and (Phadke et al., 2004) for sophisticated human computer interfaces within car information systems, cellular phones, and interaction device for physically handicapped persons (and other embedded applications) respectively.

It is also possible to perform speech recognition in smart homes by utilising a centralised server which performs the processing, connected to a set of microphones and loudspeakers scattered throughout a house: this requires significantly greater communications bandwidth than a distributed system (since there may be arrays of several microphones in each location, each with 16 bit sample depth and perhaps 20kHz sampling rate), introduces communications delays, but allows the ASR engine to operate on a faster computer with fewer memory constraints.

As the capabilities of embedded systems continue to improve, the argument for a centralised solution will weaken. We confine the discussion here to a set of distributed embedded systems scattered throughout a smart home, each capable of performing speech recognition, and VI. Low-bandwidth communications between devices in such a scenario to allow co-operative ASR (or CPU cycle-sharing) is an ongoing research theme of the authors, but will not affect the basic conclusions at this stage.

In the next section, the open source Sphinx is described as a reasonable choice among existing ASRs for smart home services. We will explain why Sphinx is suitable for utilisation in smart homes as a VI core through examining its capabilities in an embedded speech recognition context.

8. Sphinx as an ASR for smart homes

Among many automatic speech recognizers available for different applications with various features, the open source Sphinx recognizer is an excellent examples of a flexible modern speech recognition system. Sphinx, originally developed at Carnegie Mellon University in the USA, provides and integrates several capabilities that allow it to be adapted for a wide range of different speech recognition applications.

At one extreme, it can be used for single word recognition, or expanded at the other extreme to large vocabularies containing tens of thousands of words. In terms of resource constraints, it can run on anything from a tiny embedded system (PocketSphinx) to a large and powerful server (which could run the Java language version Sphinx-4). Sphinx is regularly updated and evaluated within the speech recognition research field.

Sphinx, in common with most current ASR implementations, relies upon Hidden Markov Modelling to match speech features to stored patterns (Lee, 1989). It is highly configurable and incredibly flexible – the required features used can be selected as required.

Sphinx2, the decoding engine for Sphinx II, can be a good choice for smart home services, provided several appropriate model files and databases are used. These are classified into three categories:

a. Pronunciation lexicon/dictionary defining words of current interest, and a phonemic pronunciation for each.
b. Acoustic models based on Hidden Markov Models (HMM) for base phones and triphones. Sphinx2 uses both semi-continuous and continuous density acoustic models which are typically generated by the Sphinx acoustic model trainer.
c. A predetermined language model accepting two flavours of language: either the finite state graph (FSG) and N-gram models (where N is either two or three).

Apart from ordinary words, noise or filler words can be specified for a particular application by placing them in a corresponding dictionary. The N-gram language model additionally includes begin-sentence and end-sentence symbols, denoted <S> and </S>, normally representing silence. These can be used in continuous speech applications for quiet homes, but may need to be augmented with predetermined start/stop attention phrases.

The core speech decoder operates on finite-length segments of speech or utterance, one utterance at a time. An utterance can be up to one minute long, but in practice most applications handle sentences or phrases which are much shorter than this. For real-time use, processing must be continuous, with a response latency that is not excessive. Response delays of a second or more may well lead to user annoyance.

As mentioned in section 3, smart home services are mixture of small (for command-and-control applications) and large (for email dictation and similar applications) continuous vocabulary speech systems; thus, we need an ASR which supports both modes. As a comparison, the concept is similar to (Nakano et al., 2006) in which the Honda ASIMO humanoid robot has two dialogue strategies: a) task-oriented dialogues which utilize the outputs of a small vocabulary speech recognizer, and b) non-task-oriented dialogues which utilize the outputs of a large vocabulary speech recognizer.

The major difference between Sphinx and this approach occurs during the implementation phase where ASIMO deploys two different ASR engines (Julian for small vocabulary and Julius for large one). This differs to the authors Sphinx-based system which proposed a single recognition engine that not only caters for the needs of both tasks, but has a continuously variable vocabulary instead of two extremes as in the ASIMO case. This

therefore allows a continuum of dialogue complexities to suit the changing needs of the vocal human-computer interaction. The particular vocabulary in use at any one time would depend upon the current position in the grammar syntax tree.
As a noticeable choice in embedded applications necessary for smart homes, Sphinx II is available in an embedded version called PocketSphinx. Sphinx II was the baseline system for creating PocketSphinx because it is was faster than other recognizers currently available in the Sphinx family (Huggins-Daines et al., 2006). The developers claim PocketSphinx is able to address several technical challenges in deployment of speech applications on embedded devices. These challenges include computational requirements of continuous speech recognition for a medium to large vocabulary scenario, the need to minimize the size and power consumption for embedded devices which imposes further restrictions on capabilities and so on (Huggins-Daines et al., 2006).
Actually, PocketSphinx, by creating a four-layer framework including: frame layer, Gaussian mixture model (GMM) layer, Gaussian layer, and component layer, allows for straightforward categorisation of different speed-up techniques based upon the layer(s) within which they operate.

9. Audio aspects

As mentioned in section 1, smart home VI provides a good implementation target for practical ASR: the set of users is small and can be predetermined (especially pre-trained, and thus switched-speaker-dependent ASR becomes possible), physical locations are well-defined, the command set and grammar can be constrained, and many noise sources are already under the control of (or monitored by) a home control system.
In terms of the user set, for a family home, each member would separately train the system to accommodate their voices. A speaker recognition system could then detect the speech of each user and switch the appropriate acoustic models into Sphinx. It would be reasonable for such a system to be usable only by a small group of people.
Physical locations – the rooms in the house – will have relatively constant acoustic characteristics, and thus those characteristics that can be catered for by audio pre-processing. Major sources of acoustic noise, such as home theatre, audio entertainment systems, games consoles and so on, would likely be under the control of the VI system (or electronically connected to them) so that methods such as spectral subtraction (Boll, 1979) would perform well, having advanced knowledge of the interfering noise.
It would also be entirely acceptable for a VI system, when being required to perform a more difficult recognition task, such as LVCSR for email dictation, to automatically reduce the audio volume of currently operating entertainment devices.
Suitable noise reduction techniques for a smart home VI system may include methods such as adaptive noise cancellation (ANC) (Hataoka et al., 1998) or spectral subtraction which have been optimized for embedded use (Hataoka et al., 2002).
The largest difference between a smart home ASR deployment and one of the current computer-based or telephone-based dictation systems is microphone placement (McLoughlin, 2009): in the latter, headset or handset microphones are used which are close to the speakers mouth. A smart home system able to respond to queries anywhere within a room in the house would have a much harder recognition task to perform. Microphone arrays, steered by phase adjustments, are able to 'focus' the microphone on a speakers mouth (Dorf, 2006), in some cases, and with some success.

However more preferable is a method of encouraging users to direct their own speech in the same way that they do when interacting with other humans: they turn to face them, or at least move or lean closer. This behaviour can be encouraged in a smart home by providing a focus for the users. This might take the form of a robot head/face, which has an added advantage of being capable of providing expressions – a great assistance during a dialogue when, for example, lack of understanding can be communicated back to a user non-verbally. This research is currently almost exclusively the domain of advanced Japanese researchers: see for example (Nakano et al., 2006).
A reasonable alternative is the use of a mobile device, carried by a user, which they can speak into (Prior, 2008). This significantly simplifies the required audio processing, at the expense of requiring the user to carry such a device.

10. Usability criteria

It is sensible to develop smart home VI usability criteria. Users of current ASR systems may well appreciate the frustrations of mis-tuned and under-performing systems where backtracking and corrections require more effort and time than does the process of input itself. Many give up, preferring to switch back to a clumsy, but reliable, keyboard.
Despite the theoretical reasons for adopting a VI system to free users from the constraint of KMM, most users have in fact grown up with such constraints and are not unhappy with them. By contrast, users of current ASR systems tend to experience mostly frustration in their interactions - a major reason why LVCSR is not particularly popular today. It is therefore only when viable alternatives to the KMM have been demonstrated in niche areas that the general public will adopt a new perspective on the use of computer technology without touch and vision-based user interfaces. Such a niche area is likely to be within the smart home context, where significant advantages exist for ASR: a limited set of users, relatively constant acoustic characteristics, constraints upon the tasks to be performed and so on.
The assumptions for a smart home are that training is performed in advance, the major sources of acoustic interference (such as infotainment and gaming systems) are under control or at least linked in to the smart home electronics, the VI operational syntax is self-contained, limited and known to the user, and that the user has had time to become familiar with the system.
Major performance criteria include the percentage of tasks which are completed without secondary user intervention (and then the degree of intervention required for those tasks which are not completed straight off). For use by the general public, a useful performance measure may well be how often the user must resort to a KMM solution for performing VI-oriented tasks. However, as with all technology deployments to the general public, a final successful verdict can only be pronounced when sales figures begin to indicate mass-marked adoption of such technology.

11. Conclusion

The major components of a smart home ASR system currently exist within the speech recognition research community, as the evolutionary result of half a century of applied and academic research. The command-and-control application of appliances and devices within the home, in particular the constrained grammar syntax, allows a recognizer such as Sphinx

to operate with high levels of accuracy. Results are presented here which relate accuracy to vocabulary size, and associate metrics for reducing vocabulary (and thus maximising accuracy) through the use of restricted grammars for specialised applications.

Audio aspects related to the smart home, and the use of LVCSR for multi-user dictation tasks are currently major research thrusts, as is the adaption of ASR systems for use in embedded devices. The application of speech recognition for performing WWW queries is probably particularly important for the adoption of such systems within a usable smart home context, and this work is ongoing, and likely to be greatly assisted if current research efforts towards a semantic web will impact the WWW as a whole.

The future of ASR within smart homes will be assured first by the creation of niche applications which deliver to users in a friendly and capable fashion. That the technology largely exists has been demonstrated here, although there is still some way to go before such technology will be adopted by the general public.

12. References

Boll, S. F. (1979). "Suppression of acoustic noise in speech using spectral subtraction", IEEE Transactions on Signal Processing, Vol. 27, No. 2, pp. 113-120.

Chevalier, H.; Ingold, C.; Kunz, C.; Moore, C.; Roven, C.; Yamron, J.; Baker B.; Bamberg, P.; Bridle, S.; Bruce, T.; Weader, A. (1996)"Large-vocabulary speech recognition in specialized domains", Proc. ICASSP, Vol. 1 pp. 217-220.

Chow, Y.L.; Dunham, M.O.; Kimball, O. A.; Krasner, M. A.;. Kubala, G. F ; Makhoul, J.;Roucos, S. ; Schwartz, R. M. (1987). "BBYLOS: The BBN continuous speech recognition system," Proc. ICASSP., pp.89-92.

Davis, K. H.; Biddulph, R.; Balashek, S. (1952). "Automatic recognition of spoken digits", J. Acoust. Soc. Am., Vol 24, No. 6.

Dorf, C. (2006). Circuits, Signals, and Speech And Image Processing, CRC Press.

ECHONET Consortium (2008). Energy Conservation and Homecare Network, www.echonet.gr.jp, last accessed July 2008.

Fry, D. B.; Denes, P. (1959). "The design and operation of the mechanical speech recognizer at University College London", J. British Inst. Radio Engr., Vol. 19, No. 4, pp. 211-229.

Furui, S. (2001). "Toward flexible speech recognition-recent progress at Tokyo Institute of Technology", Canadian Conference on Electrical and Computer Engineering, Vol. 1, pp. 631-636.

Glass, J.; Weinstein, E. (2001). "SpeechBuilder: Facilitating Spoken Dialogue System Development", 7th European Conf. on Speech Communication and Technology, Aalborg Denmark, pp. 1335-1338.

Gorin, A. L.; Parker, B. A.; Sachs, R. M. and Wilpon, J. G. (1996). "How May I Help You?", Proc. Interactive Voice Technology for Telecommunications Applications (IVTTA), pp. 57-60.

Hataoka, N.; Kokubo, K.; Obuchi, Y.; Amano, A. (1998). "Development of robust speech recognition middleware on microprocessor", Proc. ICASSP, May, Vol. 2, pp. 837-840.

Hataoka, N.; Kokubo, K.; Obuchi, Y.; Amano, A. (2002). "Compact and robust speech recognition for embedded use on microprocessors", IEEE Workshop on Multimedia Signal Processing, pp. 288-291.

Huggins-Daines, D.; Kumar, M.; Chan, A.; Black, A. W.; Ravishankar, M.; Rudnicky, A. I. (2006). "PocketSphinx: a free, real-time continuous speech recognition system for hand-held devices", Proc. ICASSP, Toulouse.

Itakura, F. (1975). "Minimum prediction residual applied to speech recognition", IEEE Transactions on Acoustics, Speech, Signal Processing, pp.67-72.

Kamm, C. A.; Yang, K.M.; Shamieh, C. R.; Singhal, S. (1994). "Speech recognition issues for directory assistance applications", 2nd IEEE Workshop on Interactive Voice Technology for Telecommunications Applications IVTTA94, May, pp. 15-19, Kyoto.

Kango, R.; Moore, R.; Pu, J. (2002). "Networked smart home appliances - enabling real ubiquitous culture", Proceedings of 5th International Workshop on Networked Appliances, Liverpool.

Kango, R.; Pu, J.; Moore, R. (2002b). "Smart appliances of the future - delivering enhanced product life cycles", The 8th Mechatronics International Forum Conference, University of Twente, Netherlands.

Kryter, K. D. (1995). The Handbook of Hearing and the Effects of Noise, Academic Press.

Lahrmann, A. (1998). "Smart domestic appliances through innovations", 6th International Conference on Microsystems, Potsdam, WE-Verlag, Berlin.

Lee, K. F. (1989). Automatic Speech Recognition: The Development of the Sphinx System, Kluwer Academic Publishers.

Lee, K. F. ; Hon, H. W.; Reddy, D. R. (1990). "An overview of the Sphinx speech recognition system", IEEE Transactions on Acoustics, Speech, Signal Processing, vol.38(1), Jan, pp. 35-45.

Lee, C. H.; Rabiner, L. R.; Peraccini, R.; Wilpon, J. G. (1990). "Acoustic modeling for large vocabulary speech recognition", Computer Speech and Language.

Levy, C.; Linares, G.; Nocera, P.; Bonastre, J. (2004). "Reducing computational and memory cost for cellular phone embedded speech recognition system", Proc. ICASSP, Vol. 5, pp. V309-312, May.

Martin, T. B.; Nelson, A. L.; Zadell, H. J. (1964). "Speech recognition by feature abstraction techniques", Tech. Report AL-TDR-64-176, Air Force Avionics Lab.

McLoughlin, I.; Sharifzadeh, H. R. (2007). "Speech recognition engine adaptions for smart home dialogues", 6th Int. Conference on Information, Communications and Signal Processing, Singapore, December.

McLoughlin, I. (2009). Applied Speech and Audio, Cambridge University Press, Jan.

McTear, M. F. (2004). Spoken Dialogue Technology: Toward The Conversational User Interface, Springer Publications.

Miller, G. A.; Heise, G. A.; Lichten, W. (1951). "The intelligibility of speech as a function of the context of the test materials", Exp. Psychol. Vol. 41, pp. 329-335.

Minker, W.; Bennacef, S. (2004). Speech and Human-Machine Dialog, Kluwer Academic Publishers.

Nakano, M.; Hoshino, A.; Takeuchi, J.; Hasegawa, Y.; Torii, T.; Nakadai, K.; Kato, K.; Tsujino, H. (2006). "A robot that can engage in both task-oriented and non-task-oriented dialogues", 6th IEEE-RAS International Conference on Humanoid Robots, pp. 404-411, December.

Paul, D. B. (1989). "The Lincoln robust continuous speech recognizer," Proc. of ICASSP, vol.1, pp. 449-452.

Phadke, S.; Limaye, R.; Verma, S.; Subramanian, K. (2004). "On design and implementation of an embedded automatic speech recognition system", 17th International Conference on VLSI Design, pp. 127-132.

Prior, S. (2008). "SmartHome system", http://smarthome.geekster.com, last accessed July 2008.

Rabiner, L. R.; Levinson, S. E.; Rosenberg, A. E.; Wilpon, J. G. (1979). "Speaker independent recognition of isolated words using clustering techniques", IEEE Transactions on Acoustics, Speech, Signal Processing, August.

Rabiner, L. R. (1989). "A tutorial on hidden markov models and selected applications in speech recognition", Proc. IEEE, pp. 257-286, February.

Rabiner, L. R. (1994). "Applications of voice processing to telecommunications", In proceedings of the IEEE, Vol. 82, No. 2, pp. 199-228, February.

Ravishankar, M. K. (1996). "Efficient algorithms for speech recognition", Ph.D thesis, Carnegie Mellon University, May.

Roy, D. (1999). "Networks for homes", IEEE Spectrum, December, vol. 36(12), pp. 26-33.

Sakoe, H.; Chiba, S. (1978). "Dynamic programming algorithm optimization for spoken word recognition", IEEE Transactions on Acoustics, Speech, Signal Processing, February, vol.26(1), pp. 43-49.

Sun, H.; Shue, L.; Chen, J. (2004). "Investigations into the relationship between measurable speech quality and speech recognition rate for telephony speech", Proc. ICASSP, May, Vol. 1, pp.1.865-1.868.

Tan, Z. H.; Varga, I. (2008). Automatic Speech Recognition on Mobile Devices and over Communication Networks, Springer Publications, pp. 1-23.

Tanenbaum, A, (1996). Computer Networks, 3rd ed. Upper Saddle River, N.J. London, Prentice Hall.

Velichko, V. M.; Zagoruyko, N. G. (1970). "Automatic recognition of 200 words", International Journal of Man-Machine Studies, June, Vol.2, pp. 223-234.

Wang, Y. M.; Russell, W.; Arora, A.; Jagannathan, R. K. Xu, J. (2000). "Towards dependable home networking: an experience report", Proceedings of the International Conference on Dependable Systems and Networks, p.43.

Weibel, A.; Hanazawa, T.; Hinton, G.; Shikano, K.; Lang, K. (1989). "Phoneme recognition using time-delay neural networks", IEEE Transactions on Acoustics, Speech, Signal Processing, March, Vol.37(3), pp. 328-339.

Wikipedia, (2008). http://en.wikipedia.org/wiki/Semantic_web, last accessed July 2008.

Zue, V.; Glass, J.; Phillips, M.; Seneff, S. (1989). "The MIT summit speech recognition system: a progress report", Proceedings of DARPA Speech and Natural Language Workshop, February, pp. 179-189.

28

Silicon Technologies for Speaker Independent Speech Processing and Recognition Systems in Noisy Environments

Karthikeyan Natarajan[1], Dr.Mala John, Arun Selvaraj[2]
Madras Institute of Technology, Anna University
India

1. Introduction

As the speaker independent speech recognition problem itself is highly computation intensive, the external environment adds to recognition complexity. As per Moore's law, doubling of number of transistors in a chip per year lead to the integration of various architectures in high density chips which lead to the implementation of high complex mixed signal speech systems in FPGA and ASIC technologies. Though several software based speech recognition systems are developed over the years, speech system implementations are yet to unleash the capabilities of silicon technologies. Direct mapped, completely hardware based systems will be highly energy efficient and less flexible but processor based implementation will be less energy efficient and flexible. Software based recognition systems fail to meet the latency requirements of the real time conditions whereas a completely hardware based recognition systems are power intensive. Hence in this case study, a hardware software based co-design is considered for the speech recognition implementation. Sequential algorithms which have been developed need to be modified to suit the parallel hardware systems. Hardware and software based co-design of the isolated word recognition problem will be applicable for low power systems like an AI based robotic system which could use a fixed point arithmetic and hence algorithmic optimizations needed to be considered to suit the actual hardware. Isolated word recognition problem can be split into three stages namely speech analysis, robust processing and final recognition stage. This hardware based speech recognition system is characterized for power and computation efficiency with the following parameters namely vocabulary size, robust speech recognition, speech variability, power and fixed point inefficiencies. This hardware system uses 50Mbps (Max 100Mhz) / 50Mhz NIOS 2 processor with WM8731 audio codec, DRAM controller, I2C controller, Avalon Bus bridge controller, ASIP matrix processor and parallel log Viterbi based hardware module implemented in ALTERA FPGA.

This chapter provides an Introduction to Hidden Markov model based speech Recognition. Relative merits and demerits of conventional Filter bank based feature extraction algorithm via windowed Fourier transform method is compared with a parallel linear predictive coding based CMOS implementation. Detailed description of the HMM based speech

[1] Author is currently working in IBM India Systems and Technology Engineering Labs
[2] Author is currently working in Wipro Technologies, Chennai.

recognition SOC chip is explained in this section. The robust processing step which involves the removal of external unintended noise component from speech signal and the novel Application specific matrix processor for noise removal based on signal subspace based Frobenious norm constrained algorithm are further discussed. This ASIP matrix solver consists of Singular value decomposition unit, QR decomposition unit, matrix bi-diagonilization unit, Levinison-Durbin Toeplitz matrix solver, fast matrix transposition unit based on efficient address generation module. Discussion on word recognition implementation as a parallel 32 bit fixed point 32 state univariate Hidden Markov model based system in ALTERA FPGA is carried out in the final section of this chapter.

1.1 Introduction to HMM based speech recognition system

Speech recognition can be classified into three categories namely Isolated, Connected and Continuous speech recognition systems. In an isolated word recognition system, each word is assumed to be surrounded by silence or background noise. This means that both sides of a word must have no speech input, making definite word boundaries easy to construct. This kind of recognition is mainly used in applications where only a specific digit or a word needs to be identified. Implementation of Isolated word recognition doesn't require any language information and it uses the minimum information about the source speech and has the low recognition accuracy for very large vocabulary. Connected speech (or more correctly 'connected utterances') recognition is similar to isolated word Recognition, but it allows several words/digits to be spoken together with minimal silence period between them. Longer phrases or utterances are therefore possible to be recognized. Continuous speech recognition is method for recognizing spontaneous speech. The system is able to recognize a sequence of connected words, which are not separated by pauses, in a sentence. This mode requires much more computation time and memory, and it is more difficult to operate when compared to isolated word recognition. A speaker-dependent system is a system that recognizes a specific speaker's speech while speaker-independent systems can be used to detect speech by any unspecified speaker. Currently speaker independent systems are modeled using Gaussian Mixture based quantizers which have high recognition accuracy. For speaker independent speech recognition system the training data must be exhaustive, which should incorporate all kinds of speaker variations. It is clear that the smaller the vocabulary size, the higher the recognition accuracy. In an isolated digit recognition system we can achieve higher accuracy by storing finer models of the digits. Further if the vocabulary size is increased there is significant reduction in the computational performance of the system. The training data needs to be generated from the field or the environment where we are planning to implement it.

Isolated Word Recognition problem can be divided into two parts, namely - Front End Processing and Pattern recognition. Typically, the front-end building block includes two modules, data acquisition and feature extraction. In our system we have also implemented the end-point detection and speech enhancement module to make the speech signal more adaptive and robust to the noise. The first stage in any Speech Recognition system is modeling the input speech signal based on certain objective parameters also called the Front End Parameters. Modeling of the input speech signal involves three basic operations spectral modeling, Feature extraction, and parametric transformation (Figure 1). Spectral shaping is the process of converting the speech signal from analog to digital and emphasizing important frequency components in the signal. Noise suppression and speech enhancement module can be added to the Front end processing module which will improve the recognition accuracy.

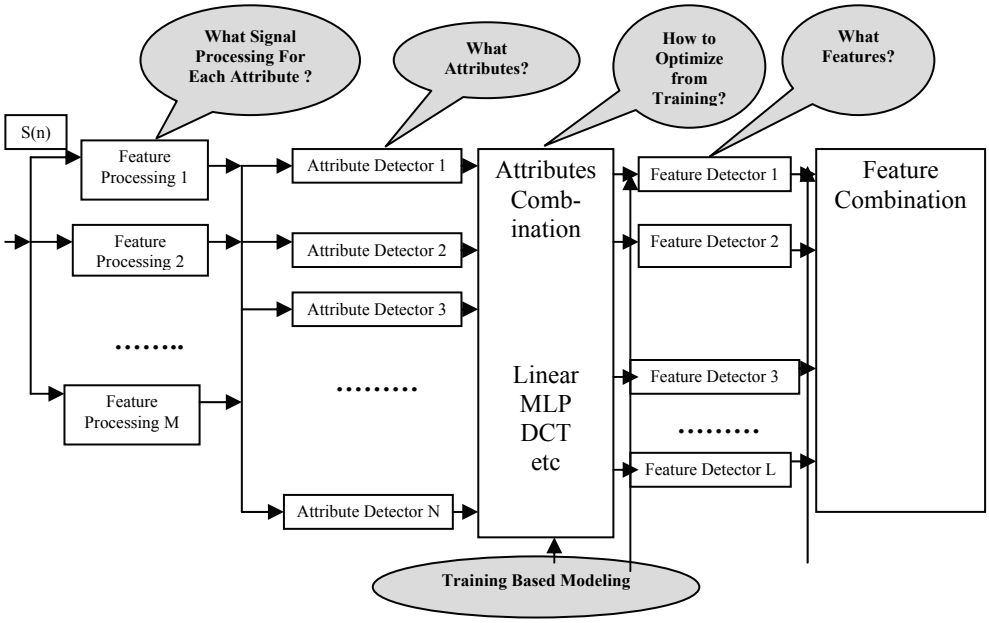


Fig. 1. Components of a speech recognition system

Two major kinds of front end Processing methods are Linear Predicative Coding and Mel Frequency Cepstral Co-efficient. The basic idea behind the linear predictive coding (LPC) analysis is that a speech sample can be approximated as a linear combination of past speech samples. By minimizing the sum of the squared differences (over a finite interval) between the actual speech samples and the linearly predicted ones, a unique set of predictor coefficients is determined. Speech is modeled as the output of linear, time-varying system excited by either quasi-periodic pulses (during voiced speech), or random noise (during unvoiced speech). The linear prediction method provides a robust, reliable, and accurate method for estimating the parameters that characterize the linear time-varying system representing vocal tract.In linear prediction (LP) the signal s (n) is modeled as a linear combination of the previous samples:

$$s(n) = \overleftrightarrow{S}(n + e(n) = \sum_{i=1}^{NlP} a_{LP}(i)s(n-i) + e(n) \tag{1}$$

a_{LP} (i) are the coefficients that need to be decided, N_{LP} is the order of the predictor, i.e. the number of coefficients in the model, and e(n) is the model error, the residual. There exists several methods for calculating the coefficients. The coefficients of the model that approximates the signal within the analysis window (the frame) may be used as features, but usually further processing is applied. Higher the order of the LP Filters used, better will be the model prediction of the signal. A lower order model, on the other hand, captures the trend of the signal, ideally the formants. This gives a smoothened spectrum. The LP coefficients give uniform weighting to the whole spectrum, which is not consistent with the

human auditory system. For voiced regions of speech all pole model of LPC provides a good approximation to the vocal tract spectral envelope. During unvoiced and nasalized regions of speech the LPC model is less effective than voiced region. The computation involved in LPC processing is considerably less than cepstral analysis. Thus the importance of method lies in ability to provide accurate estimates of speech parameters, and in its relative speed. The features derived using cepstral analysis outperforms those that do not use it and that filter bank methods outperform LP methods. Best performance was achieved using MFCCs with Filter bank processing. Even though the CPU computations and memory accesses for MFCC are more, they are less speaker dependent and more speaker Independent. In our implementation we are using Short Time Fourier Transform based MFCC Feature Extraction Method for Front End Processing(Figure 2).

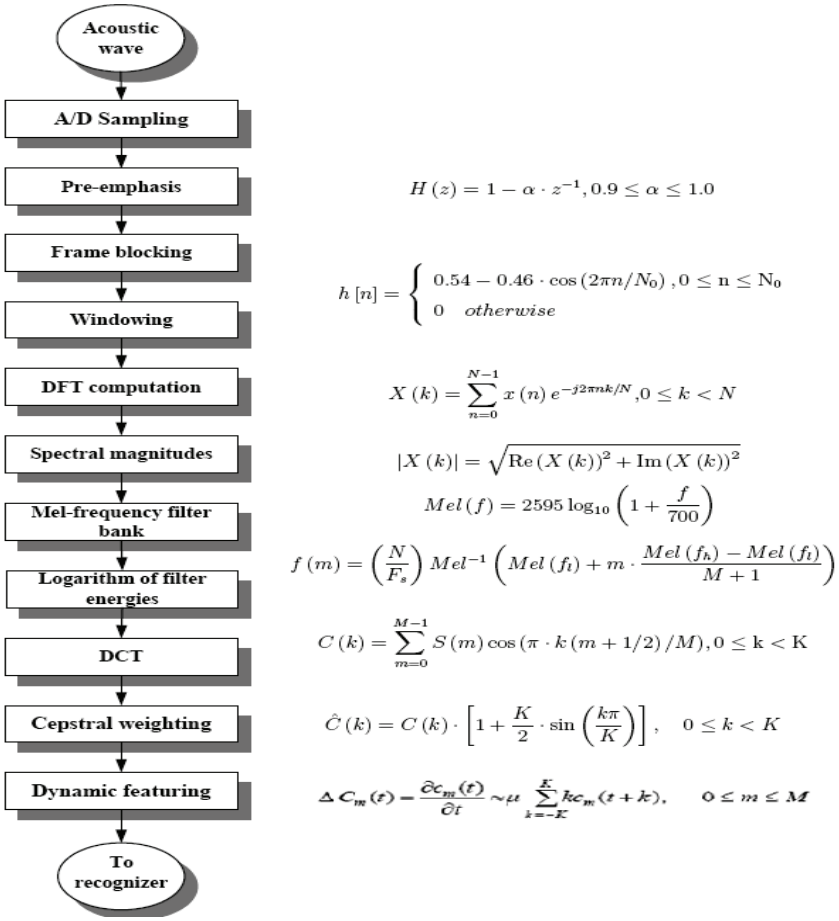


Fig. 2. Flow for Front End processing with feature extraction

We have found that with the hamming window of length 256 the signal can be represented efficiently with consideration to the hardware computational requirements of implementing

a FFT routine. After windowing the speech signal, Discrete Fourier Transform (DFT) is used to transfer these time-domain samples into frequency-domain ones. Direct computation of the DFT requires N^2operations, assuming that the trigonometric functions have been pre-computed. Meanwhile, the FFT algorithm only requires on the order of $N \log_2 N$ operations, so it is widely used for speech processing to transfer speech data from time domain to frequency domain.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} \; 0 <= k < N \tag{2}$$

The Spectral magnitudes were obtained by computing the absolute values of the FFT real and imaginary outputs. The square root is a monotonically increasing function and can be ignored if only the relative sizes of the magnitudes are of interest (ignoring the increased dynamic range).

$$|X(k)| = \sqrt{\mathrm{Re}(X(k))^2 + \mathrm{Im}(X(k))^2} \tag{3}$$

The computation still requires two real multiplications and consumes a lot of latency. A well-known approximation to the absolute value function is given as.

$$\left|A_{\mathrm{re}} + jA_{\mathrm{im}}\right| \approx \left|A_{\mathrm{re}}\right| + \left|A_{\mathrm{im}}\right| \tag{4}$$

A less frequently used approximation is only slightly more complex to implement but offers far better performance (refer table 1).

$$\left|A_{\mathrm{re}} + jA_{\mathrm{im}}\right| \approx \max\left(\left|A_{\mathrm{re}}\right|, \left|A_{\mathrm{im}}\right|\right) + \tfrac{1}{2}\min\left(\left|A_{\mathrm{re}}\right|, \left|A_{\mathrm{im}}\right|\right) \tag{5}$$

The above approximation was considered for the computation of spectral magnitude of the FFT outputs and their spectral magnitudes are taken. Human auditory system is nonlinear in amplitude as well as in frequency. We have taken logarithm to emulate amplitude nonlinearity and Mel filter banks to incorporate frequency nonlinearity. We have used 27 Mel triangular filter banks with 102 coefficients evenly spaced in Mel domain and the cepstral vectors are extracted based on the following equation 6 (refer Figure 3).

$$f(k) = (N/Fs) * Mel^{-1}(Mel(F_{low}) + k * \frac{(Mel(F_{high}) - Mel(F_{low})}{M+1}) \tag{6}$$

$$Mel(f) = 2595 * \log_{10}(1 + \frac{f}{700}) \tag{7}$$

$$Mel^{-1}(f) = 700 * (10^{(\frac{f}{2595})} - 1) \tag{8}$$

$$H_m(k)=\begin{cases}0\cdots\cdots\cdots if\cdots\cdots\cdots k<f(m-1)\\ \dfrac{(k-f(m-1))}{f(m)-f(m-1)}\cdots\cdots\cdots if\cdots\cdots\cdots f(m-1)\le k\le f(m)\\ \dfrac{(f(m+1)-k)}{f(m+1)-f(m)}\cdots\cdots\cdots if\cdots\cdots\cdots f(m)\le k\le f(m+1)\\ 0\cdots\cdots\cdots if\cdots\cdots\cdots k>f(m+1)\end{cases} \tag{9}$$

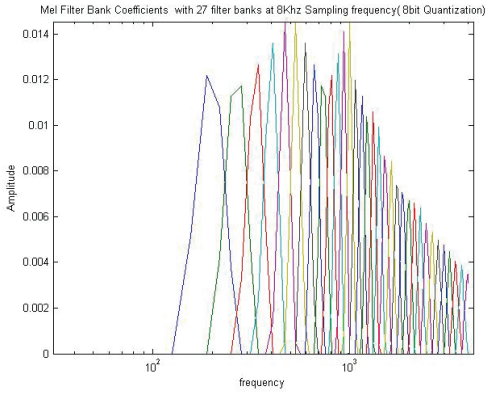


Fig. 3. Mel Filter Bank

The inverse DFT is performed on the output of the filter bank. Since the log power spectrum is symmetric and real, the inverse DFT is reduced to discrete cosine transformation (DCT). This transformation decorrelates features, which leads to using diagonal covariance matrices instead of full covariance matrices while modeling the feature coefficients by linear combinations of Gaussian functions. Therefore complexity and computational cost can be reduced. This is especially useful for speech recognition systems. Since DCT gathers most of the information in the signal to its lower order coefficients, by discarding the higher order coefficients, significant reduction in computational cost can be achieved. Typically the number of coefficients, *K*, for recognition ranges between 8 and 13. The equation is as followingSensitivity of the lower order cepstral coefficients to overall slope and a higher order coefficient to noise has necessitated weighing of the cepstral coefficients by a tapered window to minimize these sensitivities. We have used weighing by a band pass filter of the form. Temporal cepstral derivatives are an improved feature vector for forming the speech frames. They can be used with the cepstral derivative in case the cepstral Coefficients do not give acceptable recognition accuracy. Cepstral representations provide good approximations to the local spectral Properties. Derivatives of cepstral coefficients can be used to describe the dynamic movement of spectrum.In practical applications, the following approximation is used,

$$\Delta C_m(t)=\frac{\partial\, C_m(t)}{\partial t}\approx\left\{\mu*\sum_{k=-K}^{K}k*C_m(t+k)\right\}0\le m\le M \tag{10}$$

Where μ is a normalization factor.

Typical feature vector: (Figure 4):
[E(t) c1(t) c2(t)… cM(t), E(t) , ΔE(t) , Δc1(t) Δc2(t)… ΔΔcM (t-1) ΔΔc1(t) ΔΔc2(t).. ΔΔcM (t-1)]^T
Feature vector consists of both static part and the Dynamic part of the speech signal.

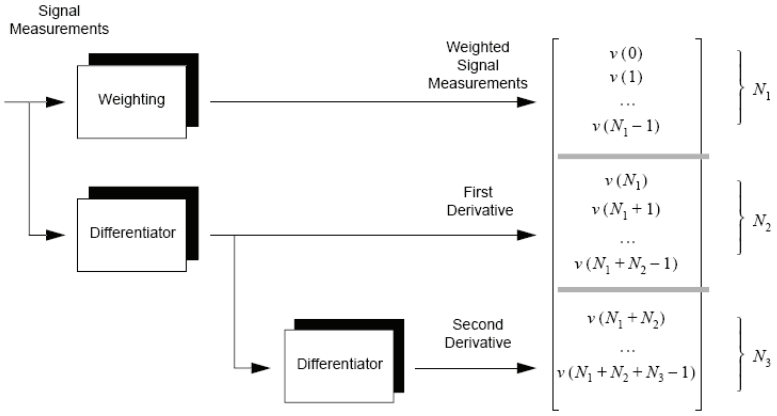


Fig. 4. Representation of Delta and Delta- Delta parameters

2. The hidden Markov models

2.1 The three basic problems of HMM

1. Given the observation sequence O = (o_1 o_2 …o_T), and a model λ=(A, B, π), how do we efficiently compute p(o | λ), the probability f the observation sequence, given the model. This is the "evaluation problem". Using the forward and backward procedure provides solution.
2. Given the observation sequence O = (o_1 o_2 …o_T), and the model λ, how do we choose a corresponding state sequence q=(q_1,q_2…q_T) that is optimal in some sense(i.e. best explains the observation).The Viterbi algorithm provides a solution to find the optimal path.
3. How do we adjust the model parameters λ= (A, B, π) to maximize p(O | λ).This is by far the most difficult problem of HMM. We choose λ= (A, B, π) in such a way that its likelihood, p(O | λ), is locally maximized using an iterative procedure like Baum-Welch method (L. Rabiner 1993).

The base speech recognizer works only with noiseless HMM states and the matrix processor is used as pre conditioning block to generate the noiseless HMM models from the noisy speech vectors. There are three kinds of Hidden Markov models described in the literature namely Discrete HMM, Continuous HMM and Semi-continuous HMM (Vaseghi) in which a Continuous HMM model is used to model the HMM states.A HMM model is characterized by the no of states N, no of distinct observation symbols M, the transition probability matrix A, the initial probability matrix Πi,the output observation probability for a feature x1 in state I , $b_i(x_1)$.

2.2 The log-viterbi algorithm:

1) Initialization:

$$\delta^{(\log)}{}_1(i) = \log a_{1i} + \log b_i(x_1) \tag{11}$$

$$\psi_1(i) = 0 \tag{12}$$

2) Recursion:

$$\delta^{(\log)}{}_{t+1}(j) = \max_{i=2}^{N-1}(\delta^{(\log)}{}_t(i) + l\log a_{ij}) + \log b_j(x_{t+1}) \tag{13}$$

$$\psi_{t+1}(i) = \arg\max_{i=2}^{N-1}(\delta^{(\log)}{}_t(i) + l\log a_{ij}) \tag{14}$$

3) Termination:

$$\log(P(O/\lambda)) = \max_{i=2}^{N-1}(\delta^{(\log)}{}_T(i) + l\log a_{iN}) \tag{15}$$

$$q^T = \arg\max_{i=2}^{N-1}(\delta^{(\log)}{}_T(i) + l\log a_{iN}) \tag{16}$$

4)Backtracking:

$$q_t = \psi_{t+1}(q_{t+1})_{for t=T-1 to 1} \tag{17}$$

The probability of observation vectors , p(O| λ) has to be maximized for different model parameter values which corresponds to HMM models for different words. The implementation of the log likely computation can be done in an efficient way using the Forward and backward procedures as described in (Karthikeyan –ASICON 2007). Since the direct implementation of Viterbi algorithm results in underflow due to very low probability values are multiplied recursively over the speech frame window, logarithmic Viterbi algorithm is implemented which is different from methods given in (Karthikeyan - ASICON 2007). Since the direct implementation of Forward, Backward as well as the Viterbi algorithm results in underflow, we took logarithm on both sides and we have implemented logarithmic versions of the above algorithm. Since the Forward algorithm uses summation which is being replaced by the following conversion in the modified forward algorithm. We have used the modified forward algorithm, backward algorithm as well as viterbi algorithm which is different from the methods given in [6].

2.3 The Baum Welch re-estimation procedure

The third, and by far the most difficult, problem of HMMs is to determine a method to adjust the model parameters (A, B, π) to maximize the probability of the observation sequence given the model. There is no known way to analytically solve for the model, which maximizes the probability of the observation sequence. In fact, given any finite observation sequence as training data, there is no optimal way of estimating the model parameters. We can, however, choose λ = (A, B, π) such that P(O | λ) is locally maximized using an iterative procedure such as the Baum-Welch method. To describe the procedure for re-estimation

(iterative update and improvement) of HMM parameters, we first define ξt(i,j), the probability of being in state Si at time t, and state Sj, at time t+1, given the model and the observation sequence.

In order to use either ML or MAP classification rules, we need to create a model of the probability p(oj) for each of the different possible classes. The PDF can be modeled using a Gaussian distribution. We can create a Gaussian model by just finding the sample Mean, and the sample covariance matrix Ui.

$$\begin{aligned} \mu_i &= \frac{1}{N}\sum_{n=1}^{N} o_n \\ U_i &= \frac{1}{N-1}\sum_{n=1}^{N}(o_n-\mu_i)'(o_n-\mu_i) \end{aligned} \tag{18}$$

$$\mathcal{N}(o;\mu,U) = \frac{1}{\sqrt{(2\pi)^p|U|}}\exp\left(-\frac{1}{2}(o-\mu)U^{-1}(o-\mu)'\right) \tag{19}$$

Probability of being in state S_i at time t, and state S_j at time t+1, given the model and the observation sequence, i.e.

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda). \tag{20}$$

2.4 Covariance selection in speech recognition

The covariance Matrix used in model based speech Recognition problem which uses N state Univariate Gaussian HMM modeling with M dimensional features can be considered in the following ways. The following 39 dimensional feature vectors are considered for designing the continuous HMM based speech recognizer.

[c1(t) c2(t)… cM(t), Δc1(t) Δc2(t)… ΔΔcM (t-1) ΔΔc1(t) ΔΔc2(t).. ΔΔcM (t-1), E(t) , ΔE(t)]^T

Where $\Delta C_m(t)$, $\Delta\Delta C_m(t)$ can be represented as below.

$$\Delta C_m(t) = \frac{\partial\, C_m(t)}{\partial t} \approx \left\{\mu * \sum_{k=-K}^{K} k * C_m(t+k)\right\} 0 \le m \le M \tag{21}$$

$$\Delta\Delta C_m(t) = \frac{\partial\, \Delta C_m(t)}{\partial t} \approx \left\{\mu * \sum_{k=-K}^{K} k * \Delta C_m(t+k)\right\} 0 \le m \le M \tag{22}$$

i. Complete covariance matrix (distance measure: Mahalanobis distance measure) Complete covariance matrix when considered results in very high implementation complexity and cannot be easily achieved with the existing hardware resources (S.Yozhizawa – 2006).

ii. The second method is to do covariance parameter tying (Pihl – 1996). In this a method a common parameter is considered for all the states and other statistical characteristics are considered different. For instance, use of common covariance matrix for all the

clusters obtained during GMM block quantization and considering mean, no of observation output different for each state.

iii. A still modified version of the above mentioned method having block covariance matrices instead of having complete covariance matrix can be considered for implementation which has complex implementation in hardware. Our assumption that the covariance are block diagonal is valid since the use of an orthogonal transform like DCT decorrelates the cepstral vectors. The correlation only exists between the time difference cepstral vectors, delta cepstral vectors, and the delta – delta cepstral vectors. So we can construct the covariance matrix as three element block diagonal matrix for which the inverse matrix can be easily found using Singular value decomposition.

iv. The last method is to consider the covariance matrix to be diagonal which yields the simplest hardware architecture. The inverse diagonal values are stored in memory locations and only multiply operations are performed and this method is computationally less intensive. Present hardware based recognizers implement this method due to its low complexity. But this method will produce significant errors and degrade the recognition performance of the system as it is doesn't efficiently represent the correlation introduced by the Vector quantizer. Earlier proposed implementations were based on this method only (Karthikeyan – ASICON 2007).Where E(.,.) represents the statistical Expectation operation on the cepstral vectors.

$$R=\begin{pmatrix} \begin{pmatrix} E(c_1,c_1) & E(c_1,\Delta c_1) & E(c_1,\Delta\Delta c_1) \\ E(\Delta c_1,c_1) & E(\Delta c_1,\Delta c_1) & E(\Delta c_1,\Delta\Delta c_1) \\ E(\Delta\Delta c_1,c_1) & E(\Delta\Delta c_1,\Delta c_1) & E(\Delta\Delta c_1,\Delta\Delta c_1) \end{pmatrix} & 0 & 0 \\ 0 & \begin{pmatrix} E(c_2,c_2) & E(c_2,\Delta c_2) & E(c_2,\Delta\Delta c_2) \\ E(\Delta c_2,c_2) & E(\Delta c_2,\Delta c_2) & E(\Delta c_2,\Delta\Delta c_2) \\ E(\Delta\Delta c_2,c_2) & E(\Delta\Delta c_2,\Delta c_2) & E(\Delta\Delta c_2,\Delta\Delta c_2) \end{pmatrix} & 0 \\ 0 & 0 & \begin{pmatrix} E(c_3,c_3) & E(c_3,\Delta c_3) & E(c_3,\Delta\Delta c_3) \\ E(\Delta c_3,c_3) & E(\Delta c_3,\Delta c_3) & E(\Delta c_3,\Delta\Delta c_3) \\ E(\Delta\Delta c_3,c_3) & E(\Delta\Delta c_3,\Delta c_3) & E(\Delta\Delta c_3,\Delta\Delta c_3) \end{pmatrix} \end{pmatrix} \quad (23)$$

Earlier we have discussed the influence of the covariance selection on the performance of the recognition and it directly influences the word error rate. Earlier implementation considers completely diagonal co variances which cause drastic errors as we have introduced correlation into the feature vectors through vector quantization as well as dynamic feature vector set. Hence we can consider the feature vectors to be correlated only among the two dynamic features set delta and delta delta feature vectors the static features. Hence we can assume the correlation matrix to be block diagonal and hence the inverse of such a matrix can be easily obtained by linear equation solvers. Computation of the Singular Value Decomposition of a matrix A can be accelerated by the parallel two sided jacobi method with some pre-processing steps which would concentrate the Frobenius norm near the diagonal. Such approach would help noise reduction is great way as the noise sub-space is computed with Frobenius norm constraints. Such a concentration should hopefully lead to fewer outer parallel iteration steps needed for the convergence of the entire algorithm.However the gain in speed as measured by total parallel execution time depends decisively on how efficient is the implementation of the distributed QR and LQ factorizations on a given parallel architecture.

3. Speech recognition architecture

3.1 NIOS embedded processor based system design

NIOS 2 is a soft Processor which can be realized in any of the Altera's FPGA Development kits. It is based on a 32-bit RISC architecture and is a natural choice in projects where CPU performance is essential.The NIOS processor can be run at different frequencies based on which the Computational capability of the processor can be choosen. Nios Processor is available in three different speed grades and can be extended with additional coprocessors, instruction sets, and so forth. By doing so, it is possible to develop a large part of the system on an ordinary PC running Windows or any variant of UNIX. By simulating IP cores (firmware modules) as software objects, a system can be developed to an advanced state before it needs to be tested on the actual target. Another benefit of this approach is that it allows concurrent development of multiple projects on a single target. The NIOS processor is a 32-bit Harvard Reduced Instruction Set Computer (RISC)architecture optimized for implementation in Altera FPGAs with separate 32-bit instruction and data buses running at full speed to execute programs and access data from both on-chip and external memory at the same time. Nios Processor has got 32 32bit general purpose registers and 16 32bit control registers, an Arithmetic Logic Unit (ALU),Exception Unit, Instruction cache and Data Cache, Hardware multiply and Hardware divide, a barrel shifter unit and 32 software interrupts. This flexibility allows the user to balance the required performance of the target application against the logic area cost of the soft processor. NIOS processor does not separate between data accesses to I/O and memory (i.e. it uses memory mapped I/O). All the system peripherals of Altera are connected through a system bus called Avalon. for sophisticated SOPC environment or the basic environment. The stack convention used in Nios processor starts from a higher memory location and grows downward to lower memory locations when items are pushed onto a stack with a function call. Items are popped off the stack the reverse order they were put on; item at the lowest memory location of the stack goes first and etc (NIOS 2006). Nios processor also supports reset, interrupt, user exception, and break and hardware exceptions. The processor will only react to interrupts if the Interrupt Enable (IE) bit in the Machine Status Register (MSR) is set to 1. On an interrupt the instruction in the execution stage will complete, one has to manually enable the interrupt enable bit for that particular device.

Writing software to control the NIOS processor must be done in C/C++ language. The NIOS tool has got gnu based have built in C/C++ compilers and debugger to generate the necessary machine code for the NIOS processor (Agarwal 2001). NIOS Processor supports word (32 bits), half-word (16 bits), and byte accesses to data memory. Data accesses must be aligned (i.e. word accesses must be on word boundaries, half-word on half-word boundaries), unless the processor is configured to support unaligned exceptions. All instruction accesses must be word aligned.

Avalon Bus system is a simple yet extremely powerful bus system which allows any no of Bus masters to be added simultaneously and offers excellent arbitration capabilities with wait cycles. It also supports a unique kind of hardware software interface called custom instruction which acts as a hardware mapped instruction to the NIOS processor (Avalon 2006). We can also accelerate the software function in NIOS processor through a technology called Custom instruction which is unique to NIOS based system. Nearly 256 custom instructions different cycle times can be integrated into the design to accelerate the underlying software. Compared to complete software performance, a system with hardware

acceleration improves 20X performance improvement. Our design utilizes this custom instruction feature (Avalon 2006). NIOS processor supports many software IP-cores such as Timer, Programmable counters, Ethernet controller, DRAM controller, Flash controller, User logic components,PLLs,Hardware Mutex, LCD controller etc. NIOS Timers can be used to compute the execution time of a software routine or used to produce trigger at regular intervals so as to signal some of the hardware peripherals. Hardware IP cores can be connected to the system in two different ways. The hardware component can be configured as Avalon Custom instruction component and the processor can access the hardware as though it being an instruction. NIOS processor supports four different kinds of custom instruction technology namely combinational; Multi cycle, Extended and Internal Register file based custom instruction. Custom instruction module can also be connected to the Avalon Bus so that one can connect some of the custom instruction signals to external signals not related to the processor signals. Hardware IP core can also be interfaced to the NIOS system through the Avalon slave or Master Interface. Avalon Slave devices can have interrupts and they request the service of the processor through the interrupts. These interrupts can be prioritized manually.

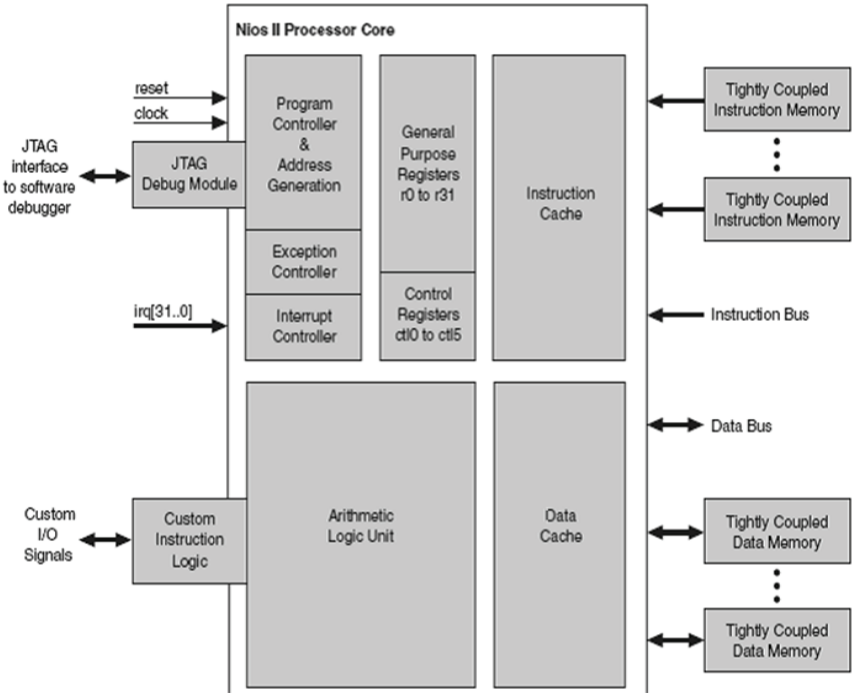


Fig. 5. NIOS Architecture

4. Design implementation using fixed point architecture

4.1 Finite word length effects

All DSP based designs strongly depend on the floating point to fixed point conversion stage as the DSP algorithm may not be implementable in floating point form. Fixed pint analysis

of the system is extremely important to understand the nonlinear nature of the quantization characteristics. This leads to certain constraints and assumptions on quantization errors: for example that the word-length of all signals is the same, that quantization is performed after multiplication, and that the word-length before quantization is much greater than that following Quantization (Meng 2004). Error signals, assumed to be uniformly distributed, with a white spectrum and uncorrelated, are added whenever a truncation occurs. This approximate model has served very well, since quantization error power is dramatically affected by word-length in a uniform word-length structure, decreasing at approximately 6dB per bit. This means that it is not necessary to have highly accurate models of quantization error power in order to predict the required signal width. In a multiple word-length system realization, the implementation error power may be adjusted much more finely, and so the resulting implementation tends to be more sensitive to errors in estimation. Signal-to-noise ratio (SNR), sometimes referred to as signal-to-quantization noise ratio (SQNR), is The ratio of the output power resulting from an infinite precision implementation to the fixed-point error power of a specific implementation defines the signal-to-noise ratio In order to predict the quantization effect of a particular word-length and scaling annotation, it is necessary to propagate the word-length values and scaling from the inputs of each atomic operation to the operation output (Haykin 1992). The precision of the output not only depends on the binary precision of the inputs, it also depends on the algorithm to be implemented. For example the fixed point implementation of complex FFT algorithm decreases 0.5 bit precision for each stage of computation (Baese 2005). So for large FFT lengths more bits of precision are lost. The Feature extraction stage was implemented in Nios Processor with fixed precession inputs. The following plots describe the fixed point characteristics of the algorithm.

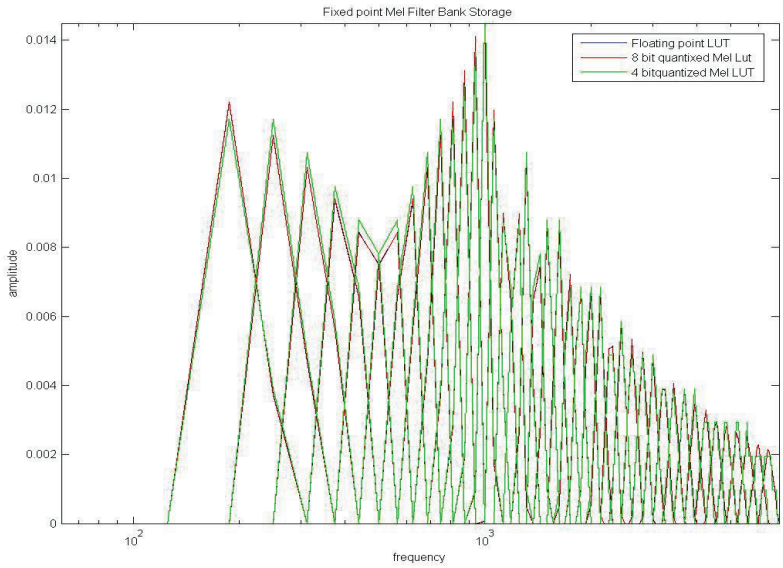


Fig. 6. Fixed point MFCC implementation

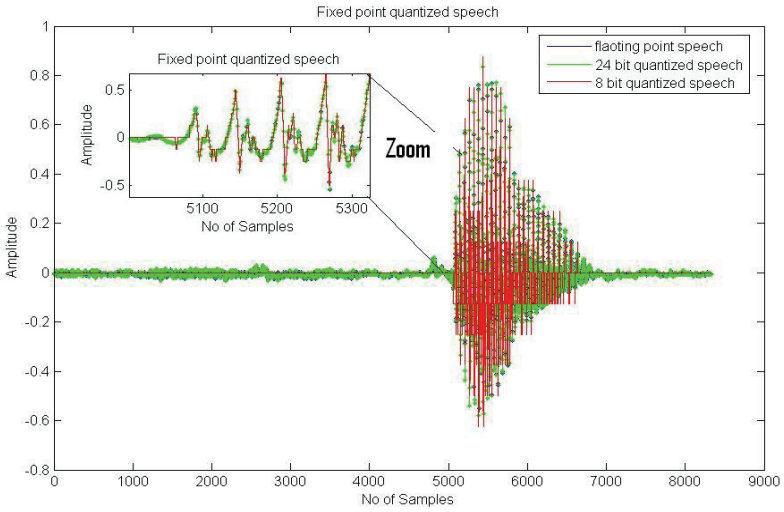


Fig. 8. Fixed point Speech input charactersistics

4.2 Flexibility

The recognition system must be able to operate under a variety of conditions (Vaseghi 2004). The signal to noise ratio may vary significantly, the word may be stretched too long or too short, some of the states may be skipped, noise content may be high and we are forced to model the noise HMM and subtract it from the actual speech HMM (Hermus 2007).The receiver must incorporate enough programmable parameters to be reconfigurable to take best advantage of each situation (Press 1992). We applied signal subspace based noise reduction algorithm based on Singular Value decomposition to reduce the noise characteristics of the speech signal (Hemkumar 1991).

4.3 FCTSVD algorithm (Abut 2005):

1. Estimate the noise vectors W from silence periods in the observed speech signal.
2. Form the Hankel matrix Hy from the observed speech signal.
3. Compute SVD of Hy.
4. Initialize the oreder of retained singular values of Hx−
5. Let S=S+1 and reconstruct the estimated matrix of Hx, Hx− using the first s eigen values.
6. Compute Frobenius Constrained norm metric and error is less than 0.0098 else goto 5.

4.4 Scaling:

As the number of frames in speech increases, the values of the variable the formal algorithm saturates whereas the log-Viterbi algorithm used in this hardware does not suffer this problem as the implementation involves only additions rather than multiplication.

4.5 Initial estimates of HMM parameters:

Uniform initial estimates were used for A and Pi Matrices. However B matrix cannot be initialized with random values as it has more influence in convergence of the Baum-Welsch

algorithm. Since continuous hidden markov models are used, the initial estimates of B, Mean, and Variance are obtained using segmental K-means algorithm..

5. Project modules:

5.1 Main modules:

1. First module is concerned with signal analysis and feature extraction (FRONT END PROCESSING → SOFTWARE EXECUTED IN NIOS 2 PROCESSOR).
2. The next module generates the values of parameters required for comparing the test speech signal with the reference signal values .Training phase and this step should be robust since it directly determines the accuracy and the application where the system is to be deployed. (TRAINING – OFFLINE DONE IN MATLAB-refer Figure 13).
3. Maximum Likelihood based word recognition (PARALLEL HARDWARE).

5.2 Supporting hardware modules:

1. Audio Codec Configuration unit based on I2C controller (MPU 2 AUDIO CODEC CONFIGURATION).
2. Custom Avalon Master Module for Audio Codec data Retrieval with integrated SRAM memory controller.
3. Custom Speech controller for hardware recognition part with efficient mode management unit based on FSMs.
4. The speech Controller has got the following modules built in:
5. Viterbi based Speech Recognition unit with memory controllers for Model parameter RAMS.
   a. Input Frame buffers for feature storage with memory controller for feature storage RAM.
   b. Output Frame buffer for Model output storage
   c. Efficient mode management unit to switch between various modes of operations using FSMs.
   d. LED Display unit to finally display the results.
6. Custom Singular Value Decomposition unit.

Software Modules	Hardware Modules interfaced to Avalon	Custom Instructions
✓ Voice Activity Detection ✓ FFT based feature extraction ✓ Mel Filter Banks ✓ Cepstral Weighing ✓ Software Back-substitution	✓ Audio Serial 2 Parallel Module(Avalon Master) ❖ Input Frame buffers ❖ Output Frame Buffers ❖ Speech Recognition Mode controller top ❖ for SVD Speech Recognition module with Viterbi.	• Max comparator for magnitude computation in FFT

Table 2. Isolated Word Recognition System Hardware/ Software Partition

- **Operating frequency of green = 18.432MHz**
- **Operating frequency of red= 12.5MHz**

✓ **Operating frequency of 50 MHz**

Audio codec is configured via I2C interface. Two signals I2C_clock and I2C_data are used to configure the internal registers of the WM8371. The WM8731 is a WRITE-ONLY device; any requests to read are ignored. Device is configured by writing data to internal registers. The internal registers are configured by transferring data and address of the internal registers serially through I2C_data pin. Clock signal is applied to the I2C_clk pin. Clock signal can be generated in two modes of operation namely USB/Normal mode master clock (AUD_XCLK, from which AUD_BCLK is generated).USB mode must have a FIXED AUD_XCLK of 12MHz, which can be easily obtained from a PLL in the SOPC system. Normal mode requires AUD_XCLK clocks of either 12.288MHz (8kHz, 32kHz, 48kHz, 96kHz) or 11.2896MHz (44.1kHz, 88.2kHz).This implementation utilizes Normal mode of clock generation at 18.432MHz. Transfer is initiated by pulling MPU_DATA low while MPU_CLK is high. The data format of the configuration of a particular internal register has got 3-bytes.

- Byte 1: {ADDR[6..0],0} →ADDR[6..0] is DEVICE ADDRESS, which is ALWAYS 0x34
- Last bit is R/W bit, which is always 0 (write,) since WM8731 is write-only
- Byte 2: {REG[6..0],DATA[8]} →REG[6..0] is 7-bit register address, DATA[8] is MSB of 9-bit DATA
- Byte 3: DATA[7..0] → Lower 8 bits of 9-bit DATA
- MPU_DATA is driven low by the CODEC between bytes as confirmation

The following operations needed to be done to make the device operate in the intended mode of operation:

- Reset device: Write 0x00 to AUDIO_RESET
- Power up device: Write '0' to WM8731_POWER_DOWN_CTL.7 bit
- Disable Line IN -> Line OUT bypass, select MIC_IN (AUDIO_ANALOG_PATH_CTL Reg)
- Turn on MASTER mode: AUDIO_INTERFACE_FMT

5.3 How this hardware system works:

The steps involved in implementing this system consist of the following steps given below:

Step 1: Audio Codec is configured via CPU 2 I2C interface with the following specifications.

- ✓ **WM8731_POWER_DOWN_CTL** is used to power up the device**.**
- ✓ **WM8731_ANALOG_PATH_CTL** Register is set to 16'h08FD to enable the MIC in facility.
- ✓ **WM8731_SAMPLING_CTL** Register is set to 16'h100E to fix the audio codec in NORMAL MODE with ADC sampling frequency of 8 KHz. Codec operating frequency is 18.432MHz

Step2: The serial input bit stream is converted in parallel data using a custom Avalon Master interface and is stored in SRAM module. The storage of audio will be interrupted by a external user controlled switch to start the processing step.

Step3: This switch will induce an interrupt signal present in the speech recognition module (AVALON SLAVE CONFIGURED) to start the feature processing of NIOS processor.

Step4: In software the speech start and end points are detected, we perform windowing (Hamming Window).

Step5: We use short time Fourier analysis on the speech signal, since speech is a Quasi-periodic signal we need to use STFT. We have used a window of duration 30ms with an overlap of 10ms.

Step6: Evaluate the distance between the speech signals and do clustering using the Gaussian Mixture based Block quantizer based on Mahalanobis distance and clustering is performed.
Step7: The features are extracted and stored in the **INPUT FRAME BUFFER** of the Speech Recognition module.
Step8: Steps 1 to 6 will continue until the end of frame is detected by the hardware module.
Step9: Starting of speech recognition in hardware and finally the results are populated and displayed in LED. Each stage output is stored in **OUTPUT FRAME BUFFER** and final recognition is done.

5.5 Implementation of continuous hidden Markov model:

Our architecture concentrates on the three major issues Power, Memory access (Throughput) and vocabulary size. There is always a trade off existing between the operating frequency and the recognition vocabulary, word accuracy, noise suppression etc. This is a word HMM based architecture which uses continuous HMM for the implementation.
Two essential steps in the recognition algorithm are:
1. Output probability calculation.
2. Log VITERBI implementation.

→Output Probability calculation is the computationally intensive process as we need to do lots of multiplies and Add operations.
→Viterbi Algorithm is also implemented as a parallel processing block for faster recognition.

5.6 Hardware design:

Our architecture (Fig 11) concentrates on the three major issues Power, Memory access (Throughput) and vocabulary size. There is always a trade off existing between the operating frequency and the recognition vocabulary, word accuracy, noise suppression etc (Pihl 1996). This is a word HMM based architecture which uses continuous HMM for the implementation (Cho 2002).
Two essential steps in the recognition algorithm are:
1. Output probability calculation.
2. Log Viterbi implementation(as in fig 12).

5.7 Modes of operation:

We can operate the system in two modes:
1. Small vocabulary mode
2. Large vocabulary mode

5.8 Small vocabulary mode:

Startup Sequence:
1. *Reset* = 0, sw0= 0, sw1=0
2. Audio is stored in SRAM by custom master Avalon interface.
3. When the user presses switch0 the Avalon master stops storing samples.
4. Speech controller interrupts processor for features after sw0 is pressed.
5. Processor starts processing the samples to extract features and once is complete activates done signal of Speech controller.

6. Load the models in the Model Rams.
7. Perform Viterbi and output probability computation (refer Figure 9 , Figure 10)
8. Store the results of word I in the Output Frame Buffers.
9. repeat the step until all word are done
10. Generate complete word signal
11. Compare the results and show the word spelt in LED.

5.9 Large vocabulary mode:

1. *Reset* = 0, sw0= 0, sw1=1
2. Audio is stored in SRAM by custom master Avalon interface.
3. Speech controller interrupts processor for features after the End of VAD is received.
4. Processor will load the models of 10 words from external memory to the INPUT FRAME BUFFERS OF SPEECH RECOGNITION MODULE.
5. Perform Viterbi and output probability computation.
6. Store the results of word I in the Output Frame Buffers.
7. Repeat the step until all word is done.
8. Generate complete word signal.
9. Compare the results and show the word spelt in LED.

Software algorithm in Floating point C		Average no of cycles to execute(ps)
Convolution		49484981.2
VAD		188143.2
Multiplication		1159.5
FFT		429496674.2
Twiddle Factors		44562.1
Magnitude	Actual (equation 2.8)	9115.8
	Approximation1(equation 2.9)	2545.4
	Approximation1(equation 2.10)	962.8
	Total computation	1224.1

Table 1. Software Execution Cycles of the Feature Extraction Algorithm

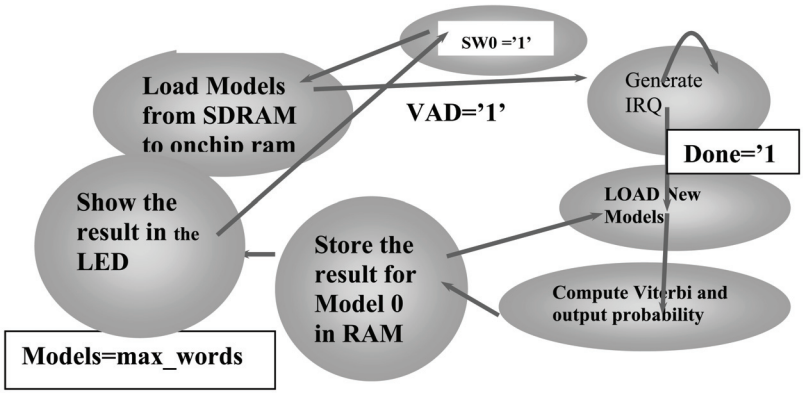


Fig. 9. FSM for main speech Recognition module.

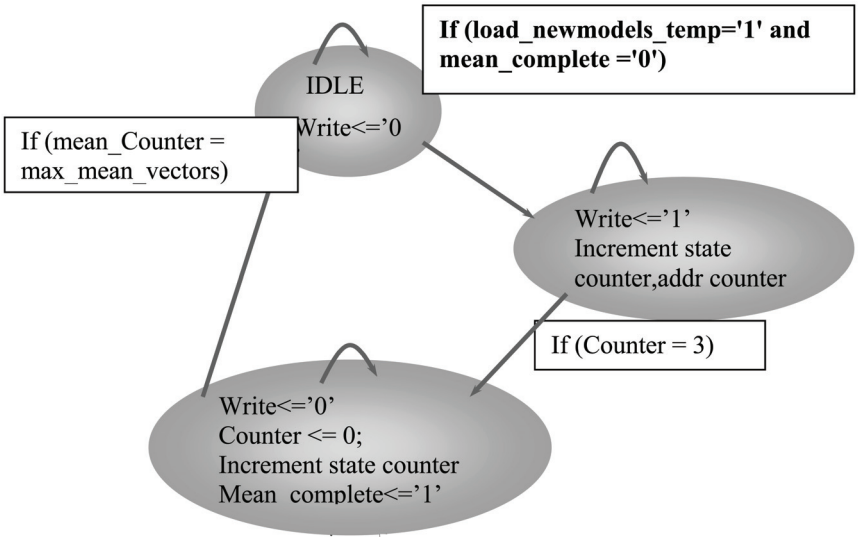


Fig. 10. FSM for memory access of (mean/Covariance Rom/transmit/initial) RAMs.

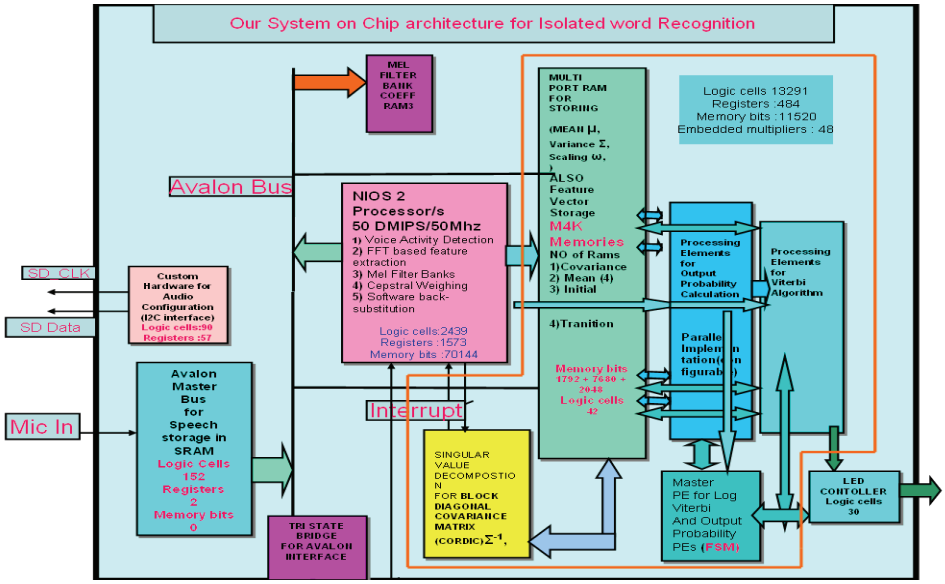


Fig. 11. The speech recognition hardware architecture

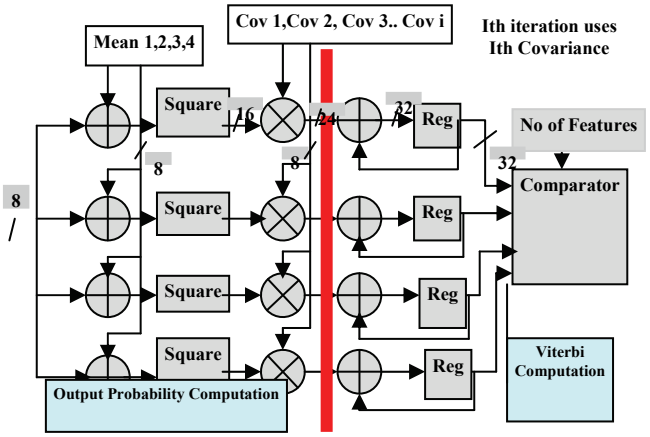


Figure 12. Pipelined-parallel Log likelihood unit

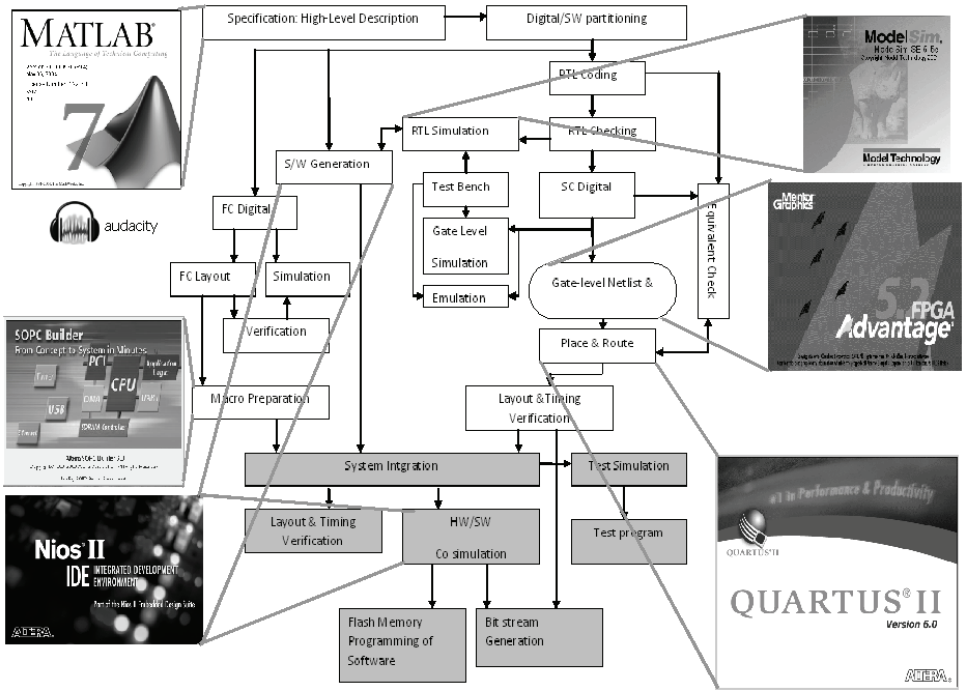


Fig. 13. Hardware/Software flow of the IWR system Development

The following table depicts the low hardware requirement for the Filter bank based feature extraction unit which is implemented as a hardware-software combined block compared to a fully hardware LPC block and hence this implementation has very low power consumption.

Hardware Requirements	Hardware LPC Implementation	Nios 2 Processor system Requirements	SOC based IWR with Nios processor(parallel)
Total logic elements	29799	4900	18393
Total registers	221	3069	3704
Embedded Multiplier 9-bit elements	52	4	52
Total memory bits	0	79360	90880

Table 3. Comparison of Hardware Logic requirements for2 different Front End processing unit

The hardware is interfaced to software unit as a custom instruction in Altera FPGA and the recognition Viterbi algorithm is implemented as a parallel hardware as shown in figure .The overall chip architecture for the speech recognition which does noise robust processing is shown in fig along with the ALTERA FPGA resource utilization values.

6. Application of matrix processor to speech recognition

6.1 Signal subspace based speech enhancement

From the noisy speech samples, H_y is constructed from a frame of noisy speech vector[y(0),y(1)...y(L-1)]^T.The relationship between noisy speech and clean speech vectors can be denoted as $H_y = H_x + H_n$ where H_n represents noisy speech vectors which are estimated during voice activity detection process [1]. SVD of a real matrix can be divided in matrix form as,

$$Hy = [U_{y1} \; U_{y2}] \begin{pmatrix} \Sigma_{y1} & 0 \\ 0 & \Sigma_{y2} \end{pmatrix} \begin{bmatrix} V_{y1}{}^T \\ V_{y2}{}^T \end{bmatrix} \tag{24}$$

If the noise vectors are considered white then the smallest eigen value of Σ_{y1} is significantly greater than largest eigen value of Σ_{y2}. One can use of the SVD reduction techniques such as Thin SVD,Truncated SVD ,Thick SVD for signal space reduction and this implementation considers major p eigen values for signal subspace reduction based speech enhancement[5].

Frobenius norm constraint

Reconstructed signal is represented as a linear combination of the eigen vectors corresponding to the most significant eigen values and the dimensions of the hankel matrix is considered based on the Frobenius norm based performance criteria.

$$\overline{H_x} = U_{y1} \Sigma_{y1}{}^{-1} (\Sigma_{y1}{}^2 - \sigma_w{}^2 I) V_{y1}{}^T \tag{25}$$

$$\phi^{(s)} = \left\| H_y \right\|_F^{\;2} - \left\| H_x{}^{(s)} \right\|_F^{\;2} - \left\| H_W \right\|_F^{\;2} \tag{26}$$

Where $\phi^{(s)}$ represents the error associated with speech enhancement considering most p significant Eigen values and the suffix F indicates the Frobenius norm constraint of the

Hankel matrix $\left\|H_x^{(s)}\right\|_F^2$ represents the frobenius norm of the reconstructed clean speech matrix with p dominant eigen values and $\left\|H_W\right\|_F^2$ represents the Frobenius norm of the noise which is computed during Voice Activity Detection stage. Though progressive implementation of SVD helps to find the optimum p eigen values, real time implementation of such a system is complex intems of computational complexity and latency. Hence the approximate number of dominant eigen values is found to be 12 for speech samples present in AN4 database using MATLAB and is used for this speech enhancement application.

7. The matrix processor

This matrix processor implements three basic Matrix operations namely Addition/Subtraction, Fast Matrix Multiplication and fast Matrix Transposition using efficient Address generation unit in $O(n\log_2 n)$ operations and complex Matrix operations namely Singular Value Decomposition, QR decomposition, Matrix Bi diagonalization and it also implements matrix inversion for toeplitz matrices through Levinison-Durbin Algorithm.

7.1 Matrix multiplication

Matrix multiplication is done using Strassens algorithm for matrix multiplication which is a superior algorithm for matrix multiplication of higher order as the computational complexity is $O(n^{2.81})$ compared to the normal row by row multiplication which requires $O(n^3)$[14]. Any matrix A can be sub divided into block matrices and the product C=A*B can be implemented using Stassen's algorithm at block level.

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \tag{27}$$

The number of multiplications involved in strassen algorithm of block matrix multiplication is $(\frac{7}{8}n + \frac{3}{2})n^2$ and in strassens algorithm the number of multiplications decreases as n increases whereas the number of addition remains same[5].

7.2 Concept of orthogonal matrix:

If vector product of columns of a matrix yields then the matrix is called as Orthogonal matrix. Thus for an orthogonal matrix inverse can be computed by simply taking the transpose of the matrix. The eigewn values of a matrix can be classified as r nonzero eigenvalues and n - r zero eigenvalues. It is common convention to order the eigenvalues so that

$$\lambda_{1(\max)} \geq \lambda_2 \geq \ldots\ldots\ldots \geq \lambda_{r(\min nonzero)} \gg \lambda_{r+1} > \lambda_n$$

that $\lambda 1$ is the largest, with the remaining nonzero eigenvalues arranged in descending order, followed by n - r zero eigenvalues. Note that if A is full rank, then r = n and there are no zero eigenvalues. The quantity λ n is the eigenvalue with the lowest value.

7.2 Givens rotation

Our Hardware implementation of QR decomposition and Singular value decomposition are based on Gentleman kung parallel systolic architecture which uses O(m+n) complexity as $A \in R^{mxn}$ matrices where as the other implementations require $3n^2(m-\frac{n}{3})$ FLOPS to compute QR decomposition (Press 1992).Two kinds of Processing Elements were used in the design of SVD,QR decomposition and matrix bi-diagonalization blocks namely Vectoring Mode CORDIC Processing Element and Rotation Mode CORDIC Processing Element. Since any orthogonal matrix can be resolved into several rotation matrices using Euler-Brauer Resolutions, SVD factorization using givens rotations suits hardware implementation (refer Figure 14).

7.3 The QR-decomposition

QR decomposition of a matrix yields an unitary matrix Q whose columns are orthogonal to each other and an upper triangular matrix *R*. Some characteristics like the l_2-norm of a vector remain unaltered after QR decomposition. Householder reduction is one numerically stable method for the computation of QR decomposition. If a vector lies in the orthogonal complement subspace then its range of vector space is zero and hence it can be used to nullify some of the elements in a matrix. In the linear equation, elimination of one variable leads to a reduced space which can be done by choosing a vector so that the reflection of it in span of a particular vector space lines up with the x-axis. Q_{pq} is an orthonormal matrix which on recursively operated on the data matrix *A*, results in triangular matrix *A1* as $A_1 = Q_{12}Q_{13}Q_{14}\ldots\ldots A^T$. This processing matrix is post multiplied on the data matrix *A* using the Givens rotation algorithm mentioned as above. One rotation premultiplication by Qpq exists for every element to be eliminated.□

$$Q_{pq} = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots \begin{pmatrix} \cos_p\theta & \sin_q\theta \\ -\sin_p\theta & \cos_q\theta \end{pmatrix} & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \tag{28}$$

The following conditions needs to be met to eliminate the row elements :

1. If a matrix A is to be orthonormal, then each J must be orthonormal.(Because the product of orthonormal matrices is orthomonal).
2. The (I,k)th element JA must be zero.

First condition is statisfied by the following relation given below.

$$Q^TQ = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots \begin{pmatrix} \cos_p\theta & -\sin_q\theta \\ \sin_p\theta & \cos_q\theta \end{pmatrix} & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots \begin{pmatrix} \cos_p\theta & \sin_q\theta \\ -\sin_p\theta & \cos_q\theta \end{pmatrix} & \vdots \\ 0 & \cdots & 1 \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots \begin{pmatrix} 1 & \\ 0 & 1 \end{pmatrix} & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \tag{29}$$

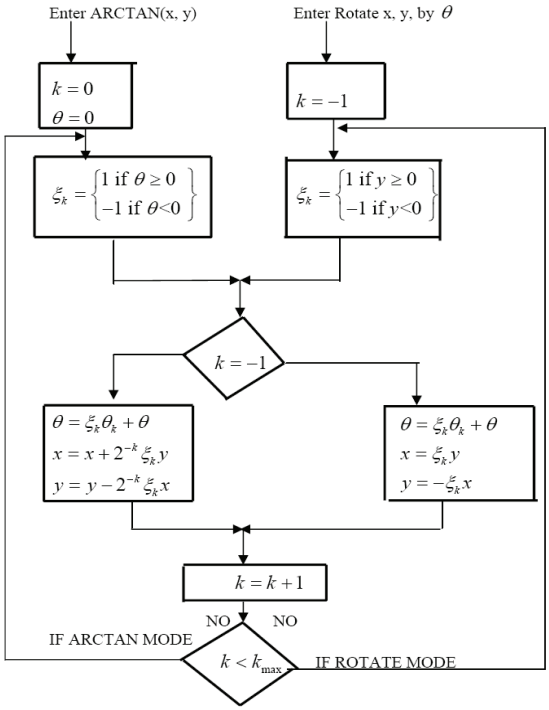
The second constraint is given by ,

$$QA = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots \begin{pmatrix} \cos_p \theta & \sin_q \theta \\ -\sin_p \theta & \cos_q \theta \end{pmatrix} & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots \begin{pmatrix} a_{kk} & a_{ki} \\ a_{ik} & a_{ii} \end{pmatrix} & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \tag{30}$$

which evaluates the following equations as,

$$\begin{aligned} -\sin_p \theta * a_{kk} + \cos_q \theta * a_{ik} &= 0 \\ \tan \ \theta &= \frac{a_{ik}}{a_{kk}} \end{aligned} \tag{31}$$

$$\begin{aligned} \sin_p \theta &= \frac{a_{ik}}{\sqrt{a_{ik}^{\ 2} + a_{kk}^{\ 2}}} \\ \cos \ \theta &= \frac{a_{kk}}{\sqrt{a_{ik}^{\ 2} + a_{kk}^{\ 2}}} \end{aligned} \tag{32}$$



A fast algorithm for evaluating the above equations is given as follows.
Fig. 14. Flow chart for Givens Rotation algorithm

7.4 The singular value decomposition

The SVD is a method of extracting a diagonal matrix from a real matrix A which satisfies the following relationship $A = UDV^T$ where U and V are unitary matrices. SVD of any matrix is computed by decomposing the given matrices to 2x2 smaller dimension matrices and taking its inverse which is described in (Press 1992). This implementation considers a two step two sided unitary transformation in which each step is a diagonalization step (Hemkumar 1991) on in which each step is a diagonalization.

7.5 Systolic array based singular value decomposition for inverse computation:

The basis of the systolic array is the processing element PE. The PE is a simple computational devise capable of performing basic multiply and accumulate operation. At the beginning of each clock cycle,the PE reads in the values Aij and Ck performs the necessary arithmetic computation using the internally stored value. By this above method only one PE is busy at a time and only one row of matrix can be considered at a time. It is possible to evaluate all elements of the matrix using a vector C concurrently with processors being busy all the time. All that is necessary to place n rows of processing elements beneath the first row. Then the computation of the second inner product involving the second row of A follows directly behind the computation of the first element of the product, similarly with third row etc. After m clock periods the mth row begins to accumulate and after m cycles the results will become stable and can be stored in the output register array. The basic advantage of using a systolic array in the matrix computation is that these computational blocks are regular. The PE's only talk to nearest neighbors. The above points make the silicon VLSI layout of this computational structure relatively simple. Only one cell need be designed the entire array is formed by repeating this design many times which is a simple process in VLSI design. The interconnections between processors are simple because they talk only to nearest neighbors.The idea behind this systolic array is to produce a massively parallel computational architecture which is capable of executing the QR decomposition in O(m*n) time units. As we see later conventional implementations require $O(3n^2(m-n)$ units

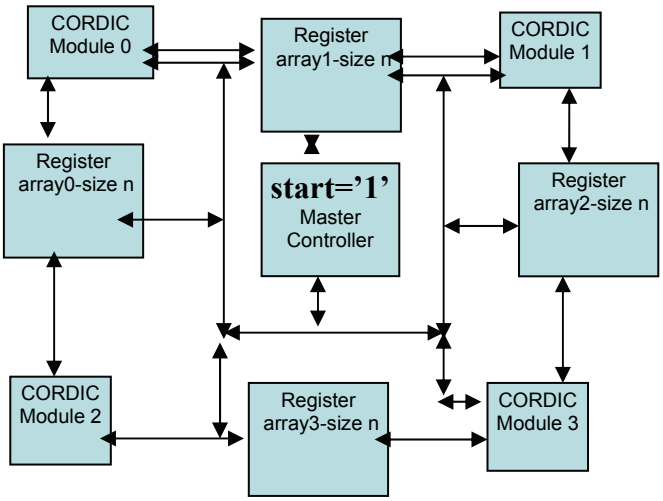


Fig. 15. SYSTOLIC CORDIC ARCHITECTURE FOR SVD

to compute the QR decomposition using the Givens procedure. Thus systolic architecture can be much faster. Further this procedure avoids the inefficiency of having to calculate the entire solution over again from the start for each iteration. Thus this systolic array gives us a new form of adaptive iterative structure, which can track changes in the LS solution as the environment changes or evolves. We now consider the systolic structure to eliminate the element a.

Two sided Jacobi algorithm is used for the computation of eigenvalue / singular value decomposition of a general matrix (Figure 15) A. As the serial jacobi algorithm is slow, a parallel systolic architecture is used with dynamic parallel ordering.This approach reduces the number of outer parallel iterations steps that two sided jacobi algorithm by 30-40 percent and hence It speeds up the inverse computation step. Such pre-conditioning should concentrate the Frobenius norm of A to diagonal as much as possible. For a diagonal matrix only one outer parallel iteration step would be required for the whole SVD computation and hence the concentration of Frobenius norm towards the diagonal might decrease the number of outer parallel iterations substantially.

7.5 Levinison-Durbin toeplitz solver

If the input matrix is symmetric as well as toeplitz system then by applying Levinison and Durbin's Algorithm one can solve the equations. Levinsons algorithm is an iterative algorithm and can be used to get only partial result for a toeplitz matrix and if the input matrix is also symmetric then one can make use of the Durbin's recursion to carry through the partial result to the final solution. Levinison-Durbin algorithm represents recursive system which solves toeplitz matrices and can be considered to model Auto Regressive models progressively and based on some criteria like AIC model order selection criteria one can choose the optimal model using LD algorithm. Normal matrix inverse computation requires $O(n^3)$ computations which can be done through Gaussian Elimination. If the matrix has got special structure as that of Toeplitz the inverse of the system can be found by this faster algorithm which models the linear set of equations to that of an auto regressive filter. Set of equations that describe Levinison- Durbin algorithm is given below. At the rth step of the LD Algorithm solves the rth truncated problem of Auto regressive filter design.

$$\begin{pmatrix} r(0) & \cdots & r(M) \\ \vdots & \ddots & \vdots \\ r(-M) & \cdots & r(0) \end{pmatrix} \begin{pmatrix} a_{M,0} \\ \vdots \\ \vdots \\ a_{M,M} \end{pmatrix} = \begin{pmatrix} \rho_M \\ 0 \\ \vdots \\ \vdots \end{pmatrix} \tag{33}$$

7.6 Fast matrix transposition unit

Large data matrices can be process efficiently can be loaded into memory using parallel architecture and for a n by n array stored in memory, it is convenient to transfer one column at a time ans transposing a matrix means swapping elements of two different columns which is indeed difficult if o the transposition by column by column access method. Transposition can be done by assuming block matrices as shown below.

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \qquad A^T = \begin{pmatrix} A^T{}_{11} & A^T{}_{21} \\ A^T{}_{12} & A^T{}_{22} \end{pmatrix} \tag{34}$$

By transposing only Totally n^2 columns must be read which necessitates O (n^2) operations which can be reduces to O($n\log_2 n$) by interchanging A^T_{21}, A^T_{12}block transpose matrices. Recursively computing the four block metrics A^T_{11}, A^T_{12}, A^T_{21}, A^T_{22}logarithmic complexity is obtained.

8. Hardware architecture

The system is implemented in ALTERA'S EP2C20f484C7 FPGA with NIOS 2 processor. Matlab tool was used for designing system level specification and the RTL simulation is carried out in Modelsim. So as to reduce energy consumption over the recognition period, the operating frequency of the system is set to 12.5 MHz compared to the maximum operating frequency of 33 MHz, which results in low power consumption of 32mW which is 23% decrease in power compared to the previous case. For ASIC implementation, RTL compiler is used for synthesis and advanced Synthesis flows like Multi-Vt, DFT have been exercised in our design and the results are shown in Table 1.

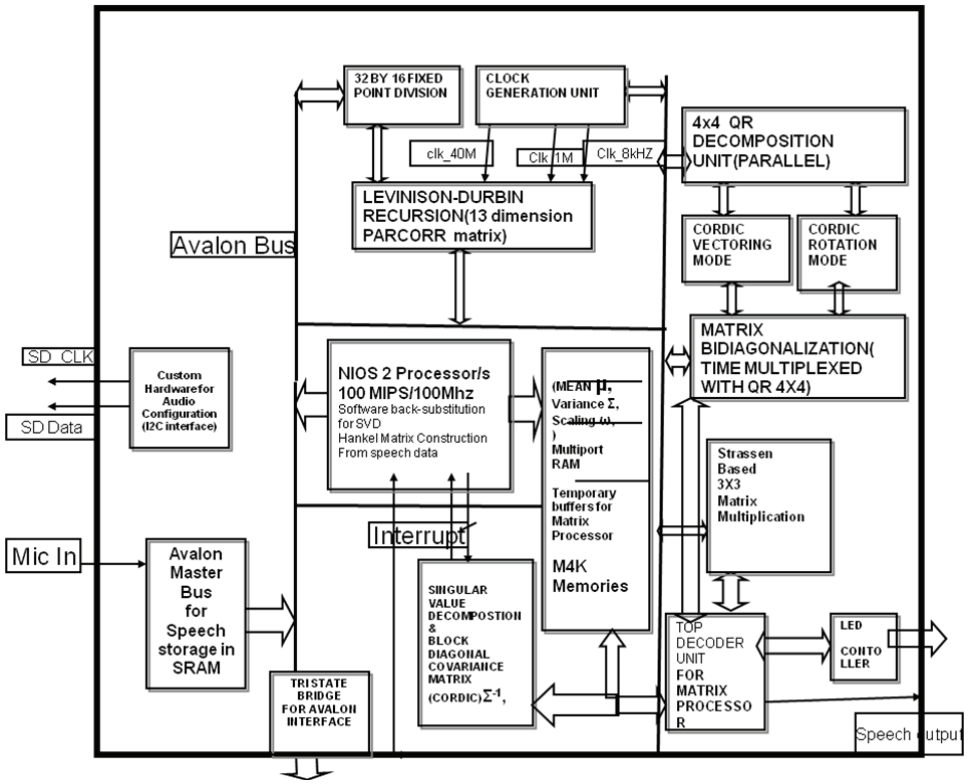


Fig. 16. Overall Hardware Architecture in Altera FPGA.

The hardware module is interfaced to software unit as a custom instruction in Altera FPGA and the latency of individual units are shown in Table2. AN4 database from Carnegie Mellon University was used to train the word models. Hardware emulation of the recognizer indicates a 5% improvement in speech recognition in noisy environment at the cost of 10%

increase in hardware complexity. Since the SVD block is used in time multiplexed fashion, 15% improvement in recognition accuracy can be obtained with the proposed unit. The following table depicts the low hardware requirement for the Filter bank based feature extraction unit which is implemented as a hardware-software combined block compared to a fully hardware LPC block and hence this implementation has very low power consumption. The following tables shows the ASIC implementation results for the matrix processor block and its power consumption values. This particular block can operate at a maximum frequency of 1Ghz. Complete power estimation for this chip was not carried out due to unavailability of the IO book power information and processor power dissipation information.

Module	Power(nW)	Area (micro m^2)	Gates	Frequency MHz
SVD	2431120	25320	3685	1000
QR Decomposition	2005727	27109	5398	1000
Levinison-Durbin	34276231	106519	8032	250
CORDIC Rotation	571500	5308	758	1000
CORDIC Vectoring	241919	4557	632	1000

Table 4. ASIC implementation results for matrix processor

9. Speech recognition results

No. of States		K=4	K=6	K=8	K=10	K=16	K=32
Memory Required for Feature Vectors – 13 Dimensions (in FPGA resources)	Covariance	910	910	910	910	910	910
	Mean	3900	5850	7800	9750	15600	31200
	Transition matri	1280	2880	5120	8000	20480	81920
	Initial probability	58	87	115	184	230	460
	Total Memory	6148	9727	13945	18844	37220	114490
Memory Required 26 Dimensions (in FPGA resources)	Covariance	1820	1820	1820	1820	1820	1820
	Mean	7800	11700	15600	19500	31200	62400
	Transition matrix	1280	2880	5120	8000	20480	81920
	Initial probability	58	87	115	184	230	460
	Total Memory	10958	16487	22655	29504	53730	145600
Memory Required 39 Dimensions (in FPGA resources)	Covariance	2730	2730	2730	2730	2730	2730
	Mean	11700	17550	23400	29250	46800	93600
	Transition matri	1280	2880	5120	8000	20480	81920
	Initial probability	58	87	115	184	230	460
	Total Memory	15768	23247	31365	40164	70240	178710

Table 5. Logic Utilization for realizing Memory in FPGA for various HMM states

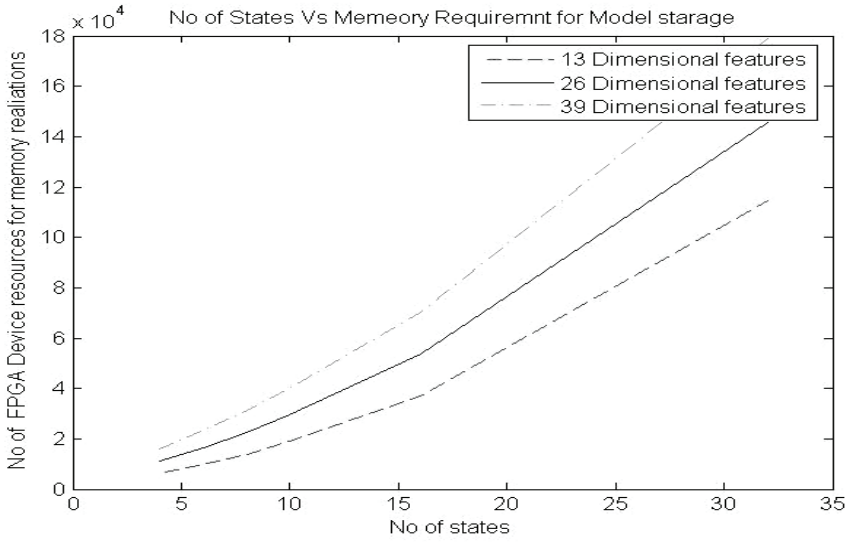


Fig. 16. Logic Utilization for realizing Memory in FPGA for various HMM states

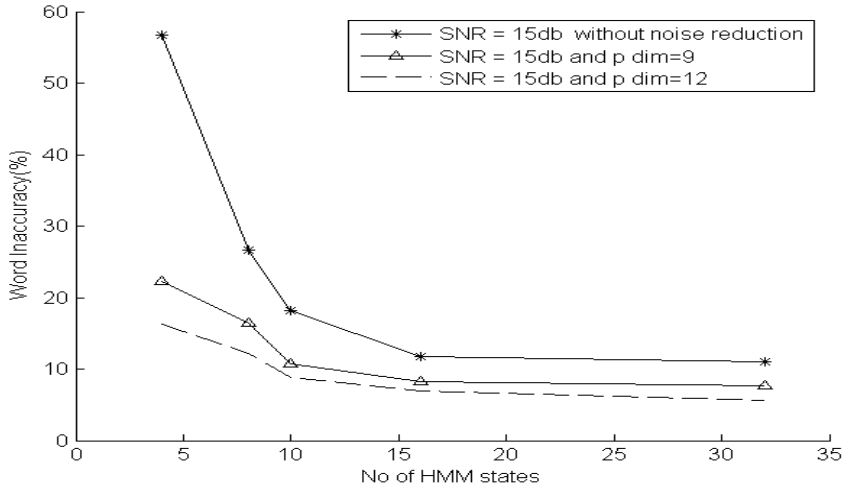


Fig. 17.Word Error rate for Enhanced speech for different SNR conditions.

AN4 database from Carnegie Mellon University has been used to train the word models. AN4 Database has got 130 different words spoken by people of different ages,dielect and gender. As the time complexity for the training of 130 words is extremely high, 10 digits from these 130 words have been used for training of the database and the system has been developed for 10 words. The AN4 database files are in raw PCM format, sampled at 16 kHz, in big endian byte order. Since the software IWR system is developed for 8 KHz sampling rate, A tool called Audacity is used to convert the 16Khz raw data file to 8Khz Wav file which was later fed to Matlab to extract the feature vectors using the feature extraction algorithm. The recognition performance of the proposed IWR system for different Number of states is plotted in Figure 16 & Figure 17.

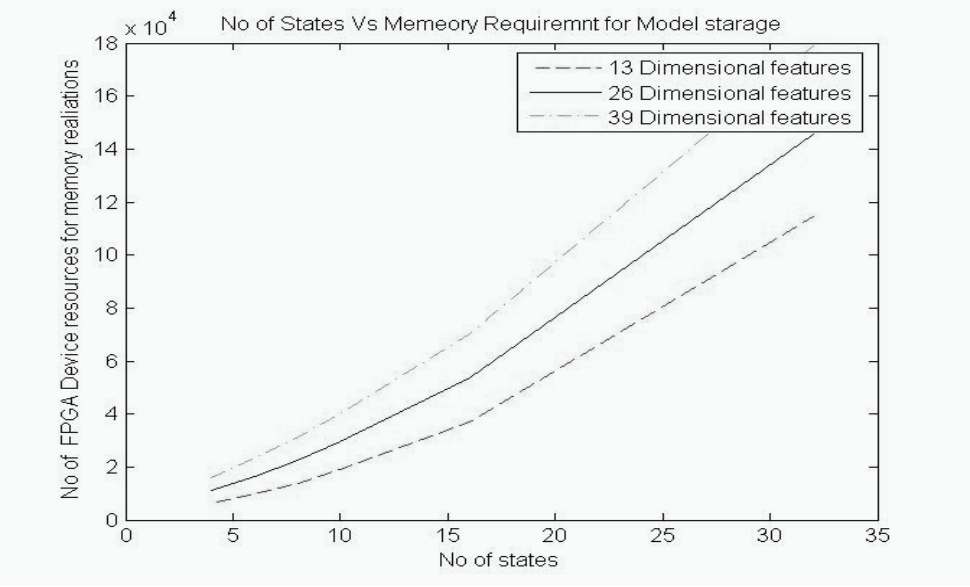


Fig. 18. Memory requiremnt for HMM models with different states and feature vectors

10. Conclusion

This 32 state Continuous Hidden Markov Model based speech recognition hardware provides 82.7% recognition accuracy in noisy speech environments at 15db for a 10 word sample space collected from words uttered by people of different age, gender and dialect as the data are processed in identical fashion. Hardware emulation of the recognizer indicates a 5% improvement in speech recognition in noisy environment at the cost of 10% increase in hardware complexity. Since the SVD block is used in time multiplexed fashion, 15% improvement in the overall recognition accuracy has been obtained.

ALU OPTION	Hardware Details	Cycles per Instruction	Result Latency Cycles	Parallel implementation of the 13 features with 4 states
LE Based Multiplier	32 x 4-bit multiplier	11	2	76
Embedded Multiplier on Cyclone	ALU includes 32 x 16-bit multiplier	5	2	24
Hardware Divide	ALU includes multicycle divide	4-66	2	0
Custom Avalon Speech Controller	Custom Instruction1	2	2	1
Custom Avalon SVD	Custom Instruction2	16	2	1

Table 6. Hardware Execution cycles of the Custom Instructions in NIOS Processor

11. References

H.Abut, H.L. Hansen, K.Takeda, "DSP for In Vehicle and Mobile Systems", Springer Publishers, 2005.

J. Pihl, T. Svendsen, and M. H. Johnsen, "A VLSI implementation of pdf computations in HMM based speech recognition," in Proc. IEEE TENCON'96, 1996, pp. 241–246.

K. Hermus, P.Wambacq, and H.V. Hamme, "A Review of Signal Subspace Speech Enhancement and Its Application to Noise Robust Speech Recognition", EURASIP Journal on Advances in Signal Processing Volume 2007, Article ID 45821.

L.R.Rabiner & B.H. Juang," Fundamentals Of Speech Recognition." Prentice -Hall, AT&T, U.S.A, 1993.

Simon Haykin, "Adaptive Filter Theory", Third Edition , Prentice Hall Information and System series,2002.

N. D. Hemkumar, "A Systolic VLSI Architecture for Complex SVD", Postgraduate Thesis, Rice University, Houston, Texas, May 1991.

N.Karthikeyan, S.Arun, K.Murugarraj, M.John, "An application specific matrix processor for signal subspace based speech enhancement in noise robust speech recognition applications", pages 766-769,7th Internation Conference on ASIC(ASICON2007), Guilin, China,2007.

N.Karthikeyan, S.Arun, K.Murugaraj, M John, "Hardware and Software acceleration of Front End Processing Unit in Scalable Noise robust Word HMM based speech recognition systems", 4th International Conference on SOC (ISOCC-2007), Seoul, Korea, (Poster).

S. Nedevschi, R.K. Patra, E.A.Brewer,"Hardware Speech Recognition for User Interfaces in Low Cost, Low Power Devices", *DAC 2005,* June 13.17, 2005, Anaheim, California, USA.

S.Yoshizawa, N. Wada, N. Hayasaka, IEEE, and Yoshikazu Miyanaga, "Scalable Architecture for Word HMM-Based Speech Recognition and VLSI Implementation in Complete System", *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS – I: REGULAR PAPERS*, VOL. 53, NO. 1, JANUARY 2006

Saeed V. Vaseghi,"Advanced Digital Processing and Noise Reduction", John Wiley and Sons Publishers, Second Edition, 2000.

U. Mayer Baese, "Digital Signal Processing with Field Programmable Gate Arrays", Springer Publishers, 2005.

William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery
"Numerical Recipes in C", CAMBRIDGE UNIVERSITY PRESS, 1992.

"Avalon Streaming Interface Specification", Version 1.0, Altera Corporation, November 2006.

29

Voice Activated Appliances for Severely Disabled Persons

Soo-young Suk and Hiroaki Kojima
Advanced Industrial Science and Technology
Japan

1. Introduction

People with severe speech and motor impairment due to cerebral palsy are great difficult to move independently and also cannot control home electric devices. Computer has much to offer people with disability, but the standard human-machine interface (e.g. keyboard and mouse) is inaccessible to this population. In this chapter, we describe a speech recognition interface for the control of powered wheelchair and home automation systems via severely disabled person's voices. In particular, we consider that our system can be operated by inarticulate speech produced by persons with severe cerebral palsy or quadriplegia in real-environment.

The aim of our research is divided two targets. One is easy to control of various home appliances by voice, and the other is to enable severely disabled person's movement independently using voice activated powered wheelchair. At first, Home automation system product for intelligent home is increasingly getting very common by the help of intelligent home technologies that increased easy, safety and comfort. Moreover, home automation is an absolute benefit and can improve the quality of life for the user. Home automation houses have been developed to apply new technologies in real environment, such as Welfare Techno Houses (Tamura et al., 2007), Intelligent Sweet Home (Park et al., 2007), Smart House (West et al., 2005). Interfaces based on gestures or voices have been widely used for home automation. However, gesture recognition based on vision technologies depends critically on the external illumination conditions. And gesture recognition is difficult or impossible for people suffering from severe motor impairments, such as paraplegia and tremors. Recently, a voice-activated system using commercial voice-recognition hardware in a low-noise environment has been developed for disabled persons capable of clear speech (Ding & Cooper, 2005).

The next, powered wheelchairs provide unique mobility for the disabled and elderly with motor impairments. Sometimes, the joystick is a useless manipulation tool because the severely disabled cannot operate it smoothly. Using natural voice commands, like ``move forward'' or ``move left'' relieves the user from precise motion control of the wheelchair. Voice activated powered wheelchair is required safety manipulation with high speech recognition accuracy because the accident can occur by a misrecognition. Although current speech recognition technology has reported high performance, it is not sufficient for safe voice-controlled powered wheelchair movement by inarticulate speech affected by severe

cerebral palsy or quadriplegia, for instance. To cope with the pronunciation variation of inarticulate speech, we adopted a lexicon building approach based on Hidden Markov Model and data mining (Sadohara at al., 2005), in addition to acoustic-modeling-based speaker adaptation (Suk at al., 2005). We also developed noise-canceling methods, which reduce mechanical noise and environmental sounds for practical use on the street (Sasou at al., 2004). However, though our voice command system has improved recognition performance by various methods, the system requires a guarantee of safety for wheelchair users in two additional conditions.

- To move only in response to the disabled person's own voice.
- To reject non-voice command input.

The first problem is to prevent operation of the wheelchair by unauthorized persons near the disabled user. A speaker verification method can be applied to solve this problem, but it is difficult to verify when using short word commands. Therefore, we are now developing a speaker position detection system using a microphone array (Jonson at al., 1993; Sasou & Kojima, 2006). The second problem is that a lot of other noise is input when the voice command system is being used. Also, a voice-activated control system must therefore reject noise and non-voice commands such as coughing and breathing, and spark-like mechanical noise in the preprocessing stage. A general rejection method has achieved a confidence measure using a likelihood ratio in a post-processing step. However, this confidence measure is hard to use as a non-command rejection method because of the inaccuracy of likelihood when speech recognition deals with unclear voice and non-voice sounds. Thus, a non-voice rejection algorithm that classifies Voice/Non-Voice (V/NV) in a Voice Activity Detection (VAD) step is useful for realizing a highly reliable voice-activated powered wheelchair system.

The chapter first presents the F_0 estimator and the non-voice rejection algorithm. Next, the inarticulate speech recognition is described in Section 3. In Section 4, we present a developed voice activated control system. And we evaluate the performance of our system in Section 5. Lastly, we offer our conclusions in Section 6.

2. Non-voice rejection using V/NV classification

The general VAD uses short time energy and/or ZCR for start and end point detection in a real-time voice command system with low complexity. However, VAD has a problem because various sounds are determined as voice sounds. Previous V/NV classification algorithms have generally adopted statistical analyses of F_0, the Zero-Crossing Rate (ZCR), and the energy of short-time segments. A method for voicing decision within a pitch-detection algorithm is presented in (Rouat at al., 1997). A combination of these methods, a cepstrum-based F_0 extractor, has been proposed (Ahmadi & Andreas, 1999). An auditory-based method for voicing decision within a pitch-tracking algorithm appears in (Mousset at al., 1996). For the purpose of non-voice rejection, we propose a V/NV classification using a reliable F_0 estimator.

2.1 YIN: fundamental frequency estimator

V/NV classification using F_0 information has been strongly tied to the problem of a pitch detection algorithm (PDA). A PDA can be formulated as an average magnitude difference function, average squared difference function, or similar autocorrelation methods in the time domain. In addition, cepstrum analysis is possible in the frequency domain by applying

the harmonic product spectrum algorithm. Among these F_0 extraction methods, we use the well known auto-correlation method based on YIN that has a number of modifications to reduce estimation errors (de Cheveigné, 2002). This method has the merit of not requiring fine tuning and uses fewer parameters. The name YIN (from "Yin" and "Yang" of oriental philosophy) alludes to the interplay between autocorrelation and the cancellation that it involves. The autocorrelation function of a discrete signal x_t may be defined as

$$r_t(\tau) = \sum_{j=t+1}^{t+W} x_j x_{j+\tau} \tag{1}$$

where $r_t(\tau)$ is the autocorrelation function of lag τ at time index t, and W is the integration window size. YIN achieves a difference function instead of an autocorrelation function that is influenced in bias value.

$$d_t(\tau) = \sum_{j=t-\tau-W/2}^{t=\tau/2+W/2} (x_j - x_{j+\tau})^2 \tag{2}$$

Here, $d_t(\tau)$ is the difference function to search for the values of τ for which the function is zero. The window size shrinks with increasing values of τ, resulting in the envelope of the function decreasing as a function of lag as illustrated in Fig. 1(a). The difference function must choose a minimum dip that is not zero-lag. However, setting the search range is difficult because of imperfect periodicity. To solve this problem, the YIN method replaces the difference function with the cumulative mean normalized difference function of Eq. (3). This function is illustrated in Fig. 1(b).

$$d'_t(\tau) = \begin{cases} 1, & if\ \tau = 0, \\ d_t(\tau) \Big/ \left[(1/\tau) \sum_{j=1}^{\tau} d_t(j) \right] & otherwise \end{cases} \tag{3}$$

The cumulative mean normalized difference function not only reduces "too high" errors, but also eliminates the limit of the frequency search range, and no longer needs to avoid the zero-lag dip. One of the higher-order dips appears often in F_0 extraction, even when using

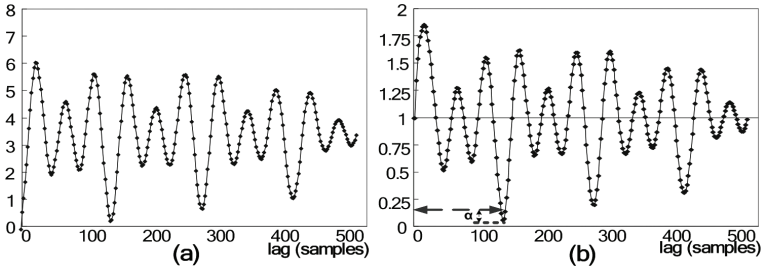


Fig. 1. (a) Example of difference function (b) Cumulative mean normalized difference function at same waveform

the modified function in Eq. (3). This error is called the sub-harmonic or octave error. To reduce the sub-harmonic error, the YIN method finds the smallest value of τ that gives a minimum of $d_t'(\tau)$ deeper than the threshold. Here, the threshold is decided by the value that adds a minimum of $d_t'(\tau)$ to the absolute threshold α in Fig. 1 (b). Absolute threshold is possible because of the achieved normalized processing in the previous step. In the final step, F_0 is extracted through the parabolic interpolation and best local estimation process.

2.2 V/NV classification

The general V/NV classification algorithm participates in the processing of each short-time speech segment. However, classification of a whole input segment is more important in reliable speech recognition in which non-voice rejection is possible. For this classification, the proposed algorithm decides V/NV from the ratio of the reliable F_0 contour over the whole input interval.

The function value $d_t'(\tau)$ defined by Eq. (3) is compared with the confidence threshold to decide the reliability of each F_0 frame. Here, the confidence threshold is selected such that the value is 0.05 to 0.2. Figures 2 and 3 depict examples of reliable F_0 contour extraction. A reliable F_0 contour using the cumulative mean normalized difference function is illustrated in Fig. 2 (b). When the confidence threshold of YIN-based F_0 is 0.1, only high reliability areas are selected, as illustrated in Fig. 2 (c).

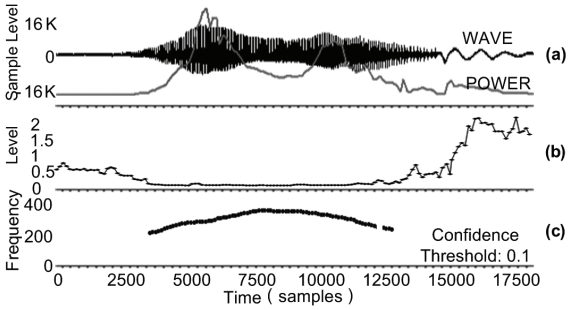


Fig. 2. (a) Example of a voice waveform (b) Cumulative mean normalized difference function calculated from the waveform in (a) (c) Reliable F0 contour in which the confidence threshold is applied

The conventional VAD method using energy and/or ZCR is detected noise as well as voice in Fig. 3 (a). However, you can see that reliable F_0 appears on only three frames because of the applied confidence threshold 0.1 in Fig. 3 (c). Furthermore, we can prove the performance by the examining that detected frequency is the inner voice frequency area. For V/NV classification from the extracted F_0 contour, we then compute the ratio of frames with the reliable F_0 as follows.

$$d = \frac{1}{M}\sum_{i=1}^{M} P_{th}(i) \tag{4}$$

$$P_{th}(i) = \begin{cases} 1 & if \;\; F_{\min} \le F_{oth} \le F_{\max} \\ 0 & otherwise \end{cases} \tag{5}$$

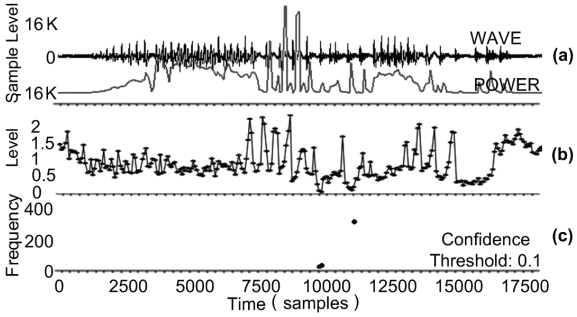


Fig. 3. (a) Example of a noise waveform (b) Cumulative mean normalized difference function calculated from the waveform in (a) (c) Reliable F0 contour where the confidence threshold is applied

Here, M indicates the total number of input frames, and $F_{\min}$=60Hz and $F_{\max}$=800Hz are experimentally chosen for a disabled person's voice. Finally, an input segment is classified as voice if d exceeds the V/NV threshold value. The cepstrum-based algorithm can also be used as the confidence threshold for extraction of the F_0 contour as indicated in Fig. 4. However, the F_0 extraction performance of a cepstrum-based algorithm is inferior to YIN, and it is difficult to determine a suitable threshold in various environments.

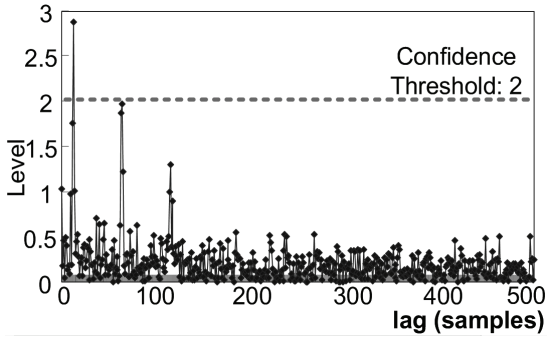


Fig. 4. Example of cepstrum signal to the applied confidence threshold

3. Inarticulate speech recognition

The severely disabled person has the problem of pronouncing even simple voice commands. Specially, our system is required high performance of speech recognition when controlling a powered wheelchair, because recognition performance is related to the disabled person's safety. Using the speech characteristics of each severely disabled person, the voice command input system needs to select words in which an utterance can be spoken easily and distinctly. Therefore, our system selected five of 12 word candidates using a test. However, as the disabled person has difficulty in speaking the command clearly, the utterance of "hi da ri" may be spoken as "hi hi hi da ri", "i a ri" etc., in spite of the selected voice command. For this reason, a single-template dictionary is unable to recognize unusual speech.

A multi-template dictionary was generated through analysis of the speech patterns of the disabled to solve this problem. To generate a multi-template dictionary, the initial dictionary

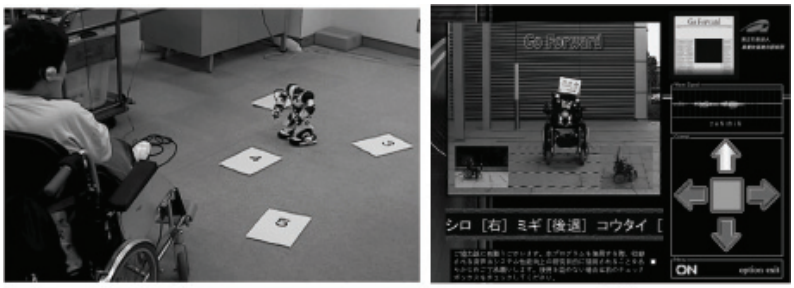
is prepared from the result of a phoneme-recognition experiment using a pre-recorded voice. The final dictionary is then generated by deleting unwanted candidates through a repetitive word-recognition experiment. The generated multi-template dictionary was comprised of the 27 templates of the "hidari" utterance, 15 templates for "migi," 13 templates for "mae," 4 templates for "koutai," and 5 templates for "ah."

Action	Command	Dictionary
Move Left	hidari hidari hidari	h i d a: r i: d a r i q h i h i i d a: r i ,…other 24
Move Right	migi	m i g i i m i z i p ,… other 13
Move Forward	mae	m a: a e m a e p i ,… other 11
Move Backward	koutai	k o u t a i ,… other 3
Stop	ah	a: ,… other 4

Table 1. Implemented set and multiple dictionaries for inarticulate speech recognition

Since speech recognition systems are known to demonstrate different results from reading speech and spontaneous speech, it is important for evaluation and modeling of voice command system. To obtain a sample of spontaneous speech affected by disability, which contains specific personal variations, we developed a voice operated toy robot and a graphical simulation demo system that uses the same recognition task such as in powered wheelchair operation.

For analysis of the input devices, each input device achieved speech detection through each recognition engine at the same time. Currently, we have collected more than 3000 unclear samples of speech affected by disability, using four types of microphones: headset(Audio-technica: AT810X), Bone conduction(Sony: ECM-TL1), Pin(PAVEC: MC-105), Bluetooth (Sonorix: OBH-0100). Transmission capacity of bluetooth microphone is limited by an 8 KHz sampling rate. Table 2 lists the recorded speech data used for the experimental evaluation. For the headset type, 579 inputs are collected. There are also 426 voice commands, 65 various noises, 76 other utterances that are not commands, and utterances of 12 another people. Therefore, V/NV classification is needed to satisfy voice activated powered wheelchair control requirements while maintaining high speech recognition accuracy.



(a) (b)

Fig. 5. Speech recording environment (a) Voice operated toy robot system (b) Graphical simulation demo system

	Voice command	Noise	Other speech	Other people
Headset	426	65	76	12
Bone conduction	405	339	88	286
Pin	399	21	90	361
Bluetooth	337	22	64	62
Total	1567	447	318	721

Table 2. Analysis of the number of recorded data

4. Voice activated control system

The voice activated home appliances control system diagram is shown in Figure 6. In the diagram, speech input device is can be use not only headset but also mobile telephone via Skype Voice of IP (VOIP) module. At First, the microphone captures the speech signal. The incoming audio stream is then segmented for recognition using the VAD module. The stream is transmitted from speech interface to the recognition engine where the recognition procedure is carried out. Our recognition engine employed a Julian decoder (Lee at al., 2001) with mel frequency cepstral coefficient and adapted a speaker-dependent acoustic model. Finally the recognition result is executed when results are satisfied with voice using V/NV classification module.

By using the result of speech recognition for infrared remote control, system can be remote control powered wheelchair and home appliances including TV, radio, VCD/DVD player, lights, and fan. System is also offers hand-free management of telephone calls and direct calling to family through one voice command. Software is designed and developed under the visual studio platform and using visual C++ programming, which can be installed and run in Ultra Mobile PC (UMPC) under operating system of embedded windows.

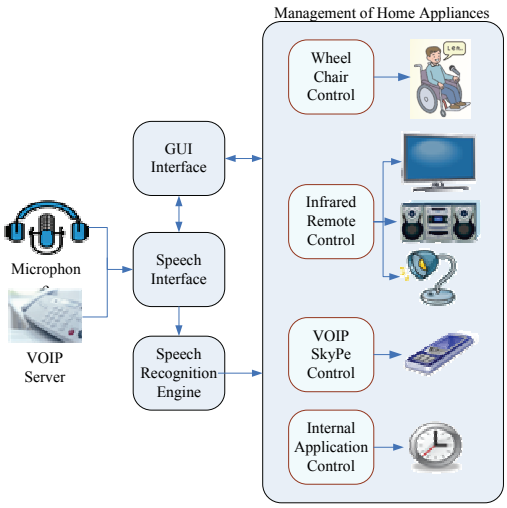
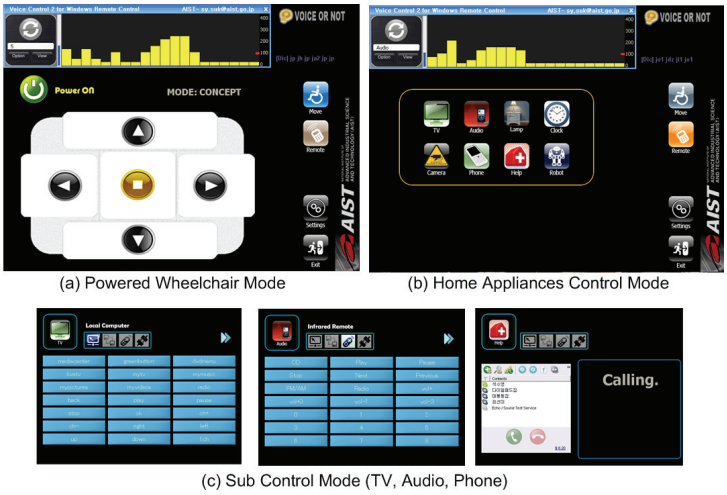


Fig. 6. Voice activated home appliance control system design

A voice controlled Graphic User Interface (GUI) is carefully designed for disabled person in Fig. 7. Click of the icon on the user's screen or voice command directly correspond to environmental commands (switching on the lamps, starting the Radio, calling the facilitator)

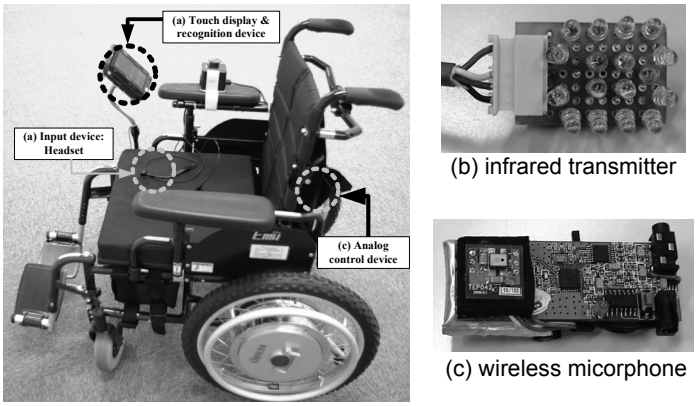


(a) Powered Wheelchair Mode (b) Home Appliances Control Mode

(c) Sub Control Mode (TV, Audio, Phone)

Fig. 7. Example of system GUI design (a) Powered wheel chair mode (b) Home appliances control mode (c) Appliance control sub mode

The developed system consists of a headset, a Pentium M 1.2GHz UMPC, infrared transmitter for long distance transmission and a wheelchair controller, as depicted in Fig. 8. Also, wireless microphone or mobile phone can be used instead of wire headsets for user convenience.



(a) Prototype powered wheelchair

(b) infrared transmitter

(c) wireless micorphone

Fig. 8. (a) Developed prototype wheelchair (b) Infrared transmitter (c) Wireless microphone

The voice commands to control wheelchair direction are not easy when use the minimum number of commands. So, our system use the state transition diagram for more free movement of voice activated powered wheelchair, as shown in Fig. 9.

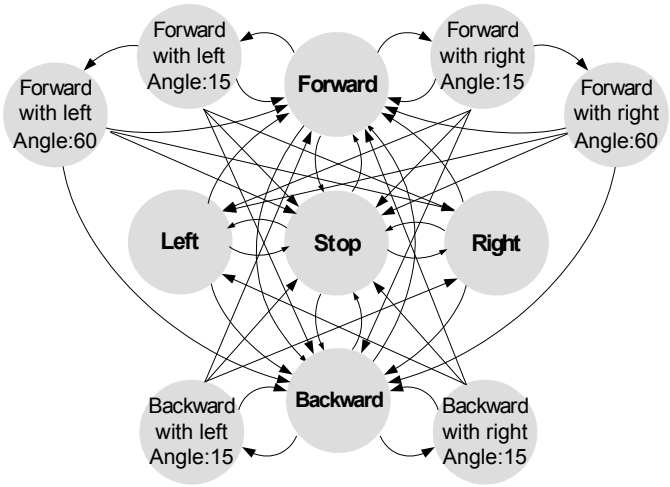


Fig. 9. State transition diagram by voice command

5. Experiment results

To evaluate the performance of our proposed method, we conducted V/NV classification experiments using the 1567 voice commands and 447 noises and employing YIN-based and cepstrum-based algorithms. The sampling frequency was 16kHz, the window size was 25ms, and the frame shift was 8ms.

Figure 10 depicts the V/NV classification performance and plots the recall-precision curves according to an individual confidence threshold. The results indicate that the YIN-based algorithm is superior to the cepstrum-based algorithm. When the confidence threshold of YIN is 0.08, the V/NV classification provides the best results with 0.97 and 0.99 rates for recall and precision. In other words, when the lowest threshold was selected for voice detection at a precision rate of 1, the miss-error rate of noise was only 4.9%.

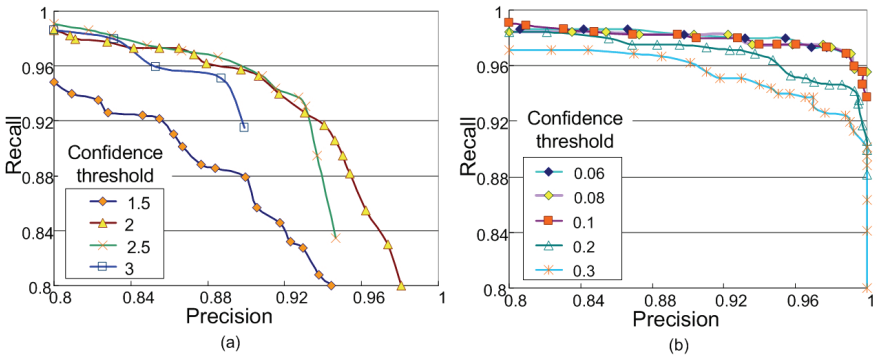


Fig. 10. Recall-precision curve of noise classification a) cepstrum b) YIN

Table 3 lists the best confidence threshold of each microphone with the best recall precision. When YIN uses the F_0 extraction method, the confidence threshold is stable at about 0.08.

Although the cepstrum algorithm can use the F_0 extraction method, it is difficult to decide on a suitable confidence threshold in each microphone environment.

	Headset	Bone conduction	Pin	Bluetooth
Cepstrum	3	2.5	1.5	2
YIN	0.05~0.1	0.06~0.08	0.07~0.1	0.08~0.1

Table 3. Confidence threshold analysis of four types of microphones with the best recall precision

A recognition experiment was performed in order to confirm the validity of the multi-candidate recognition dictionary and the adapted acoustic model for the basic performance of a voice-activated system. The recognition experiment used 2211 data elements recorded at an athletic meeting and outdoors with a noise background to evaluate the effectiveness of the proposed multi-template dictionary and adapted acoustic model. Ninety-six utterances were used for adaptation of the acoustic model, and the remaining 1334 utterances were used for evaluation.

Mix.	Baseline	Multi-Tem.	Adapt.	Multi-Tem. & Adapt
1	61.4	94.2	97.8	98.4
2	78.4	95.5	98.8	99.1
4	77.9	94.6	98.7	99.2
8	80.4	94.3	98.8	99.3
16	78.6	93.8	98.6	99.5
32	75.1	91.2	98.4	99.4

Table 4. Speech recognition accuracy with 2000-state HMnet model

The speaker-independent, 2000-state 16-mixture HMnet model was evaluated as the baseline. An average recognition rate of 78.6 was achieved, although there were five words in the dictionary because it did not consider the speech characteristics and variations of disabled persons. The average recognition rate was improved to 93.8% by applying the multi-template dictionary. The acoustic model that performed MAP adaptation achieved an average recognition rate of 98.6%. The average recognition rate was improved to 99.5% by applying the multi-template dictionary with the adapted acoustic model.

6. Conclusion

This chapter presented home appliances control system for independent life of the severely disabled person. In particular, the developed system can be operated by inarticulate speech and a non-voice rejection method for reliable VAD in a real environment with extraneous sounds such as coughing and breathing. The method classifies V/NV from the ratio of reliable F_0 contour over the whole input interval. We adopted the F_0 extraction method where YIN has the best performance among conventional methods. Our experiment results indicate that the false alarm rate is 4.9% with no miss-errors in which voice is determined to

be non-voice. And the average recognition rate was improved to 99.5% by applying the multi-template dictionary with the adapted acoustic model. Therefore, the speaker dependent acoustic model, dictionary and non-voice rejection algorithm can be helpful for realizing a highly reliable wheelchair control system.

7. Acknowledgement

I would like to thank K. Sakaue for his invaluable comments, and A. Sasou and other Speech Processing Group members for their contribution to this work. I wold also like to thank M. Suwa, T. Inoue and other members of Research Institute, National Rehabilitation Center for Persons with Disabilities for their support for the experiments. This work was supported by KAKENHI (Grant-in-Aid for JSPS Fellows).

8. References

Ahmadi, S. & Andreas S. S. (1999). Cepstrum-based pitch detection using a new statistical V/UV classification algorithm. *IEEE Trans. Speech Audio Processing*, Vol. 7, No. 3, pp. 333-339.

de Cheveigné, A. & Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustic Society of the America*, Vol. 111, pp. 1917-1930.

Ding, D. & Cooper, R.A. (2005). Electric powered wheelchairs. *IEEE Trans. Control Syst. Mag.*, Vol. 25, pp. 22–34.

Jonson, D. H. & Dudgeon, D.E. (1993). Array signal processing. *Prentice Hall*, Englewood Cliffs, NJ.

Lee, A.; Kawahara, T. & Shikano, K. (2001). Julius—an open source realtime large vocabulary recognition engine. *Proceeding of European Conference Speech Communication Technology*, pp. 1691–1694.

Mousset E., Ainsworth, W. A. & Fonollosa, J. A. R. (1996). A comparison of several recent methods of fundamental frequency and voicing decision estimation. *Proceeding of International Conference of Spoken Language Processing*, Vol. 2, pp. 1273–1276.

Park, K.; Bien, Z.; Lee. J.; Kim, B.; Lim, J.; Kim, J.; Lee, H.; Stefanov, D.H.; Kim, D.; Jung, J.; Do, J.; Seo, K.; Kim, C.; Song, W. & Lee, W. (2007). Robotic smart house to assist people with movement disabilities. *The Journal of Autonomous Robots*, Vol. 22, No. 2, pp. 183-198.

Rouat, J.; Liu, Y. C. & Morrisette, D. A. (1997). pitch determination and voiced/unvoiced decision algorithm for noisy speech. *The Journal of the Speech Communication*, Vol. 21.

Sadohara, K.; Lee, S.W. & Kojima, H. (2005). Topic Segmentation Using Kernel Principal Component Analysis for Sub-Phonetic Segments. *Technical Report of IEICE*, AI2004-77, pp. 37-41.

Sasou, A.; Asano, F.; Tanaka, K. & Nakamura, S. (2004). HMM-Based Feature Compensation Method: An Evaluation Using the AURORA2. *Proceeding International Conference Spoken Language Processing*, pp. 121-124.

Sasou, A. & Kojima, H. (2006). Multi-channel speech input system for a wheelchair. *Proceeding Mar Meeting of the Acoustical Society of Japan*, Vol 2006.

Suk, S.Y.; Lee, S.W; Kojima, H. & Makino, S. (2005). Multi-mixture based PDT-SSS Algorithm for Extension of HM-Net Structure. *Proceeding of September Meeting of the Acoustical Society of Japan*, Vol 2005, pp. 1-P-8.

Tamura, T.; Kawarada, A.; Nambu, M.; Tsukada, A.; Sasaki, K. & Yamakoshi, K. (2007). E-Healthcare at an Experimental Welfare Techno House in Japan. *The journal of Open Medical Informatics,*Vol. 1, No. 1, pp. 1-7.

West, G.; Newman, C. & Greenhill, S. (2005). *Using a camera to implement virtual sensors in a smart house.*, Smart Homes to Smart Care. IOS Press, pp. 83-90.

30

System Request Utterance Detection Based on Acoustic and Linguistic Features

T. Takiguchi, A. Sako, T. Yamagata and Y. Ariki
Kobe University
Japan

1. Introduction

Robots are now being designed to become a part of the lives of ordinary people in social and home environments, such as a service robot at the office, or a robot serving people at a party (H. G. Okuno, et al., 2002) (J. Miura, et al., 2003). One of the key issues for practical use is the development of technologies that allow for user-friendly interfaces. This is because many robots that will be designed to serve people in living rooms or party rooms will be operated by non-expert users, who might not even be capable of operating a computer keyboard. Much research has also been done on the issues of human-robot interaction. For example, in (S. Waldherr, et al., 2000), the gesture interface has been described for the control of a mobile robot, where a camera is used to track a person, and gestures involving arm motions are recognized and used in operating the mobile robot.

Speech recognition is one of our most effective communication tools when it comes to a hands-free (human-robot) interface. Most current speech recognition systems are capable of achieving good performance in clean acoustic environments. However, these systems require the user to turn the microphone on/off to capture voices only. Also, in hands-free environments, degradation in speech recognition performance increases significantly because the speech signal may be corrupted by a wide variety of sources, including background noise and reverberation. In order to achieve highly effective speech recognition, in (H. Asoh, et al., 1999), a spoken dialog interface of a mobile robot was introduced, where a microphone array system is used.

In actual noisy environments, a robust voice detection algorithm plays an especially important role in speech recognition, and so on because there is a wide variety of sound sources in our daily life, and because the mobile robot is requested to extract only the object signal from all kinds of sounds, including background noise. Most conventional systems use an energy- and zero-crossing-based voice detection system (R. Stiefelhagen, et al., 2004). However, the noise-power-based method causes degradation of the detection performance in actual noisy environments. In (T. Takiguchi, et al., 2007), a robust speech/non-speech detection algorithm using AdaBoost, which can achieve extremely high detection rates, has been described.

Also, for a hands-free speech interface, it is important to detect commands in spontaneous utterances. Most current speech recognition systems are not capable of discriminating system requests - utterances that users talk to a system - from human-human conversations.

Therefore, a speech interface today requires a physical button which on and off the microphone input. If there is no button for a speech interface, all conversations are recognized as commands for the system. The button spoils the merit of speech interfaces that users do not need to operate by the hand. Concerning this issue, there are researches on discriminating system requests from human-human conversation by acoustic features calculated from each utterance (S. Yamada, et al., 2005). And also, there are discrimination techniques using linguistic features. Keyword or key-phrase spotting based methods (T. Kawahara, et al., 1998) (P. Jeanrenaud, et al., 1994) have been proposed. However, using keyword spotting based method, it is difficult to distinguish system requests from explanations of system usage. It becomes a problem when both utterances contain a same "keywords." For example, the request speech is "come here" and the explanation speech is "if you say come here, the robot will come here." In addition, it costs to construct a network grammar to accept flexible expressions.

In this chapter, an advanced method of discrimination using acoustic features or linguistic features is described. The difference of system requests and spontaneous utterances usually appears on the head and the tail of the utterance (T. Yamagata, et al., 2007). By separating the utterance section and calculating acoustic features from each section, the accuracy of discrimination can be improved. The technique based on acoustic features is able to detect system requests reasonably because it will not be dependent on any task and it does not need to reconstruct the discriminator when the system requests are added or changed.

Also, consideration of the alternation of speakers is described in this chapter. Considering turn-taking before and after the utterance, the performance can be improved. Finally, we take linguistic features into account, where Boosting is employed as a discriminant method. Its output score is not a probability, though, so the Boosting output score is converted into pseudo-probability using a sigmoid function. Though the technique based on linguistic features is dependent on tasks and it will need to reconstruct the discriminator when the system requests are modified, the accuracy of discrimination using linguistic features is better than that of the technique based on acoustic features.

2. Utterance verification using acoustic features

We describe the system request detection based on acoustic features first, where SVM (Support Vector Machine) is used. The overview of the system is shown in Figure 1. The proposed method based on acoustic features is able to detect system requests reasonably, because it does not need to reconstruct the discriminator when the system requests are added or changed.

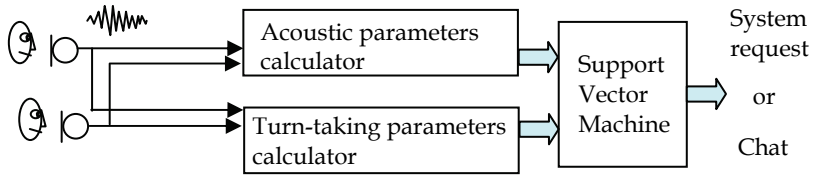


Fig. 1. System overview of utterance verification using acoustic features

2.1 Acoustic parameters

Even if we speak unconsciously, there are acoustic differences between utterances to equipments and those to humans under the condition the subject equipment is machinelike. In our work, we focus on the different characteristics of commands and human-human conversations which usually appear on the head and the tail of the utterance.

The start point and the end point of the utterance are indistinct in chatters while there are no sounds before and after the utterance in commands. There are mainly two reasons that make the start and the end point unclear. One reason is there are usually fillers and falters in chatters while there are short pauses on the head and the tail of utterances in commands. We usually put a short pause before a command to clarify and keep quiet until the system responds something. The other reason is the following person often begins to talk while the current person does not finish talking yet. In this Section, we deal with the former case. To put the former phenomenon to practical use, we calculate acoustic parameters not from the whole utterance section but from each three sections below.

To extract the head and the tail of the utterance, the power and zero-crossing are used in this work. Figure 2 is the wave form of a command utterance, and Figure 3 is that of a spontaneous utterance (chat). The head and tail of the utterance are indistinct in chatters while there are no sounds before and after the utterance in commands as described above. Therefore, as the head and tail of the utterance contain useful information written above, we do not join these margins to the detected utterance section, but calculate acoustic parameters (Table 1) also from each margin separately.

Calculated acoustic parameters are 8 dimensions shown in Table 1, but we calculate them from three sections described above. Thus, the acoustic features are 24 dimensions. The power is computed by Root Mean Square (RMS). The pitch is calculated by LPC residual correlation.

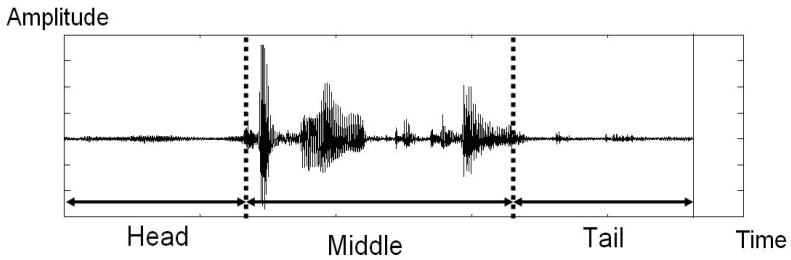


Fig. 2. A sample of system request

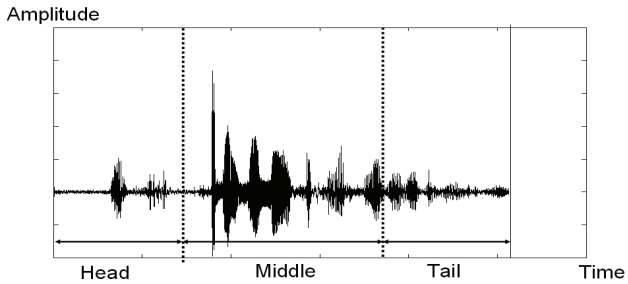


Fig. 3. A sample of spontaneous utterance (chat)

• Power	Average	Standard deviation	Max.	Max. – Min.
• Pitch	Average	Standard deviation	Max.	Max. – Min.

Table 1. Acoustic parameters (Power and Pitch are used.)

2.2 Turn-taking parameters

The sounds in the head and tail margins sometimes contain a speech of the next person, though it is not so loud. Therefore, we should separate voices of the next person from fillers and flatters. Considering which person speaks in each utterance section improves the accuracy of utterance verification. For example, the utterance seems to be a chat if speakers changes like $B \rightarrow A \rightarrow B$ in each section. In this work, we calculate these turn-taking parameters by crosspower-spectrum phase (CSP) (M. Omologo and P. Svaizer, 1996). Under the condition two microphones are set up for each person, we can tell the speaker from which microphone receives the utterance first. Considering the time lag CSP shows the maximum value, we can tell which microphone receives first. Moreover, CSP considers only the phase of the wave by normalizing the crosspower. This feature fits the condition that the distance of two microphones changes, where the power ratio of two microphones changes.

The crosspower-spectrum is computed through the short-term Fourier transform applied to windowed segments of the signal $x_i[t]$ received by the *i*-th microphone at time *t* :

$$CS(n;\omega) = X_i(n;\omega)X_j^*(n;\omega) \tag{1}$$

where * denotes the complex conjugate, n is the frame number, and ω is the spectral frequency. Then the normalized crosspower-spectrum is computed by

$$\phi(n;\omega) = \frac{X_i(n;\omega)X_j^*(n;\omega)}{|X_i(n;\omega)||X_j(n;\omega)|} \tag{2}$$

that preserves only information about phase differences between x_i and x_j. Finally, the inverse Fourier transform is computed to obtain the time lag (delay).

$$C(n;l) = F^{-1}\phi(n;\omega) \tag{3}$$

If the sound source does not move (this means it does not move in an utterance), $C(n;l)$ should consist of a dominant straight line at the theoretical delay. Therefore, a lag is given as follows:

$$\hat{l} = \arg\max_l \left\{ \sum_{n=1}^{N} C(n;l) \right\} \tag{4}$$

In the situation that the microphones are set up for each person, the reliability of the lag is the matters. Thus, we calculate *D* from each section and make them turn-taking parameters.

$$D = \begin{cases} C(\hat{l}) & (0 \le \hat{l} < (N-1)/2) \\ -C(\hat{l}) & (N-1)/2 \le \hat{l} < N-1 \end{cases} \tag{5}$$

3. Utterance verification using linguistic information

In this Section, we describe the proposed method that incorporates system request into a speech recognition system, where linguistic information in the system request task is used.

3.1 System request detection integrated with speech recognition

Speech recognition is formalized to find the most likely word sequence $\mathrm{W} = \{w_1,...,w_N\}$ as well as the system request intention $s = \{\mathrm{Request}, \mathrm{Chat}\}$. Given the sequence of observed feature vectors O, speech recognition is formalized as follows:

$$\begin{aligned}(\hat{s}, \hat{\mathrm{W}}) &= \arg\max_{s,\mathrm{W}} P(s, \mathrm{W} \mid \mathrm{O}) \\ &= \arg\max_{s,\mathrm{W}} \frac{P(s, \mathrm{W}, \mathrm{O})}{P(\mathrm{O})}\end{aligned} \tag{6}$$

The following Eq. (7) and (8) can be derived from the Bayesian theorem, where $P(\mathrm{O})$ is omitted due to independence from s and W.

$$P(s, \mathrm{W}, \mathrm{O}) = P(s)P(\mathrm{W} \mid s)P(\mathrm{O} \mid \mathrm{W}, s) \tag{7}$$

$$P(s, \mathrm{W}, \mathrm{O}) = P(\mathrm{W})P(\mathrm{O} \mid \mathrm{W})P(s \mid \mathrm{W}, \mathrm{O}) \tag{8}$$

Therefore, two scenarios (Eq. (7) and (8)) are considered in this work. First, Eq. (7) means that the acoustic model and the language model both depend on request intention s. In Eq. (7), we employ the request intention dependent language model and assume that the acoustic model is independent from request intention *s*. The N-gram which is dependent on the request intention is given by

$$P(\mathrm{W} \mid s) = \prod_i P(w_i \mid w_{i-1},...,w_{i-N+1}, s) \tag{9}$$

$P(\mathrm{W} \mid s = \mathrm{Request})$ and $P(\mathrm{W} \mid s = \mathrm{Chat})$ are learned from the system request corpus and conversation corpus, respectively. After the recognition process using two language models, we find the request intention label having the maximum likelihood.

Next, the formulation of Eq. (8) consists of normal acoustic and language models. These models are the same as speech recognition models without request intention. In addition, Eq. (8) includes the model $P(s \mid \mathrm{W}, \mathrm{O})$ that discriminates system requests based on word hypothesis **W** and observation **O** directly. $P(s \mid \mathrm{W}, \mathrm{O})$ is a discrimination model such as Boosting or Support Vector Machines (SVM). Here, we employ a Boosting model due to computational costs, flexibility of expression and ease of combining various features. However, Boosting is not a probabilistic model. It is necessary to convert Boosting output $f(\mathrm{W},\mathrm{O})$ into pseudo-probability so that it can be incorporated into the probability-based speech recognition system. Consequently, Boosting output is converted into pseudo-probability using sigmoid function as shown in Figure 4. Sigmoid function can model close to the discriminative boundary in detail, and the range of values is 0 to 1. The parameters, a and b, are weighting factors of the sigmoid function, and they are estimated by the gradient method. Converting Boosting output $f(\mathrm{W})$ into pseudo-probability leads to the following derived equations:

$$P(s = \text{Request} \mid \text{W}, \text{O}) \approx \text{sigmoid}(f(\text{W}))$$
$$P(s = \textit{Chat} \mid \text{W}, \text{O}) \approx 1 - \text{sigmoid}(f(\text{W})) \tag{10}$$

Here, language information only is used.
By integrating system request detection with speech recognition, system request detection can incorporate not only 1-best results but also hypotheses. In addition, it makes it possible to decide the hypothesis for request detection based on a probability framework. For example, there are two hypotheses, such as "Come here" and "You say come here." Here "Come here" is a system request and "You say come here" is a chat. In order to integrate these scores and speech recognition probabilities, these scores from AdaBoost are converted into pseudo-probabilities. After integration, the hypothesis with the best scores is selected as a result of system request detection. Even if the speech recognition probability, $P(\text{W})P(\text{O} \mid \text{W})$, of "You say come here." is larger than "Come here," when the boosting score of "Come here" is high enough, "Come here" will be selected as a final result.

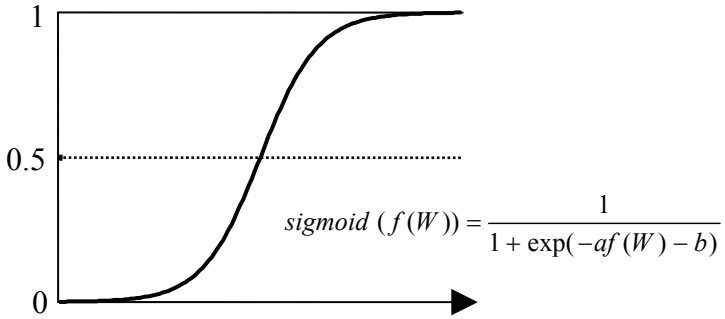


Fig. 4. Sigmoid function. Boosting output is converted into pseudo-probability using the sigmoid function.

3.2 Boosting

In this subsection, we describe a discrimination model based on Boosting in order to calculate $P(s \mid \text{W}, \text{O})$ in Eq. (10). AdaBoost is one of the ensemble learning methods that construct a strong classifier from weak classifiers (R. Schapire, et al., 1998). The AdaBoost algorithm uses a set of training data, $\{(W_1, Y_1), ..., (W_n, Y_n)\}$, where W_n is the n-th feature. In this work, the feature is a word (unigram) or a pair of words (N-gram). Y is a set of possible labels. For the system request detection, we consider just two possible labels, $Y = \{-1, 1\}$, where the label, 1, means "system request," and the label, -1, means "chat." For weak classifiers, single-level decision trees (also known as decision stumps) are used as the base classifiers (R. Schapire, et al., 2000). The weak learner generates a hypothesis $h_t : W \rightarrow \{-1, 1\}$ that has a small error. In the weak learner proposed by Schapire at el., the weak learners search all possible terms (unigram word or a pair of words) in training data and check for the presence or absence of a term in the given utterance. Once all terms have been searched, the weak hypothesis with the lowest score is selected and returned by the weak learner.
Next, AdaBoost sets a parameter α_t according to Eq. (13). Intuitively, α_t measures the importance that is assigned to h_t. Then the weight $z_{t+1}(i)$ is updated.

$$z_{t+1}(i) = \frac{z_t(i)\exp\{\alpha_t I(h_t(W_i) \neq Y_i)\}}{\sum_{j=1}^{n} z_t(j)\exp\{\alpha_t I(h_t(W_j) \neq Y_j)\}} \tag{11}$$

The Eq. (11) leads to the increase of the weight for the data misclassified by h_t. Therefore, the weight tends to concentrate on "hard" data. After T-th iteration, the final hypothesis, $f(W)$, combines the outputs of the T weak hypotheses using a weighted mojority vote. The following shows the overview of the Adaboost.

Input: n examples $\{(W_1, Y_1),...,(W_i, Y_i),...,(W_n, Y_n)\}$

Initialize: $z_1(i) = 1/n, \quad i = 1,...,n$

Do for $t = 1,...,T$

1. Train a weak learner with respect to the weight z_t and obtain hypothesis $h_t : W \to \{-1, 1\}$
2. Calculate the training error e_t of h_t.

$$e_t = \sum_{i=1}^{n} z_t(i) \frac{I(h_t(W_i) \neq Y_i) + 1}{2} \tag{12}$$

3. Set

$$\alpha_t = \log \frac{1 - e_t}{e_t} \tag{13}$$

4. Update the weight

Output: final hypothesis

$$f(W) = \frac{1}{|\alpha|} \sum_{t=1}^{T} \alpha_t h_t(W) \tag{14}$$

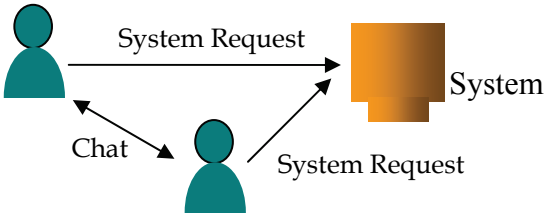


Fig. 5. Two person + one system dialog

4. Experiments

4.1 Recording conditions and details of corpus

The overview of the recording condition is shown in Figure 5. The task has the following features.

- Two people are in proximity to the system concurrently.
- People talk with each other freely and make requests to the system at will.
- The system has several kinds of functions (Table 2).
- Commands and utterances are recorded through microphones clipped to the chest of each speaker.

Functions	Sound source direction estimation based on CSP
	Move toward/away from sound source
	Obstacle avoidance
	Place a bottle using the gripper
	Take a face photo
Command examples	*Kotchi ni kite.* (Come here.)
	Shashin wo totte. (Take my photo.)
	Mukoh he itte. (Go to the other side.)
	Watashi ni tsuite kite. (Come with me.)
	Bottle wo oite. (Place the bottle.)

Table 2. Functions of the robot

It is ordinary for two or more people to be in close proximity to the system at the same time. For example, a driver uses a car navigation system while talking with passengers, or someone controls a robot in the presence of an audience. In our experiment, we used the robot for the system as shown in Figure 6. The typical usages are to call the robot by saying, "Come here" and to have the robot take a picture by speaking "Take my picture." The robot

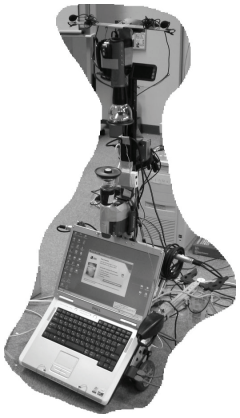


Fig. 6. Picture of mobile robot built in this work

can recognize the fixed commands shown in Table 1 at present. However, recorded speeches include many other command expressions such as "Come on," "Come rapidly," "Come, uhh ... here," etc. These utterances are spoken to control the robot. However the robot cannot recognize these at present. We labeled these utterances as a system request since one of our purposes is to accept flexible expressions (we collected these utterances on purpose).

Non-request utterances consist of ordinary conversation statements. These utterances are spoken in spontaneous speaking style, and so it is too difficult to recognize accurately. In

addition, explanation utterances of the robot usage were included. For example, "You say, 'Come here,' and the robot will come," "Come here, go away and so on," etc. Note that these utterances include the same phrases that are found in the system requests.
The length of the recording time is 30 minutes. We labeled those utterances manually. Table 3 shows the result of cutting out utterances from the recorded speech data.

Total utterance	System request	Total vocabulary size
330	49	700

Table 3. Total number of utterances and system requests

4.2 Evaluation of utterance verification using acoustic features

First, experiments were performed to test the utterance verification (system request detection) using the acoustic features. In this work, we used SVM with RBF (Gaussian) kernel. When more than two kinds of parameters are used at the same time, we combined parameters as follows:

$$U = [\alpha P_1 \quad \beta P_2] \tag{15}$$

Here U is combined vector and the original feature vectors are P_1, P_2, α and β were given experimentally.
Table 4 shows the results of utterance verification evaluated by leave-one-out cross-validation. In this experiment, we set 0.7 seconds for both margins before and after the clear utterance sections. The results are the cases F-measure became the maximum values. The F-measure became 0.86 where acoustic parameters (24 dim.) are calculated from proposed three utterance sections, while that was 0.66 where the feature values (8 dim.) are calculated from a whole utterance. Then, adding turn-taking features, it turned out to be 0.89.

	Precision	Recall	F-measure
Acoustic (8 dim.)	0.71	0.61	0.66
Acoustic (24 dim.)	0.80	0.92	0.86
Acoustic (24 dim.) + Turn-taking	0.87	0.92	0.89

Table 4. Result of utterance verificat

4.3 Evaluation of utterance verification using linguistic information

4.3.1 Conditions of speech recognition

In the acoustic model, the baseline training data consisted of about 200,000 Japanese sentences (200 hours) spoken by 200 males in the Corpus of Spontaneous Japanese (S. Furui et al., 2002). Table 5 shows the conditions of acoustic analysis and the specification of HMM (left to right). To improve speech recognition accuracy, acoustic model adaptation was performed. Utterances for adaptation are different from those in the test set, but that speaker who recorded the utterances for adaptation was the same one used in the test set.
Language models were constructed using manual transcriptions of various utterances. Here, to meet open conditions, the language model for recognizing speaker A was constructed by transcriptions of speaker B. Note that the dictionary for speech recognition includes all words spoken by A and B. Thus, the out-of-vocabulary (OOV) rate was zero. For the multi

N-gram method (corresponding to Eq. (7)), language models were constructed for each speaker and each request intention (request and conversation). As a result of speech recognition, though word accuracy was 42.1%, F-measure of keywords was 0.67.

Sampling rate / Quantization	16 kHz / 16 bit
Feature vector	39-order MFCC
Window	Hamming
Frame size / shift	20 / 10 ms
# of phoneme categories	244 syllable
# of mixtures	32
# of states (vowel)	5 states and 3 loops
# of states (consonant + vowel)	7 states and 5 loops

Table 5. Experimental conditions of acoustic analysis and HMM

4.3.2 Results of system request detection

Experiments of request detection using speech recognition results were also performed using the 10-fold cross-validation method. Four experiments (Multi N-gram, sig-Boosting, Boosting, Confidence) were performed. Multi N-gram is based on Eq. (7). Sig-Boosting is based on Eq. (8). This method is system request detection integrated with speech recognition. Sig-Boosting incorporates not only 1-best results of speech recognition but also hypotheses. Boosting incorporates only 1-best results. In order to compare a conventional method, the experiment using the "confidence" method was performed. This method discriminates system requests based on confidence measures of speech recognition. If the average confidence measure of each word is larger than a threshold, an utterance is discriminated as a system request.

The experimental results are shown in Figure 7. We can see that sig-Boosting method achieved the best performance. Intrinsically, the Boosting method showed high

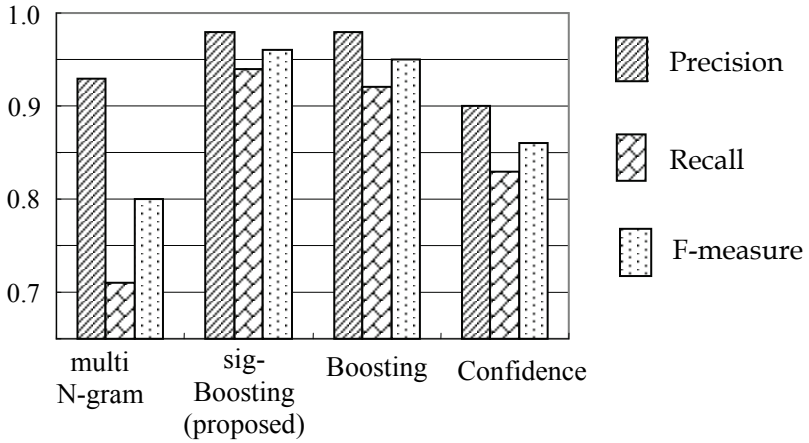


Fig. 7. Results of system request detection using linguistic information

performance. In addition, sig-Boosting recovered false-negative errors by incorporating speech recognition hypotheses. In the case where the 1-best results miss important

keywords, considering the hypotheses, the proposed method can recover the keywords from the hypotheses and improved the performance. On the other hand, the multi N-gram method and confidence method could not achieve performance as high as Boosting methods. Especially, these methods tend to mis-classify the utterances whose intention depends predominantly on one word: e.g., "toka" (meaning "etc.").

5. Conclusion

To facilitate natural interaction for a system such as mobile robot, a new system request utterance detection based on acoustic and linguistic features was employed in this chapter. To discriminate commands from human-human conversations by acoustic features, it is efficient to consider the head and tail of an utterance. The different characteristics of system requests and spontaneous utterances appear on these parts of an utterance. Separating the head and the tail of an utterance, the accuracy of discrimination was improved. Considering the alternation of speakers using two channel microphones also improved the performance.
Also we described the system request detection method integrated with a speech recognition system. Boosting was employed as a discriminant method. Its output score is not a probability, though, so the Boosting output score was converted into pseudo-probability using a sigmoid function. The experimental results showed that integration of system request detection and speech recognition improved the performance of request detection. Especially, in the case where 1-best results miss important keywords, the proposed method can recover the keywords from the hypotheses and improve the performance.
In the future, we plan to perform experiments using larger corpus and more difficult tasks. In addition, we will investigate a context-dependent approach for request detection. The consideration of new kinds of features is also the assignments.

6. References

H. G. Okuno, K. Nakadai & H. Kitano (2002). Social interaction of humanoid robot based on audio-visual tracking, *Proceedings of Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems,* LNAI2358, pp. 725-735, 2002.

J. Miura, et al. (2003). Development of a personal service robot with user-friendly interfaces, *Proceedings of Int. Conf. on Field and Service Robotics*, pp. 293-298, 2003.

S. Waldherr, R. Romero & S. Thrun (2000). A gesture based interface for human-robot interaction, *Autonomous Robots*, 9(2), pp. 151-173, 2000.

H. Asoh, et al. (1999). A spoken dialog system for a mobile robot, *In Proceedings of Eurospeech*, pp. 1139-1142, 1999.

R. Stiefelhagen, et al. (2004). Natural human-robot interaction using speech, head pose and gestures, *In Proceedings of Int. Conf. on Intelligent Robots and Systems.*, pp. 2422-2427, 2004.

T. Takiguchi, et al. (2007). Voice and Noise Detection with AdaBoost, *Chapter on Robust Speech Recognition and Understanding, Book edited by M. Grimm and K. Kroschel., I-Tech Education and Publishing,* pp. 67-74, 2007.

S. Yamada, et al. (2005). Linguistic and Acoustic Features Depending on Different Situations – The experiments considering speech recognition rate, *In Proceedings of Interspeech,* pp. 3393-3396, 2005.

T. Kawahara, et al. (1998). Speaking-style dependent lexicalized filler model for key-phrase detection and verification, *In Proceedings of ICSLP,* pp. 3253-3259, 1998.

P. Jeanrenaud, et al. (1994). Spotting events in continuous speech, *Proceedings of ICASSP*, pp. 381-384, 1994.

T. Yamagata, A. Sako, T. Takiguchi, and Y. Ariki (2007). System request detection in conversation based on acoustic and speaker alternation features, *In Proceedings of Interspeech*, pp. 2789-2792, 2007.

M. Omologo & P. Svaizer (1996). Acoustic source location in noisy and reverberant environment using CSP analysis, *Proceedings of ICASSP,* pp. 921-924, 1996.

R. Schapire, et al. (1998). Boosting the margin : A new explanation for the effectiveness of voting methods, *Annals of Statistics*, vol. 26, no. 5, pp. 1651-1686, 1998.

R. Schapire, et al. (2000). BoosTexter : A Boosting-based System for Text Categorization, *Machine Learning*, 39(2/3, pp. 135-168, 2000.

S. Furui, et al. (2002). BoosTexter : Spontaneous Speech : Corpus and Processing Technology, *The Corpus of Spontaneous Japanese*, pp. 1-6, 2002.