Tandem Feature Extraction for Automatic Speech Recognition

Sunil Sivadas
B.Tech., University of Calicut, 1995
M.Tech., Indian Institute of Science, Bangalore, 1998

A dissertation submitted to the faculty of the
OGI School of Science & Engineering
Oregon Health & Science University
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
in
Electrical Engineering

November 2004

The dissertation "Tandem Feature Extraction for Automatic Speech Recognition" by Sunil Sivadas has been examined and approved by the following Examination Committee:

Hynek Hermansky
Professor
Department of Biomedical Engineering
Oregon Health & Science University
Thesis Research Adviser

John-Paul Hosom
Assistant Professor
Department of Biomedical Engineering
Oregon Health & Science University

Nelson Morgan
Director, ICSI
Professor, Dept. of EECS
University of California, Berkeley

Misha Pavel
Professor
Department of Biomedical Engineering
Oregon Health & Science University

Dedication

To Amma, Achan and Sindhu

Acknowledgments

First of all, my thanks go to my advisor, Prof. Hynek Hermansky, not only for his continuous support and encouragement throughout my PhD years, but also for giving me the freedom to spend part of this time abroad. I would like to thank all the members of thesis committee for their invaluable suggestions that improved the quality of the thesis.

I am furthermore grateful to the past members of the Anthropic Signal Processing (ASP) Group at OGI, Sachin, Pratibha and Andre for their cooperation and for generally making this group a fun place to work at. Other people who I would like to thank for having helped make my time at the ASP group such an ideal experience include Lukas Burget, Frantisek Grezl, Petr Motlicek, Pavel Matjeka and Petr Schwarz. Special thanks to Pavel for maintaining the machines and providing an infrastructure for competitive research.

Last part of this work was completed while I was a research assistant in the Speech Group at IDIAP, Martigny, Switzerland. I am grateful to Herve Bourlard and Hynek Hermansky for making this possible. I benefited greatly from discussions with members of the Speech Group and gratefully acknowledge the use of their resources. I would like to thank Misha Pavel and Nelson Morgan for coming to Switzerland for thesis defence and Paul Hosom for agreeing to participate by teleconference.

Ultimate credit goes to my parents, this thesis would not have been completed without their emotional support throughout my education. Finally I would like to thank Manju for her love and support and understanding.

Contents

List of Tables

List of Figures

Abstract

Tandem Feature Extraction for Automatic Speech Recognition

Sunil Sivadas

Supervising Professor: Hynek Hermansky

For ASR and many other applications, feature extraction transforms the signal to a feature vector that is modeled by the subsequent stochastic classifier. One of the shortcomings of the current feature extraction methods is that they are susceptible to recording channel and speaker characteristics. This has been addressed by learning a transformation in the initial feature vector to suppress the detrimental factors. Linear and nonlinear transformations are studied. This thesis explores the transformation of features to functions of posterior probabilities of phonemes which are modeled by the subsequent stochastic classifier. Further, various models for transformation, such as generative and discriminative, are studied. A discriminative transformation was found to be superior to generative transformation. Different architectures for nonlinear discriminant features are tried. A hierarchical tree structure is found to give marginal reduction in word error rate over a monolithic transformation. The problem of target class selection for feature training is investigated. In a context dependent phoneme classifier based speech recognition system, context sensitive targets are found to be advantageous. The posterior probabilities are estimated using a three layer multi-layer perceptron (MLP). Outputs of the MLP are used as features to a hidden Markov model (HMM) based classifier. The arrangement of two classifiers, a MLP and HMM, in cascade is called Tandem feature extraction.

Chapter 1

Introduction

Speech is the natural medium of communication among humans. We are able to recognize it effortlessly in most difficult environments. It is reasonable to think that human-machine communication could be improved by using speech as a medium(modality) of communication. But speech recognition by machines is not yet reliable enough to replace other modalities of interactions. Although Automatic Speech Recognition (ASR) is an esoteric problem, it has progressed significantly from "Radio Rex" in 1914, one of the first toys to use speech recognition in which a dog jumps out of its hut when its name is shouted, to where it is being used in many applications, such as dictation, telephone call automation and command-and-control. Although ASR enjoys considerable success, it has not reached the performance of humans in realistic environments. Most of this success has been due to designing systems specific to tasks and environments. In a survey by Lippman [69], he showed that the performance of ASR systems degrade drastically in adverse environments. ASR systems have improved since that time, but the degradation under adverse conditions still a issue preventing ubiquitous deployment of ASR systems. The most successful approach to ASR so far has proven to be the one based on statistical pattern recognition [46]. The pattern recognition approach is illustrated by a simplified block diagram shown in Figure 1.1.

In the first step, acoustic analysis is performed on the speech signal using a sequence of windows, resulting in a set of acoustic parameters once every ten millisecond. Usually, some form of spectral analysis, providing a smooth envelope of the speech spectrum, is the preferred method of acoustic analysis.

In the second step, the likelihood of acoustic parameters for the unknown speech being

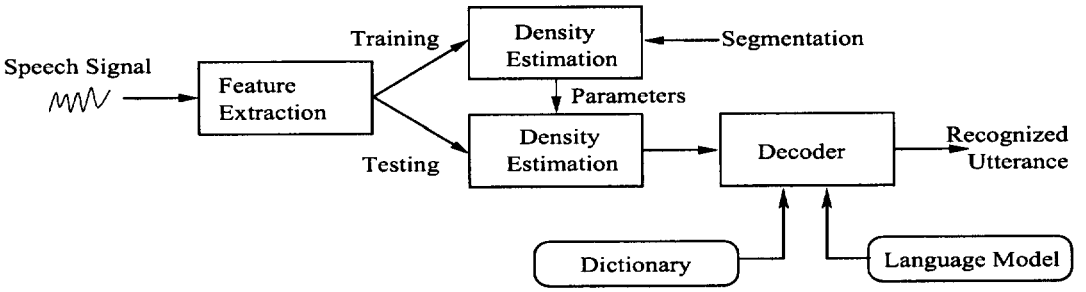

Figure 1.1: Block diagram of speech recognition system

generated by an acoustic model, learned from a large collection of labeled speech utterances from many speakers using a Hidden Markov Model (HMM) based training procedure, is computed. Finally, in the last step, the likelihood scores are augmented with higher level knowledge about the speech utterance derived from a language model, the context, or task semantics, to produce the recognized pattern with the highest likelihood score.

1.1 Feature extraction

The first transformation applied to speech signal in ASR is *feature extraction.* In feature extraction, the signal is transformed (using a linear or nonlinear transformation) to a reduced dimension space. There are several reasons for performing feature extraction:

- to reduce the bandwidth of the input data;
- to provide a set of relevant set of features for a classifier, resulting in improved performance, particulary for simple classifiers;
- to reduce redundancy;
- to recover new meaningful underlying variables or features that describe the data, leading to greater understanding of the data generation process;
- to produce a low-dimensional representation (ideally in two dimensions) with minimum loss of information so that the data may easily be viewed and relationships and structure in the data identified.

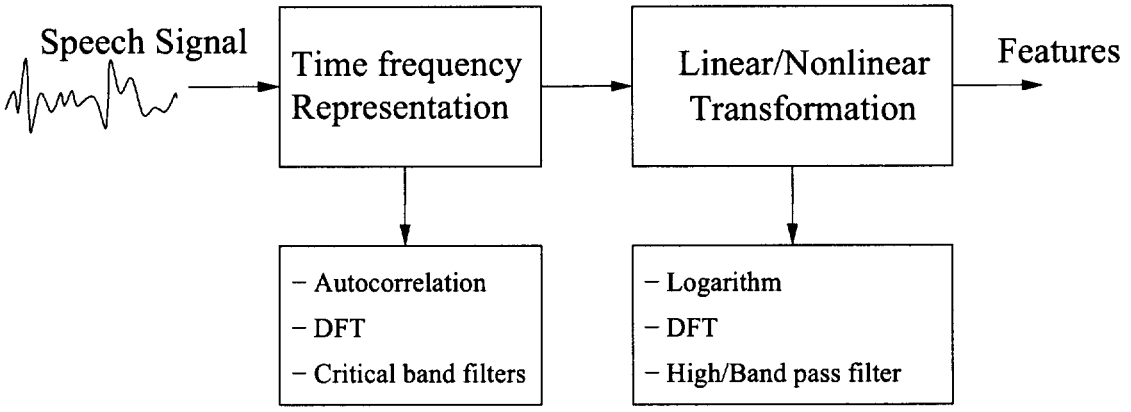

Figure 1.2: Block diagram of feature extraction.

Due to all the reasons listed above, feature extraction is a critical step in ASR. Since information lost can never be recovered, the redundancy reduction has to be carefully designed. In principle, feature extraction is not required, if the stochastic classifier has enough degrees of freedom and is trained with large amount of training data containing all the possible speaker variability and distortions introduced by communication channel.Since both the conditions are not physically attainable, features play an important role in improving the performance of classifiers. There has been numerous works in the past addressing various issues in feature extraction. The feature extraction technique may be supervised (make use of class label information) or unsupervised. Most of the current feature extraction techniques have an initial processing stage based on partial knowledge of human speech production and perception, followed by a transformation derived from data. Figure 1.2 shows a diagram of common current feature extraction techniques.

Linear Predictive Coding (LPC) is an example of calculating features based on speech production [35]. Mel-Frequency Cepstral Coefficients (MFCC) [64] and Perceptual Linear Prediction (PLP) [24] are inspired by the properties of human auditory system. Subsequent transformation could be linear or nonlinear and supervised or unsupervised. This transformation is primarily to suppress the undesired variabilities in the signal and to enhance the desired variabilities that are beyond the scope of the previous transformation. In the case of speaker-independent ASR,the useful source of variability is the inter-class variability of the classes, typically subword units, modeled by the stochastic classifier and

the undesired variabilities are inter-speaker and communication channel variabilities. Although the knowledge about human speech production and perception help us in designing features, [64, 24] it does not provide robustness against variabilities affecting the features. An example of transformation is Cepstral Mean Subtraction (CMS) [47] which provides robustness to variabilities introduced by the recording instrument or communication channel. *This thesis focuses on the design of transformation.*

1.2 Contributions of this work

The main hypothesis of this work is that the best way to derive the transformation is from data itself. The most compact set of features to discriminate between N classes are $N-1$ class conditional *aposteriori* probabilities [40]. It has been shown that using functions of class conditional *aposteriori* probabilities estimated from data can be used as features in HMM based ASR systems [31]. There are various issues in estimating the *aposteriori* probabilities.

- **How to estimate them?** *A posteriori* probabilities are non-linear functions of the observation. Estimating them reliably from data could be very difficult. Since the estimator has to learn the non-linear function from data, there could be generalization issues due to overfitting of the data. *A posteriori* probabilities can be estimated using discriminative or maximum likelihood learning techniques.

- **What are the classes to be used?** One would expect the best set of classes are the ones that closely matches the classes used in the subsequent stochastic classifier that models the distribution of the *aposteriori* probabilities. Typically HMM based recognisers use triphone models. If we assume there are approximately 50 phonemes in English, the number of context dependent triphones are $50^3 = 125,000$. Not all the possible contexts occur due to phonotactics of the language. Although it is possible to maximize the discriminability among such a large number of classes, one usually suffers from curse of dimensionality while modeling a feature vector of thousand dimensions. Also it requires an estimator with large number of free parameters.

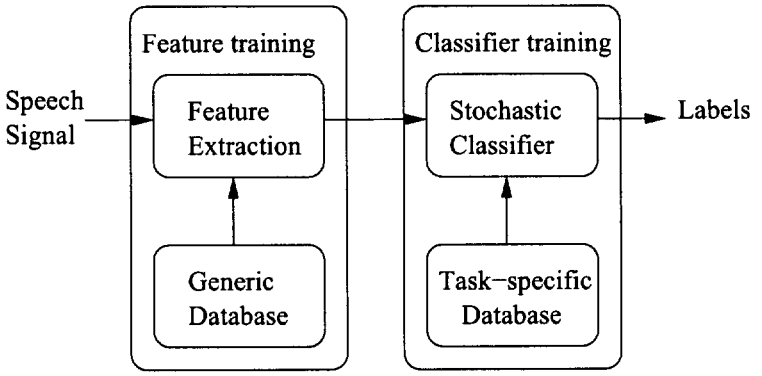

Figure 1.3: A block diagram of data-guided feature extraction.

Figure 1.3 shows a block diagram of data-guided feature extraction. The contribution of this thesis can be summarized as follows

- **Function of *a posteriori* probabilities as features for stochastic classifier** This thesis studies various models for estimating the *aposteriori* probabilities. We show that discriminatively estimating the *aposteriori* probabilities is better than maximum likelihood estimation. We investigate various architectures for estimating the posteriors.

- **Discriminative feature extraction** We compare various linear and nonlinear discriminative feature extraction methods, such as Linear Discriminant Analysis (LDA) [40, 68], and Multi-Layer Perceptrons (MLP) [41]. We show that nonlinear discriminative feature extraction is better than the linear discriminative techniques.

- **Estimation of features from task independent data** Ideally we want to train the feature estimator once and reuse it on various ASR tasks. For that the features have to be trained on a large enough database that contains all the recording conditions that you may encounter in test conditions. We show that the tandem features generalize by training the MLP on task independent database and training the HMMs on task dependent database. The best performance is obtained by training features and HMMs on the task specific database.

- **Choice of classes in data-guided feature estimation** We study the effect of

using context independent and context dependent classes in calculating the features. We derive new target classes by clustering context independent phonemes and show that they perform equally good as context independent phoneme targets. Use of context dependent phoneme targets provide only marginal improvement over context independent targets.

1.3 Organization of the thesis

Chapter 2 reviews feature extraction methods used in automatic speech recognition. We review the most commonly used short-term features based on the Short Time Fourier Transform of a speech signal, and various auditory like modifications applied to it before applying to a hidden Markov Model (HMM) based classifier. We describe data-guided feature extraction and various design issues involved. Two of the most popular data-guided feature extraction methodologies are explained, one based on parametric representation of input features and other based on connectionist approach. We review some of the existing data-guided feature extraction methods such as Linear Discriminant Analysis (LDA), Tandem and TempoRAl PatternS (TRAPS).

In Chapter 3 we explain the philosophy of Tandem feature extraction in detail. Tandem feature extraction uses two, or more, classifiers in tandem. The first one is used to map the acoustic feature vector space to posterior probability space and the second classifier models the distribution of the posterior probability based features.

Chapter 4 studies the various architectures for Tandem. We study discriminative and generative transformations of acoustic feature space. We investigate linear and nonlinear transformations. We compare monolithic versus hierarchical architecture for feature extraction.

Effect of target classes used in feature extraction is studied in Chapter 5. We examine classes based on phoneme segmentation. Specifically we study broad phonetic categories, context independent phonemes, context dependent phoneme based targets.

The effect of training the transformation and stochastic classifier on different types of databases is studied in Chapter 6. We investigate the situation where there is a large

amount of task independent training data available and only a small amount of task dependent training data.

Finally Chapter 7 summarizes the thesis and discusses extensions of the work.

Chapter 2

Data guided features

2.1 Introduction

In an Automatic Speech Recognition (ASR) system, the boundary between feature extraction and stochastic classifier is very fuzzy. Since the stochastic classifier makes assumption about the statistics of feature vectors, such as uncorrelated feature dimensions, which are violated many times, there are many approaches to bridge this gap. Feature extraction reduces the bandwidth, from the sampled signal, of the information flow by reducing the redundancy in the signal. This signal carries information about the speaker, message, and channel. Depending on the task some of the information is redundant or unwanted. For example, speaker independent ASR systems suffers from the variabilities due to the speaker and channel. Since the feature extraction does not completely remove the unwanted variability, the residual variability is modeled by the stochastic model. The performance of the system depends on the efficacy of feature extraction in reducing the redundancy and its ability to model the residual variability. If the stochastic classifier is complex enough to model all the sources of redundancy in the signal directly, feature extraction is not required. Thus as the amount of speech data increases and the number of learnable parameters in the stochastic classifier increases the role of feature extraction becomes less prominent.

In the 70's, ASR systems were purely knowledge-based [84]. They were designed using rules for pattern classification and language modeling. These rules were based on how humans read spectrograms [13, 87]. Although they worked reasonably on small tasks under controlled environment, they were very fragile [7]. Later, in the mid 80's,

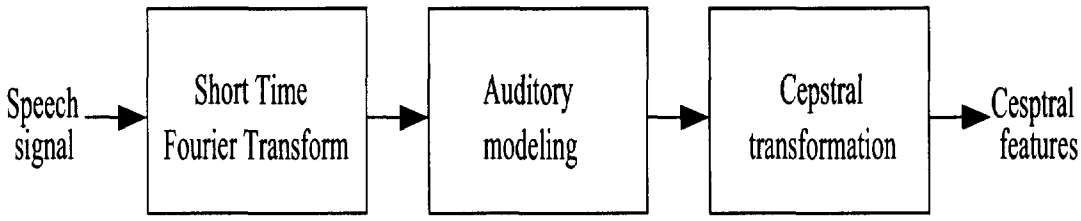

Figure 2.1: Cepstral feature extraction.

the knowledge-based systems were replaced by a combination of knowledge-based and stochastic approaches. Where as the features are largely designed based on prior knowledge about human auditory and production mechanism, the classifiers are based on stochastic modeling techniques such as Hidden Markov Models (HMM) [46] and Artificial Neural Networks (ANN) [60]. Stochastic classifiers learn the parameters of the model directly from the data. Classification is based on the likelihood of scores representing the similarity between the unknown pattern and each of the models.This was a major step towards data-guided approach to ASR.

Although data-guided approaches are used extensively in designing stochastic classifiers, it has not been applied prominently in the design of features until recently. The feature extraction methods for ASR are adopted from speech coding principles. Features are largely designed based on the partial knowledge of how humans produce or perceive speech. Initial research in feature extraction has been influenced by how humans produce speech and speech coding principles. One example is Linear Predictive Coding (LPC) [35].

2.2 Feature extraction

Typically, feature extraction for ASR consists of the processing steps shown in Figure 2.1. As a first step, most of the feature extraction systems converts the speech signal into a vector of spectral energy values using Short Time Fourier Transform (STFT) [51]. STFT is computed as

$$S(\omega, t) = \sum_{n=t-k}^{t+k} s(n)h(t-n)e^{-j\omega n} \tag{2.1}$$

where t is the discrete time index and $h(t)$ is a symmetric window applied to the short-time signal and is non-zero only in the interval $-k$ to k. Ohm's law of hearing asserts that human ear is insensitive to phase differences between the frequency components in the signal over a short duration. Typically a Hamming window of 25 ms duration is used. We retain only the magnitude of $S(\omega, t)$. Such a window when applied on a signal which is sampled at 8 kHz will band-limit $S(\omega, t)$ to 80 Hz in t dimension. Hence $S(\omega, t)$ needs to be sampled at 160 Hz [37]. However, in practice, we have observed that sampling at 100 Hz (every 10 ms) makes no difference to the processing. This is because $S(\omega, t)$ has little energy above 50Hz.

To compute energy in spectral components, the squared magnitude of the $S(\omega, t)$ is calculated. Experiments have shown that [16] the cochlear filters have non uniform frequency resolution. Psychoacoustic experiments using simultaneous frequency masking have revealed that the bandwidth of critical bands increases with the center frequency. This property of hearing is simulated in the current feature extraction modules by projecting the short-time power spectrum onto the frequency responses of the simulated critical band filters in Bark scale [24] or Mel scale [76]. This is simulated by,

$$s_k = \mathbf{f}_k^T \mathbf{s}_p \qquad k = 1, 2 \ldots 15, \tag{2.2}$$

where $\mathbf{f}_k$ and $\mathbf{s}_p$ are column vectors representing the frequency response of the kth critical bands and the short-time power spectrum respectively. This operation yields s_k energy output from the kth critical band.

Next, a compressive nonlinearity is applied to model the nonlinearities present in the human auditory system. This nonlinearity is typically logarithm or cubic root of the filter bank outputs.

$$ls_k = \ln(s_k) \qquad k = 1, 2 \ldots 15, \tag{2.3}$$

The compressed spectral energy values are then uncorrelated and dimensionality is reduced.

$$\mathbf{C} = \mathbf{W}.\mathbf{ls} \tag{2.4}$$

where $\mathbf{ls}$ is the vector of 15 logarithmic critical band energies and $\mathbf{W}$ the matrix whose rows are the decorrelating basis function. The optimal bases for decorrelation of features are the

Eigen vectors of the total covariance matrix. They are derived using Principal Component Analysis (PCA) [67]. For random variables generated by a first order Markov processes, projection onto a cosine basis approximately decorrelates the distribution [54, 1, 58]. It has been shown that the bases derived using PCA are very similar to Cosine bases. [26]. The resulting features are known as cepstral features. The cosine bases analyze the "frequency of the spectral values". This is analogous to the "ripple frequency analysis" in the auditory system [43].

Various psychoacoustic experiments on masking have demonstrated that in the human auditory system the effect of a sound can last as long as 200ms [50]. This shows the need for integration of information over longer time spans than the 25ms window size. Time derivatives of cepstral coefficients, known as "delta coefficients", [71] was an initial attempt at incorporating information over longer time spans. The typical time span of filters used in computing delta features is 50-100 ms. RelAtive SpecTrAl (RASTA) [27] processing is motivated by the the observation that modulation frequencies [25] in the range 4-15Hz are most important in the intelligibility of speech. RASTA processing enhances the modulation frequencies that are important while attenuating the the ones that are not carrying linguistic information. The time constant of RASTA filter is about 200ms. It was later shown that RASTA filter can also model forward masking [50].

Fletcher [22, 38] hypothesized that initial recognition of features in human speech processing occur independently in individual critical bands or a group of critical bands. They are later integrated to identify phonemes, subsequently into words and sentences. Sub-band based speech recognition [21, 74] is based on this hypothesis. Saul et. al. [45] designed a feature detector based on this hypothesis. More recently extraction of features independently from critical bands has been shown to be successful in ASR using TempoRAl PatternS (TRAPS) [30, 32]

2.3 Data-guided feature extraction

In the previous section we saw that the features derived from the signal are based on our partial knowledge of human speech processing. Instead we could derive a transformation

from the signal with a little prior knowledge about the constraints, which will also give insight into the nature of analysis and transformations going on inside the human auditory system. The transformation from measurements to a lower dimensional space may be a linear or nonlinear combination of the original variables and may be supervised or unsupervised. The transformation requires the optimization of some criterion function, $J(\cdot)$. The optimization is performed over all possible transformations of the measurements. We seek the transformation $\tilde{T}$, for which

$$J(\tilde{A}) = \max_{A} J(\mathbf{A}(\mathbf{x})) \tag{2.5}$$

where x is the measurement and $\mathbf{A}(\cdot)$ is the set of allowable transformations. The feature vector is then $\mathbf{y} = \tilde{\mathbf{A}}(\mathbf{x})$. The criterion function $J(\cdot)$ is usually based on some measure of distance or dissimilarity between distributions, which in turn may require distances between objects to be defined. There are various issues in designing the transformation $A(\cdot)$.

- **Architecture of feature extractor:** An important issue is the allowable complexity of the function $\mathbf{A}(\cdot)$ that extracts features. If the function $\mathbf{A}(\cdot)$ is unstructured and unconstrained, then unreasonably large amounts of data and computation may be necessary to get a good estimate of the function. Hence in this work we use a family of constrained linear and nonlinear functions to extract features.

- **Objective measure:** The effectiveness of the approach depends on the optimality measure used to select the feature extraction function $\mathbf{A}(\cdot)$. Ideally one would like to maximize the Mutual Information (MI) between the measurement $\mathbf{x}$ and the class C_i to which it belongs to, $I(\mathbf{x}, C_i)$. It has been shown that maximizing the *aposteriori* probability of class C given the measurement $\mathbf{x}$, $P(C_i|\mathbf{x})$, maximizes the MI, $I(\mathbf{x}, C_i)$, under the condition that all the classes $C_i, i = 1, \ldots, N$ are equally likely [20].

- **Learning the transformation:** The ability of the estimated feature extraction method to generalize to new databases will determine the re-usability of the data-guided feature extraction module. For good generalization, the database used for

estimating the feature extraction method should contain a sufficient sampling of all the possible sources of variability that we expect to encounter in the application.

- **Definition of classes:** If the transformation $\mathbf{A}(\cdot)$ is learned in a supervised manner, each measurement $\mathbf{x}$ needs to be labeled as one of the classes $C_i, i = 1, \ldots, N$. In speech recognition the classes used are related to phonemes. It is important to maintain consistency between the classes used in learning the transformation $\mathbf{A}(\cdot)$ and the classes used in the subsequent stochastic classifier.

- **Measurements from signal:** It is important to make minimum *a priori* assumptions about the measurement $\mathbf{x}$. Ideally we would like to use the speech signal itself as the initial representation. This would result in a huge search space for the transformation $\mathbf{A}(\cdot)$. Hence it is important to structure $\mathbf{A}(\cdot)$. An example can be found in [3] where they show the advantage of structuring the transformation $\mathbf{A}(\cdot)$ over unstructured one. We restrict the initial measurements from signal to auditory like features.

In the subsequent sections we review of some of the popular data-guided feature extraction techniques.

2.4 Linear Feature extraction

If the class of transformations $\mathbf{A}(\cdot)$ is specified as linear, we can derive linear features from the measurements. The transformations can be supervised or unsupervised. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are the most common linear feature extraction techniques used in speech. Both are linear transformations based on matrices of first and second-order statistics.

2.4.1 Principal Component Analysis (PCA)

PCA is an unsupervised method based on a correlation or covariance matrix.

$$\mathbf{y} = \mathbf{A}^T.\mathbf{x} \tag{2.6}$$

The transformation $\mathbf{A}$ is orthogonal and maximizes the variances of individual components of $\mathbf{y}$. $\mathbf{A}$ can be derived from the total covariance matrix S_T of the measurements $\mathbf{x}$

$$S_T = \mathbf{A}\Sigma\mathbf{A}^T \tag{2.7}$$

where

$$S_T = E[\mathbf{x}\mathbf{x}^T] - E[\mathbf{x}]E[\mathbf{x}^T] \tag{2.8}$$

and

$$\Sigma = diag[\sigma_1, \sigma_2, ..., \sigma_p] \tag{2.9}$$

is a matrix whose diagonal consists of the eigenvalues of S_T and columns of $\mathbf{A}$ are the eigenvectors of S_T. Detailed derivation and properties of PCA can be found in any pattern recognition textbook [67].

The most common application of PCA in speech are to uncorrelate the measurements x and to reduce dimensionality. It has been shown that if $\mathbf{x}$ is generated by a first order Markov process, then the principal components are Cosine bases [1]. Cosine bases are used extensively in uncorrelating logarithm of filter bank energy values, to compute Mel Frequency Cepstral Coefficients (MFCC) [76].

2.4.2 Linear Discriminant Analysis (LDA)

LDA is a supervised method to find a linear combination of the measurements that maximizes linear separability of the classes. The criterion proposed by Fisher [65] is a function of the ratio of between-class to within-class covariance matrices. The transformation maximizes between-class separability and minimizes within-class separability. It transforms $\mathbf{x}$ to a space of dimension at most $C-1$, where C is the number of classes. The transformation assumes that the data is normally distributed. In the transformed space, the dimensions are ordered in terms of 'importance of discrimination'. The within-class covariance represents how much the samples within a class vary. Between-class covariance matrix is the covariance of the class conditional means. It gives a measure of the separability of class conditional means and hence overlap between classes in feature space..

A property of between-class covariance matrix and within-class covariance matrix is that they sum up to the total covariance and hence LDA can be viewed as decomposing

the total variability into within-class and between-class variability. A widely used criterion for class separability is defined by

$$F = trace(S_w^{-1} S_b). \tag{2.10}$$

where S_w and S_b are within-class covariance matrix and between-class covariance matrix respectively. It has been shown that the set of basis vectors ϕ_i, that satisfies the following generalized eigen-value problem

$$S_b \phi_i = \lambda_i S_w \phi_i, \tag{2.11}$$

maximizes F. The matrix S_b has a maximum rank of $L - 1$. This constrains the rank of $[S_w^{-1} S_b]$ also $L - 1$. Hence the maximum number of linearly independent basis functions obtained from the above equation is $L - 1$.

The basis functions computed using LDA are optimal only under the following assumptions:

- The class conditional distributions are normal.
- All the class-conditional covariances are equal.
- Only two classes are involved in the classification.

If the above conditions are met, then the basis functions given by LDA are guaranteed to be optimal and the classification error will be equal to the Bayes' error bound. When these conditions are not met, dimensionality reduction using discriminant basis vectors can increase the overlap of class conditional distributions. This in turn causes an increase in the lowest theoretical achievable error. But practically, even for multi-modal distributions, $trace(S_w^{-1} S_b)$ has been found to be a good measure for class separability unless the class conditional means are all the same [39].

LDA has been studied by Hunt and colleagues [54, 48, 49, 47] to reduce the dimensionality of the various features. They used states of HMMs as classes to compute the within- and between-class covariances. These statistics were derived from trained HMMs representing words. Discriminant analysis was also used by Brown to process several concatenated feature vectors, thus addressing both temporal and spectral dimensions [62].

Doddington used a state specific discriminant transformation for improving the recognition performance. For each of the HMM states a transformation which enhances the discrimination between that state and other confusable states was derived [17]. A comparison of various linear transformations as alternatives to the DCT can be found in [80, 14]

2.5 Nonlinear feature extraction

An ideal feature extractor should be able to reduce the error to its theoretical limit, which is given by Bayes' error [39]. For an L class problem, the Bayes' classifier that yields minimum error compares L a posteriori probabilities, $p(\mathbf{x}|C_1)$, $p(\mathbf{x}|C_2)$ $\ldots p(\mathbf{x}|C_L)$, and classifies $\mathbf{x}$ to the class that gives maximum *a posteriori* probability. Another interesting point is that the a posteriori probabilities are not linearly independent, since

$$\sum_{i=1}^{L} p(\mathbf{x}|C_i) = 1 \tag{2.12}$$

Only $L-1$ of the L posterior probabilities are linearly independent. Hence these $L-1$ linearly independent features are the ideal set of features that would give an error which is equal to the error given by a Bayes' classifier. That is, by transforming the original random vector $\mathbf{x}$ into an $L-1$ dimensional vector by using the transformation, $y_i = p(\mathbf{x}|C_i)$, i = 1, 2, ... $L-1$, no classification information is lost. Even though Bayes' error is the best criterion and the *a posteriori* probabilities the best features, *a posteriori* probability functions are hard to estimate without severe biases and variances.

There are many methods in literature to estimate the *a posteriori* probabilities of classes. They can be broadly classified as *generative* and *discriminative*. The generative models maximize the likelihood of the data with respect to the parameters of the model, where are discriminative models maximize the separability of the classes. Two of the most common statistical modeling techniques for estimating *a posteriori* probabilities are parametric models and connectionist models. In general the nonlinear feature extraction can be formulated as

$$\mathbf{y} = \mathbf{W}^{\mathbf{T}}\phi(\mathbf{x}) \tag{2.13}$$

Where $\mathbf{W}$ is the $C \times m$ matrix with (ij) component w_{ij}, $\phi(\mathbf{x})$ is the m-dimensional vector

with ith component $\phi_i(\mathbf{x})$. $\mathbf{y} = \phi(\mathbf{x})$ is a nonlinear transformation of $\mathbf{x}$. In the subsequent sections we review two methods. The parametric models estimate the *a posteriori* probabilities indirectly using the Bayes' rule.

$$p(C_i|\mathbf{x} = \frac{p(\mathbf{x}|C_i)) * p(C_i)}{p(\mathbf{x})} \tag{2.14}$$

where $p(\mathbf{x}|C_i)$ is the likelihood of feature vector $\mathbf{x}$ given the model for class C_i, $p(C_i)$ is the *apriori* probability of the class C_i and $p(\mathbf{x})$ is the *apriori* probability of feature vector $\mathbf{x}$. Connectionist models estimate *a posteriori* probabilities directly from data by learning the nonlinear function $\phi_(\cdot)$.

2.5.1 Gaussian Mixture Models (GMM)

If the function $\phi_\cdot$ in equation 2.13 are Gaussian densities, then we have a Gaussian Mixture Model (GMM). GMM is a parametric model of estimating density of the data.

$$p(\mathbf{x}|C_i) = \sum_{j=1}^{M} p_{ij}\mathcal{N}_{ij}(x; \mu_{ij}, \Sigma_{ij}), \tag{2.15}$$

where p_{ij} are mixture weights, $N_{ij}(\cdot)$ are Gaussian functions and M is the number of Gaussian functions used for modeling the distribution. The Gaussian functions are each parameterized by a mean vector, μ_{ij}, and a covariance matrix, Σ_{ij}. The parameters are estimated in maximum likelihood manner from data using the well known Expectation Maximization (EM) algorithm [2]. The structure of GMM is specified by the number of components M within each class, the constraint on the covariance matrices Σ_{ij}. The number of components depends on many factors including shape of clusters, separation, relative sizes, sample size and dimension of data. We use a rule of thumb, where the ratio of number of observations to number of variables is ten.

GMMs can be trained in a discriminative manner also. LDA is equivalent to maximum likelihood classification assuming Gaussian distributions for each class with different means and common covariance matrix. This is extended to GMMs by having separate means, μ_{ij}, within a class, but the covariance matrix, Σ, is common across all mixture components and across all classes [82]. This is also one way of restricting the number of parameters to

be estimated. In *discriminant* analysis by GMMs, the class-conditional density for class ω_j is expressed as

$$p(\mathbf{x}|C_i) = \sum_{j=1}^{M} p_{ij}\mathcal{N}_{ij}(x; \mu_{ij}, \Sigma), \tag{2.16}$$

If the common covariance matrix Σ is diagonal, then we have discriminant functions similar to Radial Basis Functions (RBF) [11].

GMMs are used extensively in acoustical modelling [9, 46] for speech and speaker recognition. Recently they have been used in feature extraction also. In [56] a GMM based front-end is used to extract spectral peak information. Another architecture, which is a two dimensional HMM, known as HMM2 [44] is used to extract formant features. Eide. et. al [15] used GMM to extract distinctive features. The output of GMMs are used as features to subsequent HMM based classifier. In this thesis we use GMMs to estimate likelihood of measurement $\mathbf{x}$ given the class models. The likelihoods are used as features to a GMM based classifier.

2.5.2 MultiLayer Perceptron (MLP)

The MLP is a model that, in its simplest form, can be regarded as a generalized linear discriminant function in which the nonlinear function ϕ in equation 2.13 is flexible and adapt to the data. The basic MLP produces a transformation of observation $\mathbf{x}$ from a p dimensional space to an n dimensional space according to

$$g_j(\mathbf{x}) = \sum_{i=1}^{m} w_{ij}\phi_i(\alpha^{\mathbf{T}}\mathbf{x} + \alpha_{i0}) + w_{j0} \tag{2.17}$$

The functions ϕ_i are fixed nonlinearities, usually identical and of the *logistic* form.

$$\phi_i(z) = \phi(z) = \frac{1}{1 + exp(-z)} \tag{2.18}$$

Thus, the transformation consists of projecting the data onto each of m directions described by the vectors α_i then transforming the projected data (offset by a bias α_{i0}) by the nonlinear functions $\phi_i(z)$ and finally forming a linear combination using the weights w_{ji}.

A diagrammatic representation of MLP is given in Figure 2.2. In this thesis we use

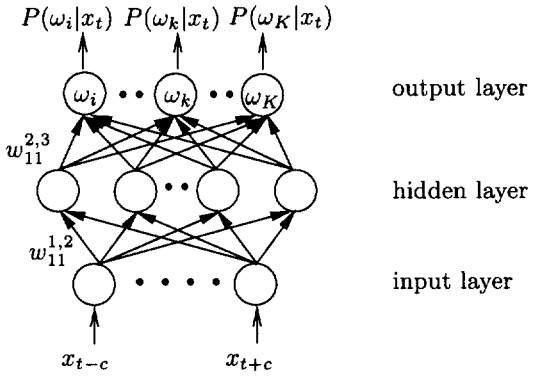

Figure 2.2: Illustration of a Multi-Layer Perceptron (MLP) with one input layer, one hidden layer and an output layer giving posterior probability estimates for each class.

a MLP with a single hidden layer to nonlinearly transform measurements $\mathbf{x}$. It has been shown that an MLP with a *single* hidden layer can approximate an arbitrary (continuous bounded integrable) function arbitrarily accurately [41]. The 'outputs' are a linear combination of the functions ϕ_i.

The MLP is a nonlinear model: the output is nonlinear function of its parameters and the inputs. The optimization criterion used to train the parameters is the minimization of classification error on a training and cross-validation databases. The parameters of the basis functions, as well as the linear weights are determined by back-propagation training procedure [10].

An MLP with enough number of learnable parameters and a softmax output layer, when trained with "1 from K" target coding can estimate the posterior probability of the classes given the input features [19, 20]. We use the output of the MLP, after some further processing, as features in a HMM based speech recognition system.

2.6 Description of various data-guided feature extraction approaches for ASR

Several researchers have applied the data-guided feature extraction techniques explained in the previous sections to speech.

2.6.1 Spectral LDA

Malayath [57] applied LDA to derive basis functions to project logarithmic critical band energy values. The discriminant basis corresponding to the largest eigenvalues were found to be significantly different from the DCT basis and that derived by PCA. Since phonemes are used as the basic units in many speech recognition systems, linear discriminant analysis (LDA) of the critical-band spectrum was carried out by with context independent phonemes as classes. They also showed that the features extracted by the alternative spectral basis functions offer advantage in phoneme classification of spectral vectors. The LDA-derived basis functions outperformed the conventional DCT in connected digit recognition tasks. The zero-crossings of the LDA-derived spectral basis functions are reasonably uniformly spaced on the Bark scale of the auditory-like critical-band spectrum. They investigated the suitability of Mel/Bark frequency warping for phoneme recognition. The shape of the discriminant vectors computed using LDA determines the spectral variations that are important in discriminating phonemes. They observed that the discriminant vectors analyze the low frequency part of the spectrum with higher resolution than the high frequency part. This trend is consistent with the properties of auditory-like filters used by the conventional Mel/Bark filter-bank analysis. It was shown that the sensitivity of the features extracted by these basis functions is approximately inversely proportional to the formant frequency. This trend in sensitivity is consistent with the properties of hearing. Non-uniform frequency resolution in the form of critical-band filters is mainly used in ASR systems to emulate the properties of basillar membrane in human ear. Using discriminant analysis they established a link between non-uniform frequency resolution, phonetic discrimination and the spectral properties of vowel-like sounds.

2.6.2 Temporal LDA

Using a well known statistical analysis method Multivariate ANalysis Of VAriance (MANOVA), Kajarekar et.al. [72] observed that the variability due to phones spreads for approximately 250 ms around the current frame. This is in agreement with the time constant of the temporal integrator in human auditory system. They included this variability in the design of

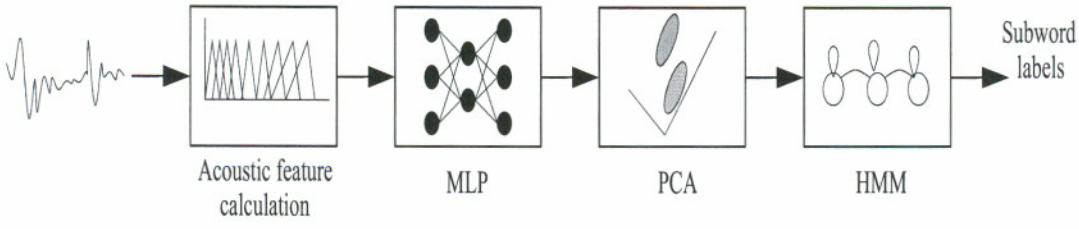

Figure 2.3: A block diagram of the application of tandem feature extraction scheme in ASR.

features using Linear Discriminant Analysis (LDA). Two types of analysis were performed. First analysis, called joint analysis, uses all the correlations in a block of spectrogram. Second analysis, called combined analysis, assumes that time and frequency domains are independent. The discriminant features from both analysis are used in speech recognition experiments. The results showed that features from joint analysis performed worse than combined analysis because joint analysis suffered from the lack of training data and did not generalize on the test data. They showed that performance of the speech recognition system improves when information from a longer time-span is included in the features.

2.6.3 Tandem

Tandem is a nonlinear data guided feature extraction method [31]. A MLP is discriminately trained using a labeled data-set to estimate posterior probabilities of phonemes classes given the measurements. The posterior probabilities are used as features in a GMM/HMM based ASR system. The distribution of the posteriors is heavily skewed and these posteriors are correlated. To make them suitable for HMM based recognizer, first a nonlinearity is applied to the posterior probability feature vector to make the distribution smooth, then they are decorrelated using PCA. Unlike LDA, MLP makes little assumptions about the statistical properties of input features. Figure 2.3 shows a block diagram of the application of tandem feature extraction scheme in ASR.

Since the features are posterior probabilities, it is possible to combine multiple MLPs trained on different transformations of the speech signal. If individual MLPs are trained on input representations that carry *complementary information*, a lot of gain can be obtained by combining their outputs [73]. Ellis et al. [6] used conditional mutual information (CMI)

as a tool to estimate the amount of information that one feature stream contains about the other, given knowledge of the correct subword unit label.

2.6.4 TempoRAl PatternS (TRAPS)

Based on the Fletcher-Allen model of human speech recognition [22, 38], features or acoustic events are first detected in individual critical bands. They are then integrated at a later stage to detect phonemes. Hermansky et.al. [30, 32] investigated the use of frequency-localized temporal patterns of the speech signal for developing robust front-end in hybrid ANN/HMM based speech recognition system. It was later extended into GMM/HMM based system using Tandem feature extraction [33]. Jain et.al proposed categories that are based on the shape of the temporal patterns. Instead of labeling each pattern in each critical band with the phoneme corresponding to it, they proposed categories that represent a small number of distinct speech-events. These categories are obtained by clustering the mean temporal patterns of context-independent phonemes using an agglomerative hierarchical clustering technique [28]. They showed that integrating information across multiple critical bands, up to three, was beneficial [29].

2.7 Summary

In this chapter we first reviewed various feature extraction methods. We explained data-guided feature extraction methodology and examined design issues. We summarized a few popular data-guided feature extraction methods.

Chapter 3

Tandem feature extraction

The motivation for Tandem feature extraction method is to use functions of posterior probabilities of abstract events in speech as features to a stochastic classifier. A data-guided feature extraction method for deriving features for ASR based on HMM was proposed by Hermansky et. al [33]. The technique used a Multi-Layer Perceptron (MLP) trained on the task-specific training data or task-independent development data to derive mapping from several (possibly different) feature sets to logarithmic posteriors of context-independent phonemes. The log likelihoods are then further transformed through a Karhunen-Loeve (KL) transform, also known as Principal Component Analysis (PCA) to a set of features for subsequent HMM modeling. The proposed technique was known as Tandem feature extraction since it has two acoustic models in tandem - first a neural-net then a GMM. The Tandem approach performed significantly better than either the hybrid or conventional baseline features such as PLP, MFCC [31, 73] on the Aurora noisy digits task [23], achieving an average 50% relative error rate reduction. Figure 3.1 shows various block diagrams of hybrid HMM/ANN, HMM/GMM and Tandem speech recognition systems.

Current speech recognition systems consists of three main stages. First , features are computed from the speech waveform to generate relatively compact feature vectors at a frame rate of 100 Hz. Secondly, these feature vectors are fed to a stochastic model which has been trained to assign feature vectors to speech units; commonly, this is realized as a set of Gaussian Mixture Models (GMMs) of the distributions of feature vectors corresponding to context dependent phonemes. Finally, the output of these models provides the relative likelihoods for the different speech sounds needed for a hidden Markov model (HMM) decoder, which searches for the most likely allowable word sequence. The stochastic

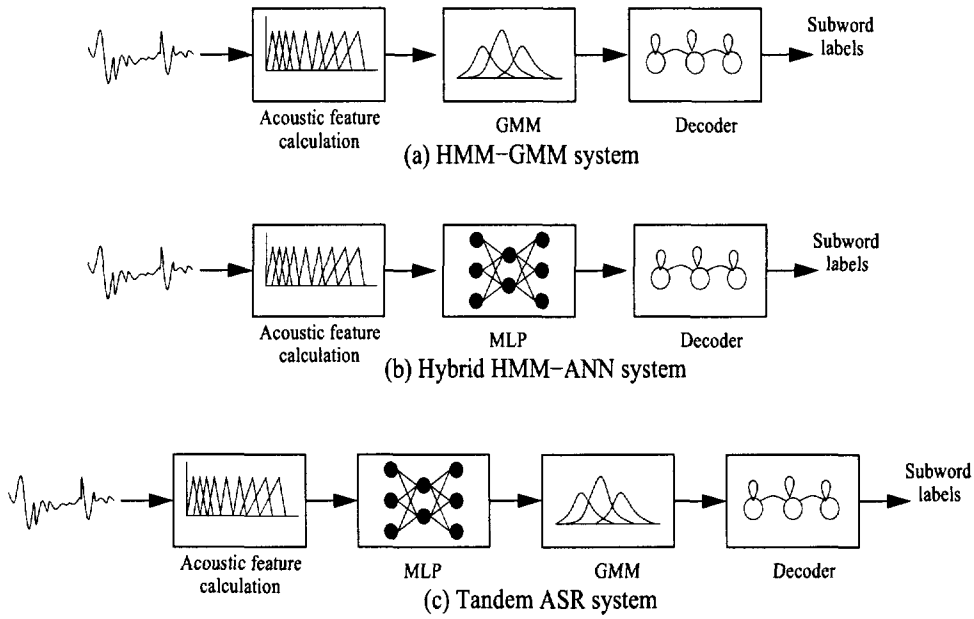

Figure 3.1: Block diagrams of (a) Hybrid HMM-ANN system (b) HMM-GMM system and (c) Tandem.

model is trained using a corpus of examples that have been manually or automatically labeled. This is done according to a Maximum Likelihood (ML) criteria using the EM algorithm. We would rather have a discriminative criterion that optimized the ability to distinguish different classes, rather than just the match within each class. The hybrid HMM-ANN framework [20] replaces the GMM stochastic model with a neural network (NN), discriminatively trained to estimate the posterior probabilities of each subword class given the data. Hybrid systems have been shown to have comparable performance to GMM-based systems for many corpora.

This is not the first time that neural networks have been proposed as feature preprocessors for speech recognition. Bengio et al. [86] suggested using them to increase state likelihoods in HMM systems, and Rigoll et al. [18] showed significant improvements from an MLP inserted as a feature preprocessor into a previously-trained Gaussian-model HMM system, again training the net based on the HMM state. Fontaine et al. [83] use the first 3 layers of a four-layer net as a form of Non-Linear Discriminant Analysis (NLDA), to emphasize the relationship to the better known linear discriminant analysis (LDA). They

achieved a 20-25% relative error reduction for the Phonebook large-vocabulary isolated-word corpus.

3.1 Methodology

First, a hybrid connectionist-HMM system is trained , which amounts to training the neural network acoustic model (a conventional multi-layer perceptron (MLP) structure with one hidden layer) to estimate the posterior probabilities of each possible subword unit. The network is trained by backpropagation with a minimum-cross-entropy criterion to one-hot targets obtained from either hand labeling or a forced alignment of the training data generated using an earlier acoustic model. (For the results below, the entire training and realignment process was repeated several times to stabilize the labels). Input to the MLP is nine frames, four frames from past and four frames in future, of 8 PLP cepstral coefficients, 8 delta and 8 double delta features (24x9=216) after utterance based mean subtraction. The output of the neural network is a vector of posterior probabilities , with one element for each phone; one such vector is generated for context windows centered on each input feature vector. Conventionally (for an HMM/ANN hybrid system) these would go directly (after division by priors) to an HMM decoder to find the word sequence , but instead we use them as the feature inputs for a Gaussian-mixture-based HMM system. Typically, the number of phones is between 30 and 50, so the total dimensionality of the feature space is much the same as with normal features augmented by deltas and double-deltas. Figure 3.2 shows spectrogram of a waveform, its auditory version using PLP processing and the posterior probability outputs of MLP. The figures on the left column illustrate the time-frequency representation of clean waveform and the output of MLP trained on a mixture of clean and noisy speech. The figures on the right column are the corresponding noisy versions of the same utterance. It can be seen that the outputs of the MLP are not affected much by the decrease in signal to noise ratio of the waveform.

Because the posterior probabilities have a very skewed distribution, we find it advantageous to warp them into a different domain, for instance by taking their logs. An alternative to this is to omit the final nonlinearity in the output layer of the neural network

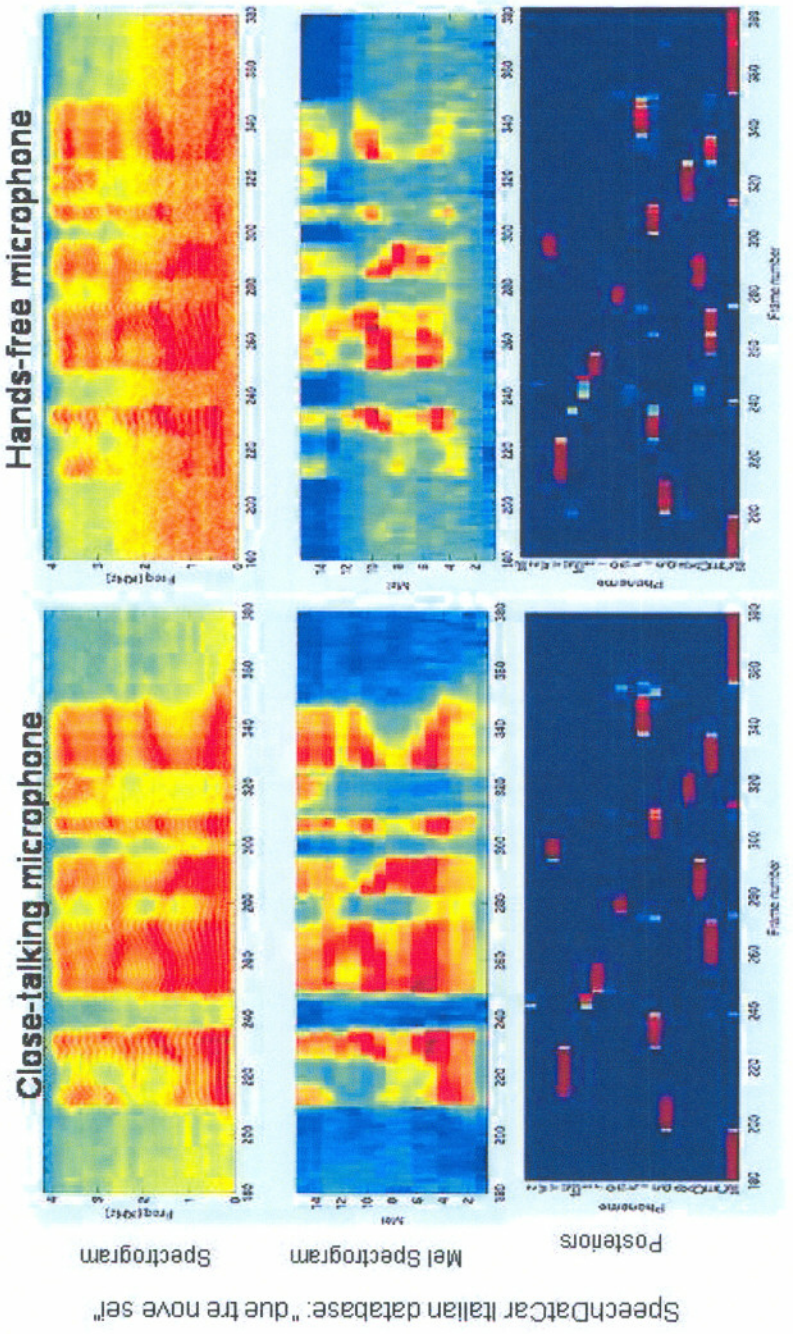

Figure 3.2: Time-feature plot of spectral features and tandem features.

during forward pass. We use the softmax nonlinearity in this position (exponentials normalized to sum to 1, so simply skipping it is very close to taking the log of the subsequent probabilities. Figure 3.3 shows the histogram of posterior probabilities, pre-nonlinearity outputs and log of the posterior probabilities. These are the two outputs, corresponding to "silence" and phoneme "ah", of a MLP trained to classify 23 monophones. It can be observed that the pre-nonlinearity outputs are more "gaussian like" distributed.

The features constituted by the log-posteriors have the rather unusual property of tending to contain one large value (corresponding to the current phone) with all other values much smaller. We find that applying a global decorrelation via the Karhunen-Loeve (KL) transform improves system performance, presumably by improving the match of these features to the Gaussian mixture models. Figure 3.4 shows the histograms of the first two cepstral coefficients C_0 and C_1 and the histogram of first two tandem features after KL transform. The two dominant modes in the histogram of the first tandem features are for speech and non-speech classes.

What makes functions of posterior probabilities useful features? This has been addressed in the seminal paper on tandem feature extraction [31]. Figure 3.5 shows the scatter diagram of features belonging to "speech" and "silence" classes. The top figure shows how two cepstral features $X = (C_0, C_1)$ are distributed, the middle one illustrates the distribution of posterior probabilities,$p(speech|X)$ and $p(silence|X)$, and the bottom plot shows how the log posterior probabilities are distributed. The feature distributions of two classes are more separable in the posterior feature space than the cepstral feature space. Here, the MLP maps the input feature space, that is cepstral coefficients, to posterior feature space in which the boundaries between classes are magnified. This implies a low entropy regions in feature space. The overlapping regions are mapped to equal posterior or high entropy regions in feature space. This results in a transformed feature space which magnifies the relevant sources of variability in input feature space and minimizes the unwanted sources of variability.

When the posterior for the correct class is not properly estimated (which may occur due to limited training data, other conditions for estimating posterior probabilities not being met, and/or test data that has characteristics dissimilar to the training data), the

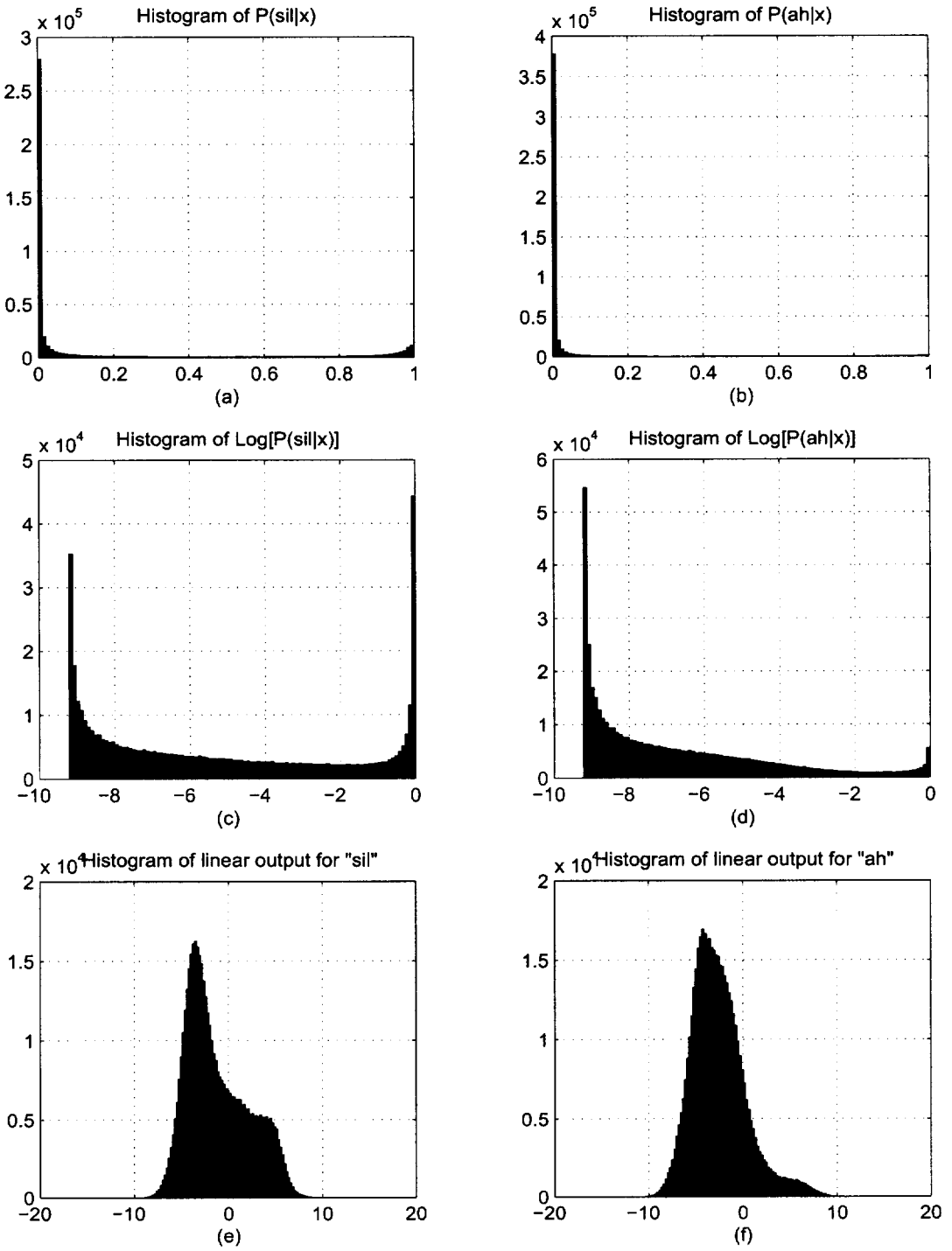

Figure 3.3: Histogram of MLP outputs:posterior probabilities, pre-nonlinearity outputs and log posterior probabities.

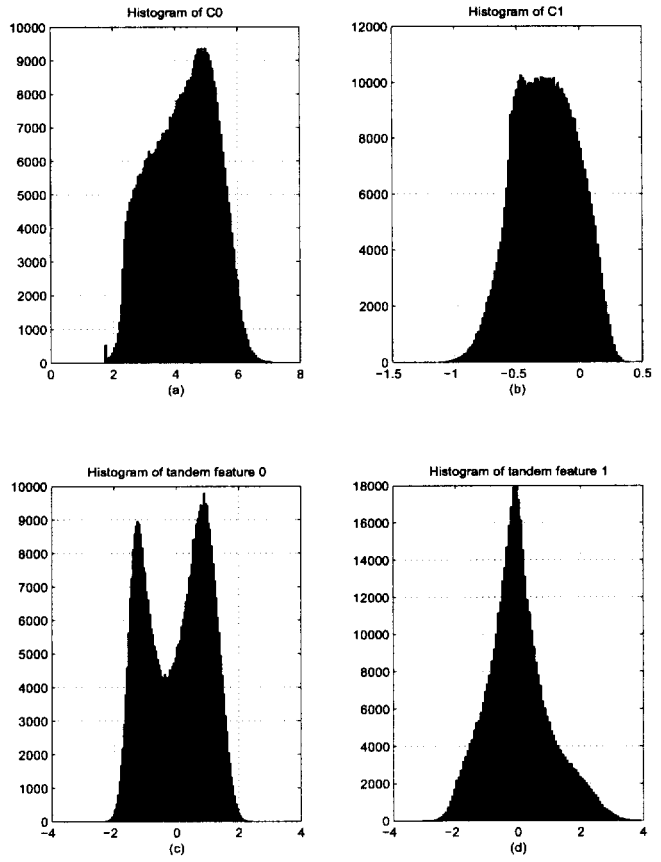

Figure 3.4: Histogram of first two cepstral coefficients (C0 and C1) and first two tandem features after PCA.

System	WER (%)	Relative Improvement(%)
Hybrid HMM-ANN	5.9	-
PLP HMM-GMM	5.1	13.6
Tandem posterior probabilities	5.7	3.4
Tandem log probabilities	5.2	11.9
Tandem linear outputs	4.9	16.9
Tandem log probabilities + KLT	4.6	22.0
Tandem linear outputs + KLT	4.4	25.4

Table 3.1: Word error rates on connected digit recognition task for hybrid HMM-ANN and HMM-GMM systems.

tandem approach shows improvement over the hybrid approach. If the MLP estimates posterior probabilities exactly, the tandem approach provides no advantage over hybrid systems; however, perfect probability estimation is almost never observed in practice.

3.2 Experiments

We use digits-only subset of the OGI Numbers database [66] for recognition experiments. It contains a vocabulary of ten American English digits in continuous utterances of lengths varying from one to seven words, labeled by twenty-three phonemes. The database is split into approximately 20000 digits for training and 12000 digits for testing. Phoneme label corresponding to the center frame is used as the target class. The MLP has 216 input units, 500 hidden units and 23 output units. HTK [77] is used to build HMMs. Context dependent phonemes are modeled with 3 states and 8 gaussian components per mixture HMMs.

Table 3.1 lists the WER and relative improvements. From the table, we see that the PLP HMM system, is 13.6% better than the hybrid HMM-ANN system. In the PLP HMM system acoustic models are context dependent phonemes compared to the context independent phonemes used in hybrid system. The best tandem system, using the pre-nonlinearity outputs of the MLP plus Karhunen-Loeve Transformation (KLT), reduces the baseline word error rate by 25.4%. The tandem features utilizes information over 195 milliseconds of speech, whereas the PLP HMM-GMM integrates only 105 milliseconds (9

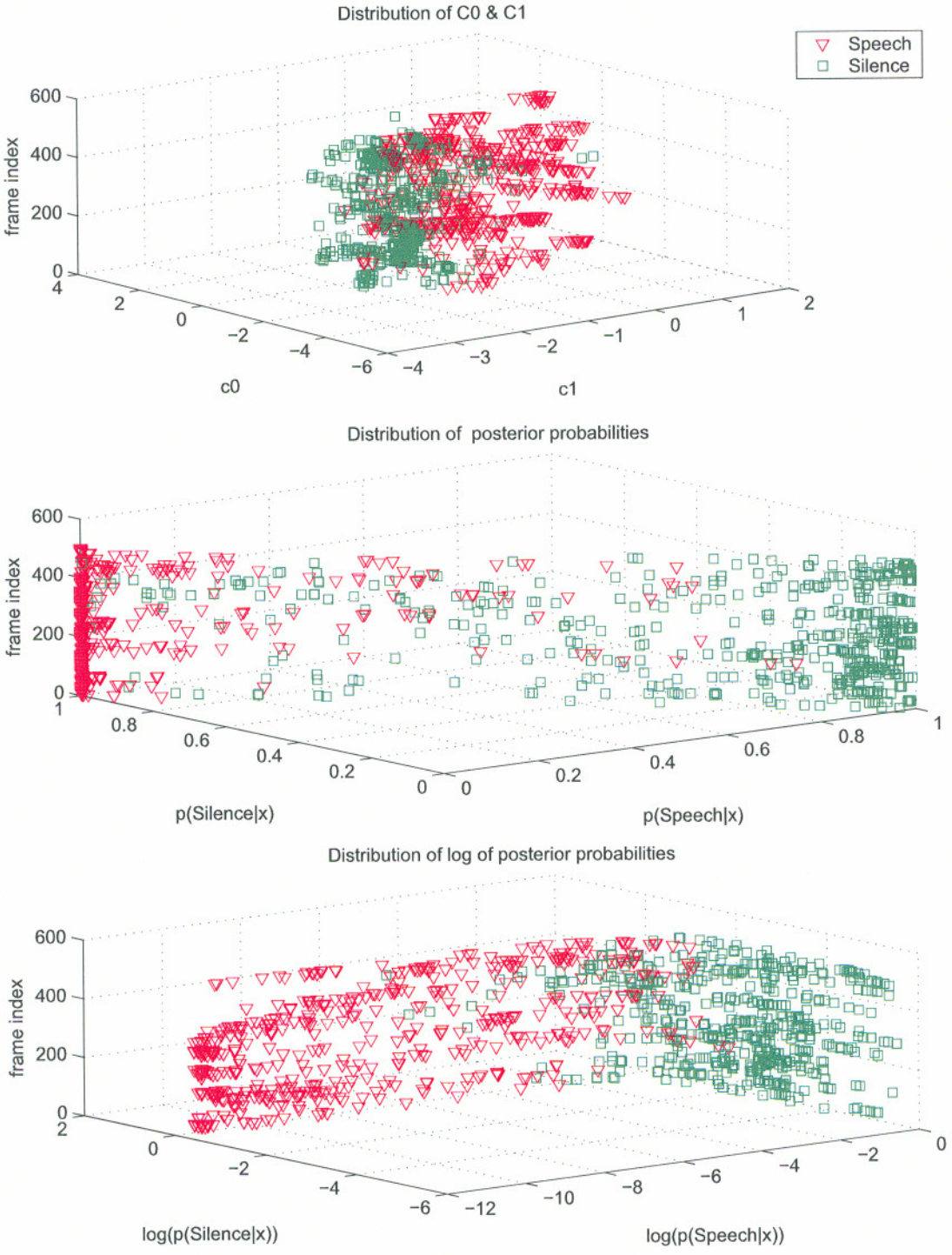

Figure 3.5: Scatter plot of first two cepstral coefficients (C0 and C1) and posterior probabilities for speech and non-speech classes.

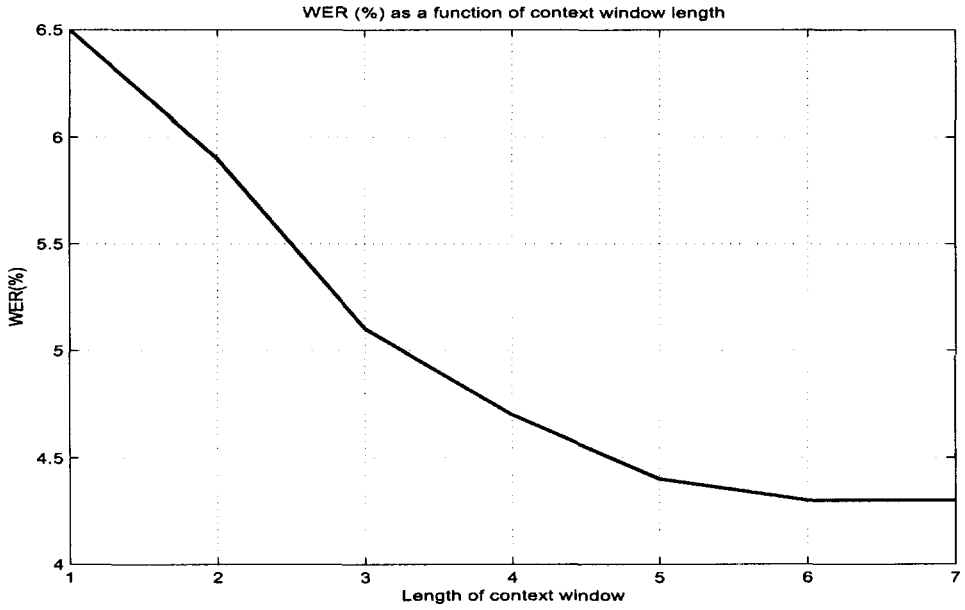

Figure 3.6: Word error rates vs. number of input frames to multi layer perceptron.

frames for computing delta and delta-delta features) of information.

3.2.1 Effect of number of parameters

We analyze the effect varying the length of input window size and number of hidden units in MLP on word error rate. While varying the number of input units the size of hidden layer is fixed at 500. Figure 3.6 illustrates the change in word error rate with respect to width of context window. It can be observed that the decrease in word error rate saturates at a window length of nine. This has been previously observed by Bourlard and Morgan [20]. The effect of varying the number of hidden units keeping the number of input units constant is shown in Figure 3.7. The improvement in word error rate diminishes with increasing number of hidden units. As evident from the figure, it saturates at 500.

3.3 Summary

In this chapter we explained a method of combining the discriminative classification power of neural networks with the modeling capability of hidden Markov models. It is named

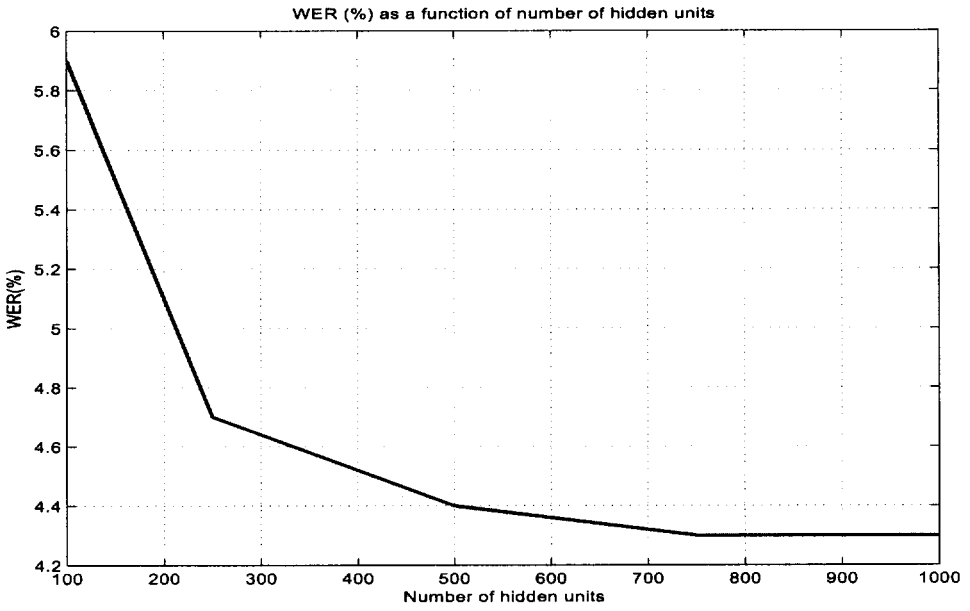

Figure 3.7: Word error rates vs. number of hidden units in multi layer perceptron.

tandem feature extraction due to having two classifiers in cascade. We showed, at least for this task, that tandem features give a notable reduction in word error rate compared to PLP cepstral features. The reasons behind the improvement could be many. They may be due to the fact that the features are functions of posterior probabilities or it may be due to the discriminative training of the features or both. We analyze the reasons behind the success of tandem features in the next chapter.

Chapter 4

Tandem Architecture

In the previous chapter we described a tandem feature extraction method using a multi layer perceptron (MLP)as a feature mapping tool.

This chapter further investigates the factors responsible for improvement seen with tandem modeling, whether they are due to the discriminative training of the MLP or because they are functions of posterior probabilities. The posterior probabilities can be estimated from data using a discriminative or generative model. A generative model maximizes the likelihood of the data given the model parameters. This is usually done assuming a parametric model for the distribution data. Most popular example of generative model is Gaussian Mixture Models (GMM), where the the distribution is assumed to be mixture of Gaussian components. A discriminative model, on the other hand, models the boundary between the classes in the feature space. A typical example of discriminative model is Multi-Layer Perceptron (MLP). In the next section we explain how the posteriors are estimated using GMMs and the next section examines various MLP architectures for estimating tandem features.

4.1 Gaussian Mixture Model (GMM) front end

In estimating the density function, $p(\mathbf{x}|C_i)$ we assume some parametric form of distribution. If we assume the distribution of the form

$$p(\mathbf{x}|C_i) = \sum_{j=1}^{M} p_{ij}\mathcal{N}_{ij}(x; \mu_{ij}, \Sigma_{ij}) \tag{4.1}$$

where M is the number of mixture components, p_{ij} are the component weights ($\sum_{j=1}^{M} p_{ij} = 1$) and $\mathcal{N}_{ij}(x; \mu_{ij}, \Sigma_{ij})$ is a multivariate Gaussian distribution with mean μ_{ij} and variance Σ_{ij}. The parameters of the model are estimated using the well known Expectation Maximization (EM) algorithm [2]. The EM algorithm maximizes the likelihood of the training data given the model parameters, $p(\mathbf{x}|C_i)$, in an iterative manner. Once we have the density function $p(\mathbf{x}|C_i)$, the posterior probabilities $p(C_i|\mathbf{x})$ can be estimated using the Bayes' rule given by

$$p(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i)p(C_i)}{p(\mathbf{x})} \tag{4.2}$$

where $p(C_i)$ is the prior probability of class C_i and $p(\mathbf{x})$ is the prior probability of feature vector $\mathbf{x}$. In the Tandem feature extraction framework, Ellis et. al [12] investigated the use of HMMs in estimating the posterior probability. Using Bayes' rule in Equation 4.2, they estimated posterior probabilities for each phone class $p(C_i|\mathbf{x})$ from the likelihoods $p(|C_i)$.

Thus, the output of a first-stage GMM distribution model, trained to model phoneme states, and convert it into posteriors that are an approximation to the neural net outputs. Here GMMs were trained using maximum likelihood approach. In maximum likelihood estimation, densities of each states probability distribution function are estimated independently. It does not address the class boundaries while modeling the distributions. We train GMMs in both maximum likelihood and in discriminative manner to test the hypothesis whether it is the use of posterior probabilities as features or estimating them discriminatively that gives the improvement.

GMMs can also be trained discriminatively. Modeling using GMM is a simple way of extending Gaussian model, or LDA, to nonlinear discriminant functions. If we assume a common covariance matrix for mixture components, the decision boundary is not linear. If we let $\Sigma_{ij} = \Sigma$ in equation 4.1 then in EM procedure we re-estimate only component weights and means only. This results in Discriminative GMM [82].

A logarithm is applied to the posterior probability estimates from GMM to increase the spread of their distribution. They are then decorrelated using PCA. The final feature vector is applied to a GMM/HMM recognizer.

Feature transformation	Frame error rate
GMM-Tandem	36.8%
Discriminative GMM-Tandem	37.5%
MLP-Tandem	34.9%

Table 4.1: Frame error rates on OGI stories for GMMs and MLPs.

4.1.1 GMM training

The English part of OGI Stories database [66] was used to train GMMs. It is approximately 3 hours of hand-labeled speech data. It is labeled by the ICSI56 phoneme set. GMMs are trained to model the feature space constructed using 8 PLP cepstral coefficients, 8 delta and 8 double delta features (=24 dimensions) after utterance based mean subtraction. There are 56 GMMs, each GMM models the conditional density of the phonemes in the training set. Table 4.1 gives the frame error rates for GMMs trained using the standard EM algorithm and the discriminant method. The feature space for MLP discriminant functions is constructed using nine frames, four frames from past and four frames in future, of PLP cepstral coefficients after utterance based mean subtraction. Phoneme label corresponding to the center frame is used as class. MLP has 216 input units, 500 hidden units and 56 output units. MLPs are trained using the backpropagation algorithm [10] using the learning rate reduction approach that was developed by [20]. The stopping criterion is based on the performance on a held out cross-validation data, which is 10% of the entire training data. In GMM, the best frame accuracy was obtained using 32 mixture components per phoneme. Increasing the components resulted in decrease of frame accuracy. In the case of Discriminative GMM the number of components per mixture was increased to 64. From Table 4.1 it can be seen that the "Discriminative GMM" gives better frame error rates then standard GMM inspite of having a much simpler model assumption. Next we perform connect digit recognition experiments using the tandem features derived using GMMs.

Type of feature	Number of features	WER (%)
PLP with Δ and Δ^2	24	5.1
GMM-Tandem	56	5.8
	24	6.1
Discriminative GMM-Tandem	56	5.3
	24	5.8
MLP-Tandem	56	4.7
	24	4.9

Table 4.2: Word error rates on connected digit recognition task for GMM tandem system. In the case of Tandem features the dimensinality of the features vector is the number of components retained after PCA.

4.1.2 Connected Digit Recognition Results

Digit recognition experiments are performed on OGI Numbers database. Subword units are modeled using 3 state left-to-right context dependent phoneme HMMs with 8 Gaussians/state and diagonal covariance matrix. The HMM recognizer is implemented using HTK [77]. Table 4.2 gives the baseline Word Error Rate (WER) for 8 PLP cepstral coefficients, 8 delta and 8 double delta features after utterance based mean subtraction. The PLP cepstral features are nonlinearly transformed using GMM and MLP estimates of posterior probability of context dependent phonemes. The distribution of posterior probabilities are smoothed using logarithm nonlinearity and decorrelated using PCA. The PCA transformation is derived on OGI Stories database. Table 4.2 tabulates the WER for baseline and tandem systems.

4.1.3 Discussion

From Table 4.2 we observe that the GMM tandem system does not give us any advantage over the baseline system. By training GMM discriminatively we make the systems comparable. This shows the discriminative training is the right approach. But GMMs lack the discriminative modeling power of MLPs. This is evident from the word error rates in the Table. The biggest difference between the Discriminative GMM front end and MLP front end is that MLP uses larger context, upto 105ms, compared to the short

Feature transformation	Dimensionality of tandem feature	WER (%)
GMM (9 frames PLP+d+dd - PCA216)	56	9.3
GMM (9 frames PLP+d+dd - PCA56)	56	5.9
GMM (9 frames PLP+d+dd - PCA24)	56	6.2
MLP (24 x 500 x 56)	56	6.5

Table 4.3: Word error rates for GMM and MLP based tandem system. "PCAxxx" denotes the number of dimensions retained after PCA on the 9 frames of cepstral features.

term context, upto 25ms, used in Discriminative GMM front end. Moreover MLP does not make any strict assumptions about the characteristics of the statistical distribution of the input features. To test the advantages of using a longer context to construct the input feature space for the nonlinear discriminant functions, we train GMMs on 9 frames context and MLP on single frame of input features. Since the Gaussians use diagonal covariance matrices, we decorrelate the concatenated feature stream using PCA before modeling using GMMs. Here we train only the "Discriminative GMM". Table 4.3 gives the WER for training the GMMs on 9 frames of input cepstral features. The last row gives the result with training MLP front end on single frame of cepstral features. It is evident from the table that the performance of the MLP front end became comparable to discriminative GMM when trained using one frame of cepstral features. The performance of the GMM front end weakened by adding more context. Using a large feature vector of 216 dimensions will result in data sparsity problem and when the dimension is reduced to a manageable number the dimensions do not carry enough variability to represent the information in the 216 dimensional feature vector.

4.2 Generalized MultiLayer Perceptron (MLP) front end

To investigate whether we need a MLP with two layers of nonlinearity to compute discriminant features, we study the use of a generalized Multi Layer Perceptron (MLP) architecture to Tandem feature extraction.In the Tandem feature extraction scheme a MLP with softmax output layer is discriminatively trained to estimate phoneme posterior

probabilities on a labeled database. The outputs of the MLP after nonlinear transformation and whitening are used as features in a Gaussian Mixture Model (GMM) based recognizer. Here we consider three layer MLPs with linear output layer. They nonlinearly transform the input data to a higher dimensional space defined by output of hidden units and performs Linear Discriminant Analysis (LDA) on the hidden unit outputs [8]. We compare the performances of these features with directly applying LDA on input data, which is equivalent to MLP with linear hidden and output layers.

The objective of discriminant feature extraction is to maximize the separability of classes modeled subsequently by Gaussian Mixture Models/hidden Markov models (GMM/HMM). The discriminative power of MLPs lies in their ability to perform nonlinear transformation of the input patterns into a higher-dimensional space spanned by the outputs of the hidden units in which class separation may be easier [8]. The MLP maximizes a discriminant function determined by the transfer function of the output layer and the particular target coding scheme. This can be seen as a nonlinear extension of Fisher's LDA.

The MLPs with linear output layer performs generalized LDA on the outputs of hidden units [8]. They nonlinearly transform the input data to a higher dimensional space defined by output of hidden units and performs a linear transformation that minimizes the mean-square error to a set of targets.Next section describes the link between MLP and discriminant analysis and the characteristics of *features* at various layers of MLP.

4.2.1 MLP and Discriminant Analysis

Linear Discriminant Analysis

LDA techniques were successfully applied to discriminative feature extraction. LDA finds directions of maximum linear separability between classes. The input vector is linearly transformed into the new feature vector. One can find many discriminant criteria in literature based on covariance matrices [39, 67]. One such criterion is

$$J = trace\left(\Sigma_W^{-1}\Sigma_B\right) \tag{4.3}$$

where Σ_W and Σ_B are the within-class and between-class covariance matrices respectively. The discriminant directions (A) are obtained from these matrices as

$$A = eig\left(\Sigma_W^{-1}\Sigma_B\right) \tag{4.4}$$

Discriminant features ($\tilde{\mathbf{x}}$) are obtained by projecting original features ($\mathbf{x}$) on discriminant directions (A), i.e., $\tilde{\mathbf{x}} = A^t\mathbf{x}$. The input vector $\mathbf{x}$ is typically a concatenation of a few adjacent frames of cepstral features and the classes are phonemes or subphone units [68]. The relationship between LDA and linear MLP is examined in [63].

Nonlinear Discriminant Analysis

The Nonlinear Discriminant Analysis (NDA) is based on a nonlinear transformation of the feature vectors into a new space with maximum discrimination between the classes. The *ideal* discriminant features are the posterior probabilities $P(C_i|\mathbf{x})$, where $C_i, 0 < i < L$ are the L classes [39]. They can be estimated using a three layer MLP [19]. Lowe and Webb have shown that, under the one-from-C target coding, a nonlinear MLP with linear output layer maximizes the discriminant function [8],

$$J = trace\left(\Sigma_{hT}^{-1}\Sigma_{hB}\right) \tag{4.5}$$

where Σ_{hT} and Σ_{hB} are the total and weighted between-class covariance matrices of the hidden unit activation vector $\mathbf{z}$. Note that in this case the outputs are not estimates of a-posteriori probabilities $P(C_i|\mathbf{x})$, although the outputs sum to one. In the case of MLP with softmax output nonlinearity, the outputs can be interpreted as probabilities of class membership conditioned on the outputs of the hidden units, $P(C_i|\mathbf{z})$, assuming that the class-conditional densities $P(\mathbf{z}|C_i)$ belong to *Normal* distributions that differ only in *mean* [5].

$$P(\mathbf{z}|C_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_{hT}|^{1/2}} exp\left(-\frac{1}{2}(\mathbf{z}-\mu_i)^t\Sigma_{hT}^{-1}(\mathbf{z}-\mu_i)\right) \tag{4.6}$$

$$P(C_i|\mathbf{z}) = \frac{exp(a_i)}{\sum_{j=0}^{L-1} exp(a_j)} \tag{4.7}$$

$$a_i = \mathbf{w}_i^t\mathbf{z} + w_{i0} \tag{4.8}$$

and

$$\mathbf{w}_i = \mu_i^t \Sigma_{hT} \tag{4.9}$$

$$w_{i0} = -\frac{1}{2}\mu_i^t \Sigma_{hT}^{-1} \mu_i + log(P_i) \tag{4.10}$$

where $\mathbf{w}_i$ and w_{i0} are the output layer weights and biases connected to the output node i, μ_i is the class conditional mean of the hidden unit activation vector $\mathbf{z}$. In the Tandem approach the features are the linear transformations of hidden unit activation vector $\mathbf{z}$. From Equations (6) and (7), tandem features can be viewed as linear discriminant functions of $\mathbf{z}$ [5]. Depending on the type of nonlinearity present in the output layer, the features derived from MLP with nonlinear hidden units are linear or nonlinear discriminant functions of the high dimensional *hidden space*. As a final step, we decorrelate the features using PCA and may reduce the dimensionality.

4.2.2 LDA and MLP Training

We used two databases to train the LDA and MLP. One is the English part of OGI Stories database [66]. It is approximately 3 hours of hand-labeled speech data. It is labeled by ICSI56 phoneme set. The other is the database on which recognition is performed, namely, OGI Numbers database. It contains ten continuous digits in utterances varying between one and seven digits, labeled by twenty-three phonemes. The database is split into approximately 20000 digits for training and 12000 digits for testing. The reason for choosing OGI Stories for *training* the *feature space* is to test the generalizability of the various discriminant functions. Since phonemes in spoken digits have limited context variability, the MLP if trained on it may learn the specific contexts and may not generalize to other tasks.

The feature space for discriminant functions is constructed using nine frames, four frames from past and four frames in future, of 8 PLP cepstral coefficients, 8 delta and 8 double delta features (24x9=216) after utterance based mean subtraction. Phoneme label corresponding to the center frame is used as class. For OGI Stories there are 56 classes and for OGI Numbers there are 23 classes. The linear transformation matrix A is obtained by maximizing the criterion in Equation (1). The MLPs, with softmax and linear output

Database	Linear	Softmax
Stories	42.4%	34.9%
Numbers	21.6%	16.6%

Table 4.4: Frame error rates on training databases for MLPs.

layer, are trained using the same target classes using one-from-N target coding scheme. Each MLP has 216 input units, 500 hidden units and 56(23) output units for training on OGI stories(Numbers). MLPs are trained using backpropagation algorithm [10]. The stopping criterion is based on the performance on a held out cross-validation data, which is 10% of the entire training data. The frame error rates for MLPs on the training databases are given in Table 4.4. It can be seen that the frame error rate for Stories database is approximately twice that of Numbers database. This may be due to the higher context variability in spontaneous speech and larger number of classes. The smaller error rate for softmax MLP may be due its ability to form more complex decision surfaces than MLP with just one layer of nonlinearity [5]. Figure 4.1 shows the output of MLPs, trained on Numbers database, with respect to time. It can be seen that estimates of posterior probabilities obtained using softmax nonlinearity are more distinct than the linear output case.

4.2.3 Feature Extraction

In the case of MLP features, PCA bases (derived from the MLP training data) are used to whiten the feature space and optionally to reduce the number of features for subsequent modeling by HMM. For the *linear* and $softmax$ feature space, there are $L-1$ linearly independent features, where L is the number of classes. This is because the outputs sum to one. As explained in the previous Chapter, for computing the Tandem features, the softmax activation function is replaced with linear activation function. Here again, PCA bases are applied similarly.

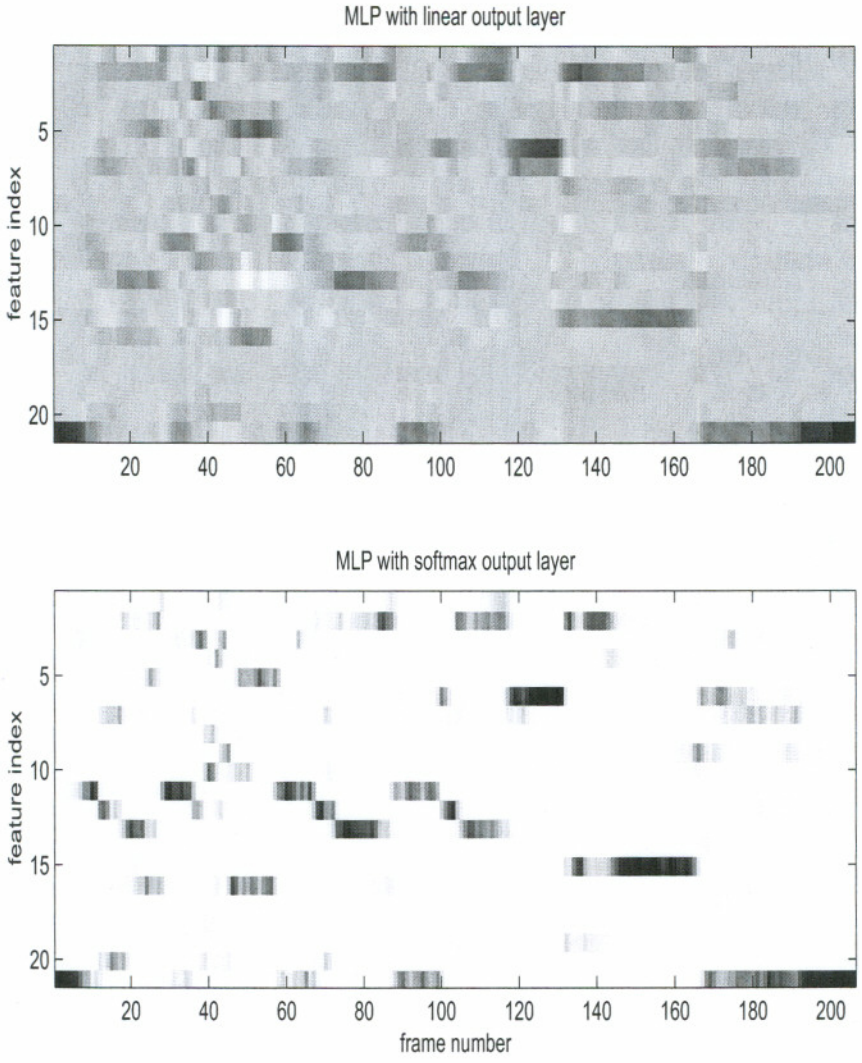

Figure 4.1: Outputs of MLPs with linear and softmax output layer.

Feature	Dimension	WER (%)
PLP Cep+Δ+$\Delta\Delta$	24	5.1

Table 4.5: Baseline Word error rates on connected digit recognition task.

MLP training	Feature	Dimension	WER (%)
Stories	LDA	24	6.0
	MLP-Linear	55	5.9
	MLP-Linear	24	6.3
	Tandem	56	**4.7**
	Tandem	24	4.9
Numbers	LDA	22	5.7
	MLP-Linear	22	5.6
	Tandem	23	**4.4**

Table 4.6: Word error rates on connected digit recognition task. "MLP-Linear" for outputs of MLP trained with linear output layer.

Feature	Dimension	WER (%)
LDA on hidden layer	22	5.7

Table 4.7: Word error rate for LDA on hidden unit outputs.

4.2.4 Connected Digit Recognition Results

Digit recognition experiments are performed on OGI Numbers database. Subword units are modeled using 3 state left-to-right context dependent phoneme HMMs with 8 Gaussians/state and diagonal covariance matrix. The HMM recognizer is implemented using hidden Markov tollkit (HTK) [77]. Table 4.5 gives the baseline Word Error Rate (WER) for 8 PLP cepstral coefficients, 8 delta and 8 double delta features after utterance based mean subtraction. To keep the dimension of the feature vector same as the baseline, PCA is used to reduce the dimensionality of the MLP outputs from 56 to 24 when the MLP is trained on OGI Stories. Table 4.6 gives the WER for discriminative features, when they are trained on Stories and Numbers databases. It can be seen that only the Tandem features perform consistently better than the baseline irrespective of the training database.

4.2.5 Discussion

Both LDA and MLP with linear output layer fail to provide any improvement over the baseline system. Both of them are derived by maximizing discriminant criteria based on covariance matrices. In the case of LDA the criterion is maximized on the input

feature space, whereas it is maximized on the hidden unit activation space in the case of MLP. Although the linear output MLP gives better performance than LDA, the expected advantage by the nonlinear transformation of input features to a higher-dimensional space is not visible here.

To check the performance of using another discriminant criterion, other than the one in Equation(3), we compute the linear discriminants on the hidden units space separately after the MLP has converged. We use the same discriminant function as in Equation (1). From Table 4.7 it can be seen that the WER obtained is similar to the MLP discriminant function.

Replacing the softmax activation function of each output node by a linear function [31] at the output of MLP increases the spread of the distribution of the features. This results in better modeling by HMMs without any loss in discrimination compared to posterior features. T

Figure 4.2 shows the normalized eigenvalues of the PCA matrix of MLP outputs and the LDA transformation matrix. The rate of roll-off of normalized eigenvalues of LDA is steeper than that for the MLP outputs. This shows that all the dimensions carry significant amount of class information in the case of MLP outputs where as they are highly compressed in the case of LDA.

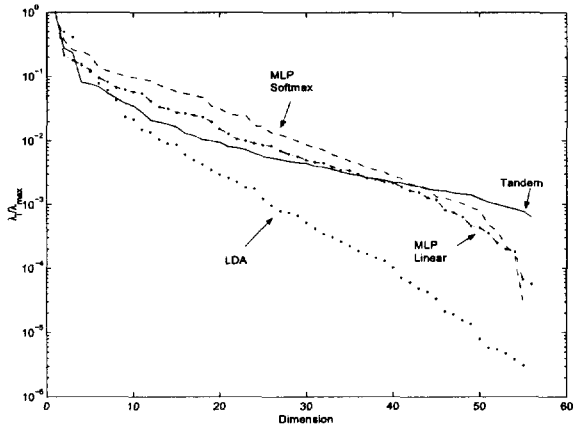

Figure 4.2: Normalized eigenvalues of LDA and MLPs trained on OGI Stories.

4.3 Hierarchical discriminative front end

Modular and hierarchical neural networks have been studied extensively in pattern recognition literature [52, 53]. These networks divide the overall classification task among several networks. The decisions from networks are combined in a hierarchical manner to arrive at the overall network output. Thus the task of classifying a global set of classes, context-independent phonemes in the case of tandem approach, is divided into subsets. The partition is based on prior knowledge about the task or by data-driven clustering algorithms. For example, natural choice of first partitioning in the case of phonemes will be speech and silence.

The hierarchical systems can have shorter training times and can have fewer parameters than the monolithic neural networks. This technique has been applied to build connectionist acoustic models [34]. Here we investigate the effectiveness of hierarchical approach in feature extraction under tandem framework. This is implemented as hierarchies of MLPs. We make soft splits of data using soft classification trees. This is based on the statistical method of factoring posteriors [36, 59] which is explained in the next section.

4.3.1 Hierarchical Classification

Factoring Posterior Probabilities

Let L denote the set of classes C_k to be discriminated. Consider the partition of L into M disjoint and non-empty subsets L_i such that members of L_i are least confused with members of L_j ($\forall j \neq i$). A particular class C_k will now be a member of L and only one of the subsets L_i. Therefore, we can rewrite the posterior probability of class C_k as a joint probability of the class and the corresponding subset L_i and factor it according to

$$\begin{aligned} p(C_k|\mathbf{x}) &= p(C_k, L_i|\mathbf{x}), C_k \in L_i \\ &= p(L_i|\mathbf{x})p(C_k|L_i, \mathbf{x}). \end{aligned}$$

Thus, the global task of discriminating between all the classes in L has been converted into discriminating subsets L_i and independently discriminating the classes C_k remaining

within each of the subsets L_i. Recursively repeating this process yields a hierarchical tree-organized structure. The posterior probability for a specific class can be computed by multiplying all the conditional posteriors from root node to the leaf corresponding to the specific class.

Conditional node posteriors can be estimated by restricting the training set of the corresponding MLP to the subset L_i on which the probability is conditioned. Thus the training data for each node is shared among all its child nodes according to the partitioning of classes and the amount of training data decreases with increase in specialization. Due to the diminishing training data as we traverse down the tree and the errors in posterior estimation, the the design of hierarchical structure become crucial.

4.3.2 Hierarchical Tandem System

Hierarchical Tree Structure

If all the nodes in the tree would compute true conditional posteriors, the tree structure would have no influence on the classifier performance because any kind of factoring yields an exact decomposition of the class posteriors.Since this not true in practice, the choice of tree structure is important. Due to the large number of choices at each node it is impossible to find an optimal structure through an exhaustive search. Hence we apply evidence from data and heuristics to design the tree structure.

In speech recognition the obvious first partitioning is speech and silence. At the root of the tree we discriminate speech and background noise. This is motivated by the observation that these classes are easy to distinguish acoustically. The speech subset is further split into voiced and unvoiced classes. The leaf nodes of the tree compute context independent phoneme (monophone) posteriors conditioned on voiced and unvoiced classes. Figure 4.3 shows the topology of the hierarchy. In this paper we design a hierarchical tree with three levels. Table 4.8 shows the hierarchical splitting of classes. "Tandem 0" system is the basic tandem system with single MLP. "Tandem 1" has two levels of hierarchy and "Tandem 2" has three levels of hierarchy.

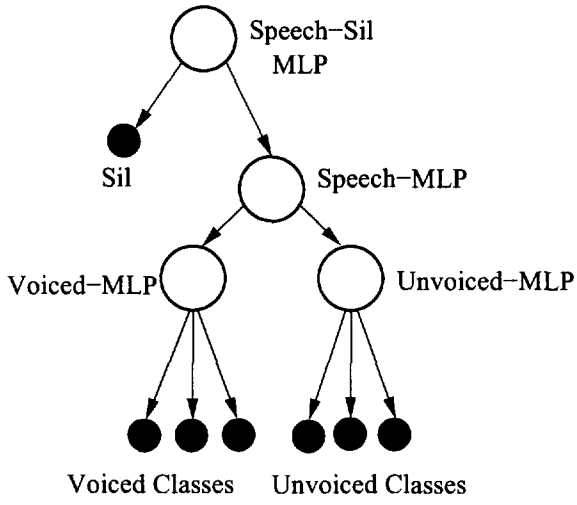

Figure 4.3: Hierarchies of MLPs

Classifier	Hierarchy	Classes
Tandem 0	0	monophones + sil
Tandem 1	0	speech - sil
	1	monophones
Tandem 2	0	speech - sil
	1	voiced - unvoiced
	2	voiced classes
	2	unvoiced classes

Table 4.8: Hierarchical splitting of classes.

Postprocessing of Posteriors

The posterior probabilities have a skewed distribution, making them harder to be modeled by mixture of Gaussian components. Different postprocessing methods to warp the posteriors into a different domain has been tried [31]. Replacing the softmax nonlinearity at the output layer with a linear function is shown to make the distribution more Gaussian. Removing the softmax is equivalent to the logarithm of the posteriors with a normalization constant.

$$l_i = log(p(C_i|\mathbf{x})) - K, 0 \leq i \leq N-1$$

where l_i is the linear output corresponding to class C_i and $K = log(\sum_{i=0}^{N-1} exp(l_i))$. Since we have no means of obtaining K from $p(C_i|\mathbf{x})$ we approximate it by the average of the log posteriors.

$$K = \frac{1}{N} \sum_{i=0}^{N-1} log(p(C_i|\mathbf{x})), 0 \leq i \leq N-1$$

The distribution of resulting features is found to be similar to the one obtained by removing the softmax. As explained in the previous Chapter, diagonalization of the global covariance matrix of the features by PCA transformation improves the performance because the GMM/HMM assumes that features are uncorrelated. We retain all the feature components after PCA.

4.3.3 Experimental Evaluation

We tested the hierarchical system on SPINE1 task [61]. We need a bigger database than OGI stories and OGI Numbers for hierarchical splitting, because as we traverse down the tree, the amount of data available to train each node decreases. The task focuses on transcribing speech produced in noisy environments with emphasis on noisy military environments. It involves a medium-sized vocabulary of about 5000 words. The data consists of conversations between two communicators working on a collaborative, Battleship-like task in which they seek and shoot at targets.Each person is seated in a sound chamber in which a previously recorded military background noise environment is accurately reproduced. The speech is sampled at 16KHz.

Perceptual Linear Prediction (PLP) cepstral features are extracted from a frame of 25 ms of speech, every 10ms. The feature vector consists of 13 PLP coefficients augmented by deltas and double-deltas. They are then normalized over the utterance to zero mean and unit variance. The input to each MLP is a window of 9 successive feature vectors.

The labels for training MLP are generated by the process of forced alignment as explained in [12]. From ICSI56 context-independent phoneme set a subset of 50 phonemes occurring in SPINE1 data was derived. Each MLP in the hierarchy is trained by back-propagation with a minimum-cross-entropy criterion to 'one-up' targets obtained from the

Classifier	Hierarchy	No. of Classes	IU	HU	OU
Tandem 0	0	50	351	1000	50
Tandem 1	0	2	351	500	2
	1	49	351	750	49
Tandem 2	0	2	351	500	2
	1	2	351	500	2
	2	37	351	500	37
	2	12	351	500	12

Table 4.9: Architecture of MLPs in the hierarchy. IU stands for number of input units, HU for hidden units and OU for output units.

labels. The outputs from the MLPs are fed to the GMM/HMM system after the post-processing. The GMM/HMM system is trained according to the standard EM algorithm. We used CMU SPHINX-III recognizer [85] with 3 states per context-dependent phoneme with 2600 tied states, each modeled by a mixture of 8 Gaussians.The context-independent phonemes are also modeled using 3 state HMMs with 8 Gaussians per state. The tandem MLP and GMM are trained independently and use different number of context-independent phonemes. Table 4.3.3 shows the architecture of each MLP in the hierarchy.

Results

The SPHINX system was trained on 8 hours of data. Models were trained for three tandem features and the PLP features. Recognition was performed on 9 hours of evaluation data. The word error rates for all the systems are shown in Table 4.10. The recognizer was not tuned to improve the performance of individual systems.

When Context Independent models are used for decoding, the tandem systems outperform the PLP system. The word error rates for Context Independent models are 71.6% for PLP system and 50.5% for Tandem respectively. The performance of the systems tend to converge when context-dependent models are used for recognition. The tandem systems are marginally better than the PLP system. However, Tandem 2 system performs 3% better than the Tandem 0 system and 5% better than the PLP system.

Type of feature	Dimensions	WER (%)
PLP with Δ and Δ^2	39	39.1
Tandem 0	50	38.3
Tandem 1	50	38.2
Tandem 2	50	37.1

Table 4.10: Word error rates (%) with SPHINX-III system for various feature sets.

4.3.4 Discussion

We find that the tandem systems perform notably better than the PLP system with context-independent models whereas only marginal improvement is obtained with context-dependent models. It is observed that the hierarchical tandem system perform marginally better than the monolithic classifier based system with context-dependent models and worse when context-independent models are used.

In [12] we interpreted the MLP in tandem modeling as a transformation of the feature space that magnifies regions around phonetic boundaries and suppressing the non-phonetic variability due to speaker and noise within the region corresponding to class. In the tandem approach we train the MLP to maximize the separability of context-independent phonemes with a block of 9 successive frames of feature vectors as input. The target phoneme corresponds to the frame at the center of the window. This introduces shift-invariance and suppresses the context, speaker and environmental variability. Thus there is little information left to be modeled by context-dependent GMMs. This explains why the improvements without the context-dependence in the HMM from tandem are much more dramatic.

It can be observed from Table 4.3.3 that the number of parameters in Tandem 2 system is half of that in Tandem 0. This has reduced the training time and system complexity without affecting the performance.

To investigate further the reasons for the disparity in improvements from context-independent and context-dependent models we tested the features with GMMs of varying complexity. Figure 4.4 shows the performance curve of PLP system and Tandem 2 system for different number of Gaussians/state keeping the number of states per model unit same

(=3). Increasing the number of Gaussians gives GMM additional parameters to model the variability in feature space within each phoneme. It can be seen from the figure that the performance of the tandem system and PLP system tend to converge with increasing number of Gaussians/state. The word error rate of PLP system reduced by 25% from 1 Gauss/state to 8 Gauss/state whereas the tandem system improved by only 10%. Thus with increasing modeling complexity the GMM is able to minimise the advantages of discriminant transformation.

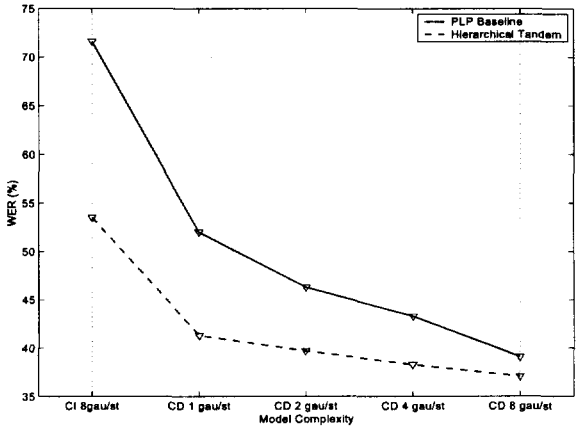

Figure 4.4: Word error rates (%) of feature sets vs. Complexity of models

4.4 Summary

We investigated the roots for the improvement in word error rate using tandem features. We showed that discriminative training of the feature extractor and the use of large context are the critical factors that contributes to the improvement. We established this comparing a GMM based front end with MLP based front end. The GMM based front end fails to model the likelihood density function well when a large context is used. This may be due to the non Gaussian nature of the distribution of features. This is one of the advantages in using MLP as a feature transformation. MLP does not make any assumptions about the nature of the distribution. We showed that discriminative training of the feature transformation is better than maximum likelihood training. Discriminative training emphasizes the class boundaries, whereas the maximum likelihood training is unaware

of the class boundaries.

Using a generalized MLP architecture we compared various linear and nonlinear discriminant functions for feature extraction in this paper. An MLP trained with softmax nonlinearity at the output layer to estimate posterior probabilities outperforms other discriminant functions. This may be due to multiple reasons. First when the MLP is trained with nonlinearity in the output layer to estimate class conditional posterior probabilities, it can learn more complex decision regions than the the MLP with linear output layer. By removing the nonlinearity during the forward pass the features are more suitable for Gaussian mixture modeling without loss of discriminating information in the posterior probabilities.

Next we presented a study on hierarchical feature extraction under tandem framework using a MLP tree. Hierarchical modeling offers a power method of combining multiple classifiers into a tree structure. We have shown that it achieves better word error rates to a monolithic MLP, with dramatically fewer parameters. The design of the tree was based on the prior knowledge of classes. The tandem modeling approach offers considerable advantages for low complexity systems with few subword classes especially when signal to noise ratio is low.

Chapter 5

Target class selection

Tandem features are trained on Context Independent (CI) targets. Subsequent stochastic classifier typically uses subword units that are different than CI phonemes. A common example is Context Dependent (CD) phoneme models that are used in state of the art ASR systems. Maximizing the discriminability between a set of target classes in feature extraction other than the classes modeled by the subsequent stochastic classifier may not be the optimum choice. The focus of this chapter is to investigate the effect of using different set of target classes in feature extraction and in acoustic model. In [12] the MLP is trained to discriminate between Hidden Markov Model (HMM) states of whole word HMMs. They observed a slight improvement in performance compared to phoneme targets. In a similar approach [4] neural networks are trained to map short-term spectral features to the posterior probability of distinctive features. They used 60 distinctive features comprising articulatory features [42] , plus some broad phonetic classes as targets. Using 44 CI phoneme targets, on a large vocabulary task they obtained better performance than the distinctive features .

5.1 Context Independent Targets

Are CI phonemes optimal targets for tandem feature extraction? In this section we try to find smaller set of target classes than CI phoneme classes. This results in smaller set of features and fewer parameters in the GMM classifier. The dimensionality of the feature vector can be reduced using PCA without changing the number of classes. Another method is to cluster the phoneme classes. Clustering using prior knowledge such as

voiced/unvoiced and vowel/consonant results in broad category targets. We do not know whether this is an optimal way of clustering. Another approach is data-driven clustering.

Since we use a nonlinear and complex HMM classifier with multiple states and Gaussians, we may not require the MLP to discriminate among all the classes. We start with ICSI56 phoneme set. We try four clustering methods to reduce the number of classes, 1) using apriori knowledge to cluster the phonemes to broad phonetic categories 2) decision tree based clustering [78] 3) data driven clustering of phoneme models [79] and 4) Mutual Information (MI) based clustering to reduce the phoneme confusions.

5.1.1 Broad phonetic categories

We cluster CI phonemes based on their phonetic properties to obtain seventeen broad phonetic categories. The categories are front vowels, central vowels, back vowels, retroflexes, diphthongs, voiced plosives, unvoiced plosives, nasals, flaps, voiced fricatives, unvoiced fricatives, affricates, glides, voiced closures, unvoiced closures, syllabics and silence. Table 5.1 shows the grouping of phonemes into broad phonetic categories. The mapping from phonemes to the broad-phonetic-class is obtained from a International Phonetic Alphabet (IPA) chart. By training a MLP on these targets we are extracting phonetic "features".

5.1.2 Data derived classes

Extracting features based on hardwired phonetic attributes may not be optimal for classification of phonemes. There are many approaches in literature to derive classes using clustering techniques.

Decision tree based clustering

Here phoneme models are clustered using a phonetic decision tree [78]. A phonetic decision tree is a binary tree with yes/no questions attached to each node. We use HTK [77] to build the decision tree. Initially the 56 CI phoneme are modeled using single state, single gaussian models. Each phoneme is renamed to have the same central phoneme. For example phoneme "aa" is renamed as "aa-phn+aa" and phoneme "p" as "p-phn+p", so that all phonemes are placed in a single cluster at the root of the tree corresponding to

Broad phonetic category	Phonemes
front vowels	iy, ih, eh, ae
central vowels	ix, ux, ax
back vowels	uw, uh, ah, ao, aa
retroflexes	er, axr
diphthongs	ey, ay, oy, aw, ow
unvoiced plosives	p, t, k
voiced plosives	b, d, g
nasals	m, n, nx, ng
flaps	dx
unvoiced fricatives	f, s, th, sh, hh
voiced fricatives	v, dh, z, zh
affricates	ch, jh
glides	l, y, r, w
voiced closures	bcl, gcl, dcl
unvoiced closures	pcl, kcl, tcl
syllabics	em, en, el
silence	h#, q

Table 5.1: Grouping of phonemes into broad categories based on International Phonetic Alphabet (IPA) chart.

the central phoneme "phn". The decision tree asks whether the phoneme to the left/right of the central phoneme is in a certain set, e.g. "Is the phoneme to the left or right a plosive?". The questions sets are borrowed from HTK's demonstration system for large vocabulary speech recognition. Examples are: "Vowel", "Fricative" , "Stop", etc. The question that gives the maximum increase in log likelihood is chosen at each node. This process is repeated until the increase in log likelihood falls below a specified threshold. We vary the number of classes from two to fifty five by changing the threshold. Figure 5.1 shows an example of splitting classes using decision tree.

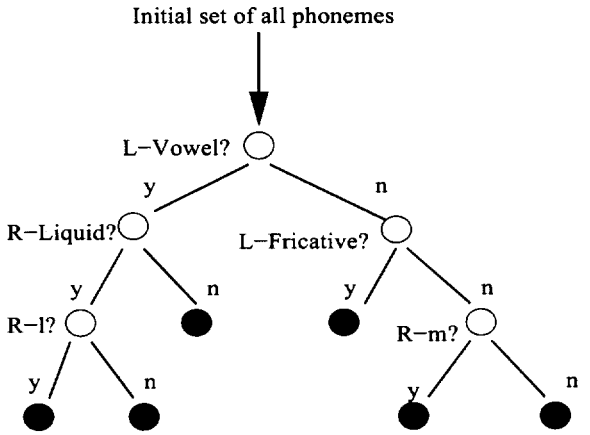

Figure 5.1: Example of a phonetic decision tree.

Broad phonetic categories are a special case of classes obtained using decision tree clustering.

Data driven clustering of phoneme models

Initially all the phoneme models are placed in individual clusters. Here again the phonemes are modeled using single state, single gaussian HMM. The pair of clusters which, when combined, would form the smallest resultant cluster are merged. This process repeats until the number of clusters have reached the specified number. The size of the cluster is defined as the greatest distance between any two phoneme models. Euclidean distance between the class conditional means weighted by the inverse of the variance is used as the distance metric. We use HTK to implement the clustering.

Mutual information based clustering

First a hybrid HMM/MLP [20] is trained to estimate the phoneme posterior probability using the manually labeled training data. Using the frame level phoneme classification results on the training data a confusion matrix is obtained. A confusion matrix (CM) is a matrix of hits and misses for all phonemes. A joint Probability Distribution Function (PDF) is estimated from the confusion matrix by dividing each element in it with the total number of phoneme segments. We compute the Mutual Information (MI), $I(X;C)$ between the feature vector X and phoneme C from the joint PDF. $I(X;C)$ is the reduction in uncertainty of the phoneme C due to the knowledge of X [81]. The pair of phonemes which, when combined, would result in the maximum reduction in $I(X;C)$ are merged to form new classes. For example, "em" and "en" are merged in the first step to form a new class "em_en". The process is repeated until all the phonemes are paired.

5.1.3 MLP training

We used two databases in our experiments. One is the English part of OGI Stories database [66]. It is approximately 3 hours of hand-labeled speech data. It is labeled by ICSI56 phoneme set. This is used to train the MLP and in deriving new categories from data. The other is the database on which recognition is performed, namely, OGI Numbers database. The database is split into approximately 20000 digits for training and 12000 digits for testing. Note that all the clustering schemes are performed on a database independent of the final recognition task.

The baseline tandem system is trained on ICSI56 CI phonemes. Input to the MLP is nine frames, four frames from past and four frames in future, of 8 PLP cepstral coefficients, 8 delta and 8 double delta features (24x9=216) after utterance based mean subtraction. The phoneme label corresponding to the center frame is used as the target class. MLP has 216 input units, 500 hidden units and 56 output units.

The single state, single gaussian HMMs used in clustering are trained on 8 PLP cepstral coefficients, 8 delta and 8 double delta features. After clustering the classes using the aforementioned techniques, the CI phonemes are mapped to the new classes and an MLP

Feature	CI models	CD models
PLP Cep+Δ+ΔΔ	6.2(%)	5.1(%)
Baseline Tandem	5.7(%)	4.7(%)

Table 5.2: Word Error Rates (WER) on connected digit recognition task using CI phoneme targets.

is trained on each of the new classes. Each MLP has same number of input and hidden units as the baseline system, only the number of output units vary from two to fifty-five.

5.1.4 Results

Connected digit recognition experiments are performed on OGI Numbers database. The 23 context independent phonemes are modeled using 3 state left-to-right HMMs with 8 Gaussians/state and diagonal covariance matrix. Table 5.2 gives the Word Error Rate (WER) for 8 PLP cepstral coefficients, 8 delta and 8 double delta features after utterance based mean subtraction and the baseline tandem system. Context dependent (CD) HMM system used 3 states per HMM with 1300 tied states, each modeled by a mixture 8 Gaussians.

Figure 5.2 shows the WER, using CI models, for various clustering schemes and different number of classes. It can be seen that WER rolls off much faster with increasing number of clusters using MI based clustering than decision tree based clustering and data driven clustering of phoneme models. The WER for MI based clustering saturates at 29 classes, the WER=6.0% (not statistically significant compared to 5.7% at 95% confidence). The best WER of 5.5% is obtained using 34 categories. The WER for tree-based and data-driven clustering of phoneme models continues to improve with increase in number of classes. The better performance of MI based clustering could be attributed to the complexity of the phoneme models used in clustering. Both tree based and data driven clustering use single state, single gaussian model (because of the limitations of the HTK software), whereas MI based clustering use a MLP to generate the confusion matrix. To verify the effect of complexity of the models used in clustering on WER we trained two sets of HMMs to generate phoneme confusion matrices. A single state, single gaussian model

Phoneme model	Number of classes			
	12	24	36	48
HMM-1 state	10.1	8.7	7.6	6.3
HMM-3 states	7.6	6.8	6.2	5.9
MLP	7.1	6.4	6.0	5.7

Table 5.3: Results of MI based clustering.

and three state, eight gaussian components per state model on OGI stories. Phoneme recognition is performed on training data and phoneme confusion matrix is obtained as explained in section 5.1.2. Table 5.3 compares the WER obtained for 12, 24, 36 and 48 categories obtained using HMM and MLP. It can be seen that the more the complexity of the model used in estimating MI the better the WER.

Table 5.4 compares the performance of seventeen broad categories with the same number of classes obtained using MI based clustering. It shows that "better" target categories can be obtained using data driven methods than grouping based on phonetic properties.

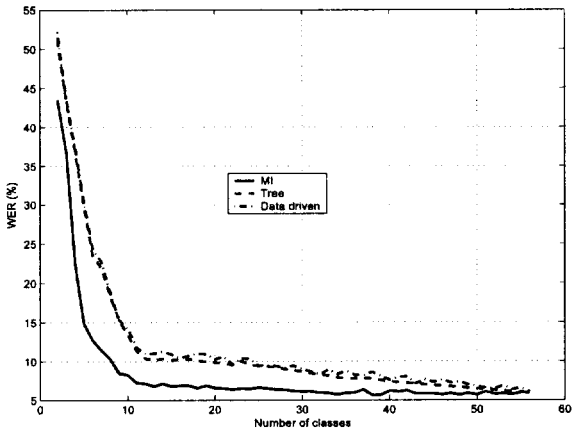

Figure 5.2: Word Error Rates (WER) for different clustering schemes and number of classes.

Clustering method	WER (%)
Broad categories	8.1
MI based	6.9

Table 5.4: Word Error Rates (WER) on connected digit recognition task for seventeen categories obtained by MI based clustering.

Clustering	WER (%)
Random-1	10.9
Random-2	12.4
Random-3	12.5

Table 5.5: Results using random clustering of phonemes.

Random clustering

To test whether it is the "meaningful" clustering approach that is providing the improvement, we cluster the phonemes randomly. A random number generator produces the indices of phonemes to be grouped. We generated three different random mappings to reduce the number of classes to seventeen. As shown in Table 5.5 the WER increases.

PCA versus clustering

The dimensionality of feature vector can be reduced using PCA. We reduce the dimensionality of the baseline tandem features from 56 and compare the results with same number of classes obtained using MI based clustering. From Table 5.6 it can be seen that the performance of clustering is better than PCA, but not statistically significant. This suggests that clustering retains marginally more information for discriminating among phonemes than PCA.

	Number of features			
	12	24	36	48
PCA	7.2	6.8	6.4	5.9
MI	7.1	6.4	6.0	5.7

Table 5.6: PCA vs. MI based clustering.

Target classes	MLP architecture (Input-Hidden-Output)	Training	Cross-validation
CI phoneme	216-500-56	65.3(%)	57.1(%)
HMM state	216-500-168	41.4(%)	36.1(%)
	216-750-168	45.2(%)	39.8(%)
	216-1000-168	45.5(%)	40.2(%)

Table 5.7: Frame accuracy for HMM state targets on training and cross validation data.

5.1.5 HMM state targets

When each CI phoneme is modeled using 3 state HMMs, each state models the temporal variability in phoneme due to context and speaking style. Here we can assume that the center state captures the steady part of the sound that is least affected by the context and the adjacent states model the effect of left and right context. There are 168(=56x3) state targets, resulting in a 168 dimensional feature vector. The dimensionality is reduced using PCA. To obtain the state targets, we force aligned OGI stories using the CI phoneme HMMs trained on baseline tandem features. MLP is trained on these 168 targets in the same fashion as with CI phoneme targets. We increased the hidden layer size from 500 to 750 to improve the discrimination among acoustically similar targets. Table 5.7 shows the frame accuracy on training and cross-validation set for the MLP for various hidden layer sizes.

As expected the frame error rate for 3-state targets is higher than the single state targets. Most of the confusions may be the result of same phoneme getting confused among different states within the phoneme.

5.2 Context dependent targets

In the previous chapter we saw that the difference in performance between the PLP cepstral and tandem features diminished with increase in complexity of the HMMs. The MLP maximizes separability between CI phonemes whereas HMMs model context dependent subword units. The reduction in confusion among CI phonemes may be coming at the expense of increasing the confusion among context dependent phonemes. To test whether

Number of target classes	Training	Cross validation
75	56.9	51.8
100	53.6	48.5
125	50.7	44.2
168	47.2	41.1

Table 5.8: Frame accuracy for decision tree derived targets on training and cross validation data.

this is due to the mismatch in target classes used in feature extraction and acoustic modeling we use context dependent targets in MLP. Training an MLP with large number of output units that are highly overlapping is a challenging task. It will result in a high dimensional feature vector. We address these problems in the following subsections.

5.2.1 Decision tree clustering of context dependent phonemes

In HMM training decision tree clustering of context dependent HMMs is used to handle unseen contexts in training data and to reduce the number of free parameters [78]. In Section 5.1.2, we applied decision tree clustering of CI phonemes to reduce the number of target classes. Here we apply the same technique to cluster some of the context dependent phonemes to arrive at a manageable number of targets. From the single state single gaussian CI phoneme models in Section 5.1.2 we created "cloned" context dependent phoneme models. These models are then clustered using the same decision tree approach used in Section 5.1.2. By varying the threshold in the clustering the number of classes is controlled. We reduce the number of classes to 168, same number of classes as HMM state targets. We further reduce the number to 125, 100 and 75 to find an optimum number of classes that can be obtained using decision tree clustering. Table 5.8 shows the frame accuracy for various number of target classes. We kept the number of hidden units at 750. As seen from the Table 5.8, the frame error rate increases with increase in number of decision tree clustered context dependent targets.

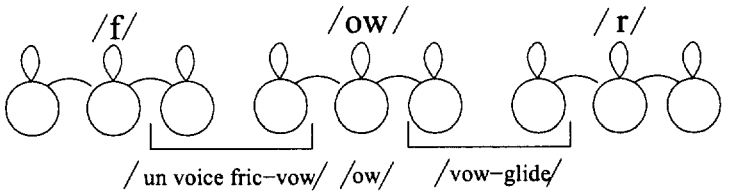

Figure 5.3: Generalized context dependent transition targets

5.2.2 Generalized context dependent transition (diphone) targets

In [70] Dupont et al. used MLP to estimate posterior probabilities of context independent phoneme targets combined with generalized context dependent transition (diphone) targets. As shown in Figure 5.3 the center part of the CI phoneme segment is retained as it is the least context sensitive part of the phoneme. To make the number of targets manageable the transitions are clustered to generalised transitions. We cluster the HMM state labels obtained in Section 5.1.5 to diphone transitions based on left and right context.

The transition clusters are based on the broad phonetic classes of the left and right phoneme. Instead of labeling transitions defined by the left and right phonemes, the generalised transitions are labeled by the left and right broad phonetic classes. Here it is assumed that phonemes in the left and right phonetic class have similar transition characteristics. We use silence + 10 broad phonetic classes. Table 5.9 lists the classes defined based on the ICSI56 phoneme set. This results in 56 (CI phonemes) + 11x11 (CD transitions) = 177 targets. This is comparable to the number of targets we have in HMM state based targets and decision tree clustered CD phonemes. Table 5.10 shows the frame accuracies for the generalized diphone targets for various hidden layer sizes. It can be seen that the frame error rate is higher compared to the HMM state based targets and decision tree clustered CD phonemes. This is due to the difficulty in mapping disparate acoustic realisations into same generalized diphone transition target.

5.2.3 Results

Twenty three context independent phonemes are modeled using 3 state left-to-right HMMs with 8 Gaussians/state and diagonal covariance matrix. Table 5.2 gives the Word Error Rate (WER) for 8 PLP cepstral coefficients, 8 delta and 8 double delta features after

Broad phonetic classes	Phonemes
vowels	iy, ih, eh, ae, ix, ux, ax, uw, uh, ah, ao, aa, er, axr
diphthongs	ey, ay, oy, aw, ow
unvoiced plosives	p, t, k, pcl, kcl, tcl
voiced plosives	b, d, g, bcl, gcl, dcl, dx
nasals	m, n, nx, ng
unvoiced fricatives	f, s, th, sh, hh
voiced fricatives	v, dh, z, zh
affricates	ch, jh
glides	l, y, r, w
syllabics	em, en, el
silence	h#, q

Table 5.9: Broad phonetic classes.

MLP architecture (Input-Hidden-Output)	Training	Cross-validation
216-500-177	33.2(%)	31.1(%)
216-750-177	35.5(%)	32.7(%)
216-1000-177	36.1(%)	34.3(%)

Table 5.10: Frame accuracies for MLP trained on generalized phoneme transition targets.

utterance based mean subtraction and the baseline tandem system. Context dependent (CD) HMM system used 3 states per HMM with 1300 tied states, each modeled by a mixture 8 Gaussians. The dimensionality of the feature vector is too huge to be modeled directly by HMMs. We reduce the dimensionality of the feature vector using PCA. The dimensionality is varied from 56 to 100 to study the effect of dimensionality reduction on Word Error Rate (WER). Table 5.11 shows the performance of Tandem features with different feature vector dimensions. It can be observed that performance of the CD models improved at the expense of the performance of CI models. But the WER reduction obtained by CD targets is not statistically significant compared to the CI phoneme targets. Although the improvement is not significant, it appears that Tandem features with CD targets improve the performance when CD subword units are used as classes in HMMs.

Feature	Dimensionality	CI models	CD models
Tandem CI phoneme targets	56	5.7(%)	4.7(%)
Tandem state targets	56	6.5(%)	4.4(%)
	75	6.6(%)	4.4(%)
	100	7.1(%)	4.7(%)
Tandem decision tree targets	56	6.1(%)	4.6(%)
	75	6.3(%)	4.5(%)
	100	6.9(%)	4.6(%)
Tandem diphone transition targets	56	6.6(%)	4.5(%)
	75	6.9(%)	4.7(%)
	100	7.1(%)	4.7(%)

Table 5.11: Word Error Rates (WER) on connected digit recognition task.

5.3 Summary

We investigated alternative target definitions to CI phonemes. First, we tried to see if we could reduce the number of targets by grouping phonemes based on different clustering schemes. The MLP is trained on these new target classes. Grouping phonemes to reduce the mutual information between classes and features based on the phoneme confusion matrix has shown promising results. We have obtained performance comparable to 56 CI phoneme targets using 34 data derived classes. This results in a classifier with fewer parameters without sacrificing the performance. But the advantages gained by the feature dimensionality reduction vanishes when the complexity of HMMs is increased, by modeling CD subword units or increasing number of Gaussian components. We studied the effect of mismatch in nature of targets used in feature extraction and acoustic modeling. By optimizing the separability of HMM state targets in feature extraction, we observed a marginal improvement in performance with CD subword HMMs.

Chapter 6

Effect of feature training data

In the tandem feature extraction scheme a classifer is used as feature extractor. In the previous chapters we examined various architectures and target classes for tandem. We showed that a MLP trained with softmax nonlinearity in the final layer and one-from-N target coding scheme to estimate posterior probabilities of target classes is better than using a GMM or other MLP architectures. Since the MLP and HMM are trained separately, they can be trained on different databases as well. Current HMM based classifiers require large amounts of task specific training data to achieve competitive performance. In this chapter we investigate whether the features should be trained on a large amount of task independent data to reduce the requirement of task specific training data for the subsequent stochastic model based classifier. By task independent database, we mean a database that is not specific to any task but contains all the variability that is encountered in the test condition. Here the MLP learns to suppress the variability in the data that is not helpful to classification of features and enhances the variability that is helpful. Since the features are already trained, we expect that the HMMs require smaller amounts of task specific training data than when training them directly on acoustic features, such as PLP cepstral coefficients. This is particularly helpful in practical situations where one has very limited task specific data. The ultimate goal of this data-guided feature extraction paradigm is to acquire permanent knowledge from a large amount of task independent training data and use the features in all kinds of speech recognition tasks. In this chapter we systematically study the performance of HMM based speech recognizers as a function of the amount of task specific training data.

The next section compares the performance of features trained on task specific and task independent data. Subsequent sections study performance of the systems by varying the amount of task specific training data.

6.1 Using Both Task Independent and Task Specific Data

We use two databases in our experiments.

- The English part of the OGI-Multilingual Corpus [66], known as OGI-Stories, as task independent data.
- OGI-Numbers as task specific data.

8 PLP cepstral features, its first and second derivates are calculated from the speech signal. The features are then mean and variance normalized over an utterance. The MLP uses 9 frames of normalized cepstral features (9x24=216) as input. It has 500 hidden units and one node per phoneme. The MLP trained on OGI-Stories ($Tand_{Stor}$) has 41 output nodes and the MLP trained on OGI-Numbers ($Tand_{Dig}$) has 23 output nodes. To make the number of features comparable to cepstral features, only the 24 dimensions corresponding to the largest 24 eigenvalues are retained at the output of $Tand_{Stor}$ after PCA. We train Hidden Markov Model (HMM) using HTK [77]. We use 3 state context-dependent HMMs, each state modeled by mixture of 8 Gaussians. HMMs are trained on both OGI-Stories (HMM_{Stor}) and OGI-Numbers (HMM_{Dig}). The Word Error Rates (WER) using various combinations of training and testing using available databases are tabulated in Table 6.1. From Table 6.1 the following things can be observed.

- Tandem features perform better than PLP cepstral features irrespective of the type of training data.
- Training HMMs on the task specific data is better than training on task independent data.
- The Tandem system trained on task independent data ($Tand_{Stor} - HMM_{Stor}$) performs better than the PLP system trained on task independent data ($PLP -$

System	WER (%)
$PLP - HMM_{Stor}$	5.7
$PLP - HMM_{Dig}$	5.1
$PLP - HMM_{Stor+Dig}$	5.3
$Tand_{Stor} - HMM_{Stor}$	5.2
$Tand_{Stor} - HMM_{Dig}$	4.7
$Tand_{Dig} - HMM_{Dig}$	4.4
$Tand_{Stor+Dig} - HMM_{Stor+Dig}$	4.5

Table 6.1: Results using the entire task specific and task independent data.

HMM_{Stor}) and comparable to the PLP system trained on task specific data ($PLP - HMM_{Dig}$).

- The best performance is obtained by training both the MLP and HMM on task specific data ($Tand_{Dig} - HMM_{Dig}$).

6.2 Limited Amount of Task Specific Training Data

In the previous section we compared the performance of Tandem features and PLP features, when all the available task-specific training data was used for training HMMs. To study the effect of WER on the amount of HMM training data, we compare the performance of HMMs trained on PLP features and HMMs trained on tandem features. We use the the tandem features trained on the entire task independent data. and only the HMMs are trained on varying amounts of task-specific data.

The dash-dot line and dash-dash line in Figure 6.1 show the WER as a function of the amount of HMM training data. It can be seen that the performance of the HMM trained on cepstral features (dash-dot) degrades faster with reduction in training data than the HMM trained on tandem features (dash-dash). To confirm that this is actually due to the training of features and not due to discriminative features, we train the MLP and HMM on the same amount of task specific data. The solid line in Figure 6.1 shows the WER when both the MLP and HMM are trained on same amount of task specific data. From the figure it can be observed that the performance of tandem and cepstral features are comparable when the HMMs are trained on the entire task specific data. Also, the

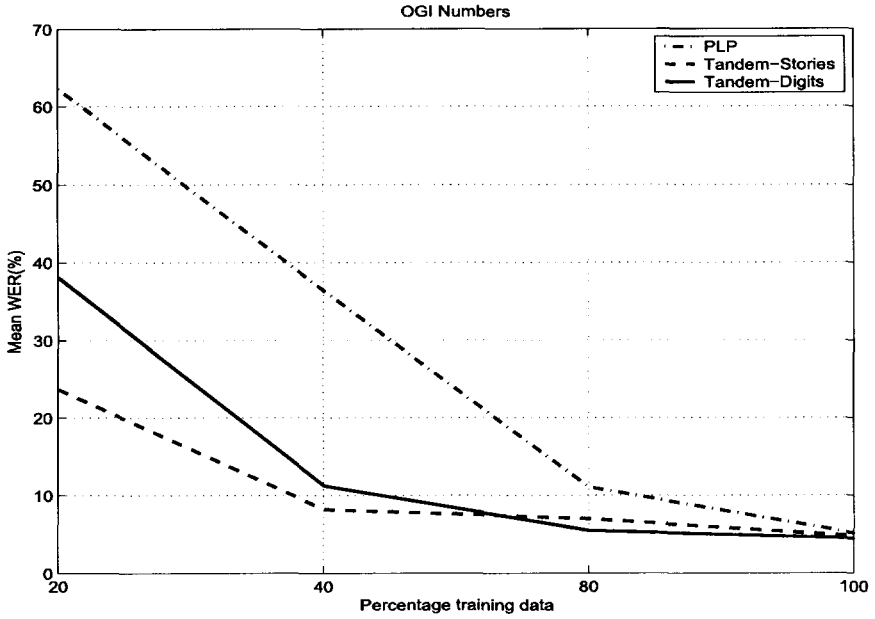

Figure 6.1: Word Error Rates (WER) for continuous digits recognition task as a function of the amount of training data.

difference is greatest when there is less training data. The best performance is obtained when the tandem features are trained on task specific data. This explains why the WER for $Tand_{Dig} - HMM_{Dig}$ is lower than $Tand_{Stor} - HMM_{Dig}$. From the figure, it can be observed that when the training data for MLP is reduced severely (solid line), it starts over-fitting the data and performance on test data suffers. This is evident by the cross-over of $Tand_{Dig} - HMM_{Dig}$ (solid line)performance around 60% training data.

To verify whether this observation holds for another task, we use the Speech In Noisy Environments (SPINE) database [61]. It involves a medium-sized vocabulary of about 5000 words. The data consists of conversations between two communicators working on a collaborative, Battleship-like task in which they seek and shoot at targets. Each person is seated in a sound chamber in which a previously recorded military background noise environment is accurately reproduced. The speech is sampled at 16KHz. PLP cepstral features are extracted from a frame of 25 ms of speech, every 10ms. The feature vector consists of 13 PLP coefficients augmented by deltas and double-deltas. They are then normalized over the utterance to zero mean and unit variance. The input to each MLP is

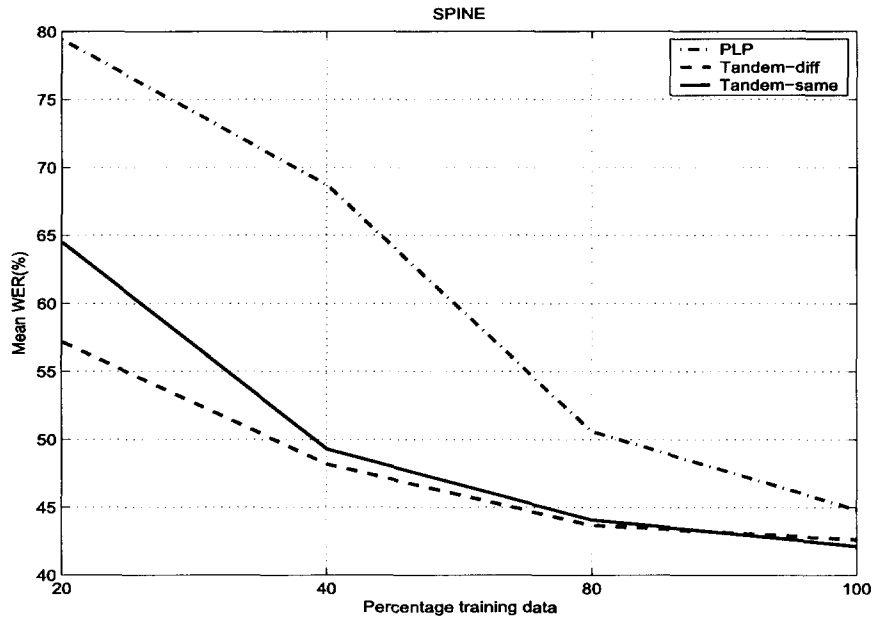

Figure 6.2: Word Error Rates (WER) for large vocabulary recognition task as a function of the amount of training data.

a window of 9 successive feature vectors. The training set is divided into two parts, one is used to train MLP and the other to train HMM to simulate the task specific and task independent data. Figure 6.2 shows the results on SPINE data. The trend is similar to the small-vocabulary test data, except that the WER is higher due to the higher complexity of the task.

To study the situation when the availability of task specific data is very limited, as in many practical situations, we use only 20% of the task specific data. From Figure 6.1 it can be seen that the difference in performance between cepstral features and tandem is the largest when the HMMs are trained with the least amount of data.

6.2.1 Using task independent data together with a small amount of task specific data

We train both HMM and MLP using the entire task independent data and 20% of the task specific data. Here we use only OGI Stories and Digits. Table 6.2 lists the WER for various combinations of training data. The following observations can be made from

System	WER (%)
$PLP - HMM_{Dig_{20\%}}$	62.2
$PLP - HMM_{Stor}$	5.7
$PLP - HMM_{Stor+Dig_{20\%}}$	5.6
$Tand_{Dig_{20\%}} - HMM_{Dig_{20\%}}$	38.1
$Tand_{Stor} - HMM_{Dig_{20\%}}$	23.6
$Tand_{Stor} - HMM_{Stor}$	5.2
$Tand_{Stor+Dig_{20\%}} - HMM_{Stor+Dig_{20\%}}$	5.0

Table 6.2: Results using task independent data and small amount of task specific data.

Table 6.2.

- Using small amounts of the task specific training data to train tandem features and HMM, the WER is reduced by 39% relative to HMM trained on cepstral features with the same amount of training data.
- Using the MLP trained on task independent data to extract features, and training the HMM on small amounts of task specific data, we obtain relative WER reduction of 62% compared to the cepstral system.
- By training the MLP and HMM on the combination of task independent data and a small amount of task specific data, the WER is reduced by 11%.

6.3 Adaptation Using MLLR

An alternate way of utilizing small amounts of task specific training data is to adapt the acoustic models trained on large amount of task independent data. This has been proved to be successful in ASR community [55, 75]. There are two types of adaptation, *supervised* and *unsupervised*. If the true transcription of the adaptation data is known then it is termed supervised adaptation, whereas if the adaptation data is unlabeled then it is termed unsupervised adaptation.

The most successful method of adaptation technique is Maximum Likelihood Linear Regression (MLLR) [55]. MLLR computes a set of transformations that will reduce the mismatch between an initial model set and the adaptation data. More specifically MLLR

is a model adaptation technique that estimates a set of linear transformations for the mean and variance parameters of a Gaussian mixture HMM system. The effect of these transformations is to shift the component means and alter the variances in the initial system so that each state in the HMM system is more likely to generate the adaptation data.

The transformation matrix used to give a new estimate of the adapted mean is given by

$$\hat{\mu} = \mathbf{W}\xi \tag{6.1}$$

where $\mathbf{W}$ is the $nx(n+1)$ transformation matrix (where n is the dimensionality of the data) and ξ is the extended mean vector,

$$\xi = [w\mu_1\mu_2...\mu_n]^T \tag{6.2}$$

where w represents a bias offset whose value is fixed (within HTK) at 1. Hence $\mathbf{W}$ can be decomposed into

$$\mathbf{W} = [\mathbf{bA}] \tag{6.3}$$

where $\mathbf{A}$ represents an nxn transformation matrix and $\mathbf{b}$ represents a bias vector. The transformation matrix $\mathbf{W}$ is obtained by solving a maximization problem using the Expectation-Maximization (EM) technique. This technique is also used to compute the variance transformation matrix.

We use 20% of the task specific data to adapt the acoustic models trained on task independent data. If a small amount of data is available then a global adaptation transform can be generated. A global transform is applied to every Gaussian component in the model set. We use HTK to implement the MLLR adaptation. Table 6.3 gives the performance of the HMMs trained using PLP features on various combination task independent data and task specific data.

From the Table 6.3 it can be seen that when we use the a little amount of task specific data to adapt the acoustic models trained on task independent data, instead of mixing them, the performance of the baseline system becomes comparable to tandem system. This shows that if we have a little amount of task specific data and large amount of

System	WER
$PLP - HMM_{Stor}$	5.7
$PLP - HMM_{Stor+Dig_{20\%}}$	5.6
$PLP - HMM_{Dig}$	5.1
$PLP - HMM_{Stor+MLLR}$	4.7

Table 6.3: Results using MLLR to adapt HMMs trained on task independent with a little amount of task specific data.

task independent data, they can be utilised effectively either using a data-guided feature extraction technique trained on task independent data or adapting the acoustic models using the available task specific data. Since they are two complementary approaches, they can also be combined.

6.4 Summary

In this chapter we addressed the problem of how features trained on large amounts of task independent training data reduces the requirement of task specific training data for the HMM. With small amounts of task specific training data, the tandem system outperforms the cepstral system. This may be due to the knowledge acquired by the tandem features from the task independent data. We showed that the performance of tandem features is superior to cepstral features even when all the available training data is used to train HMM. Another efficient way to utilise the available task specific data is to adapt acoustic models trained on task independent data. The performance of the adapted acoustic models become comparable to tandem system trained on all the available data.

Chapter 7

Summary

In automatic speech recognition, the speech signal is converted to the text of the message it carries. The message is a sequence of words and words are made of sequence of phonemes. The objective in speech recognition is to get the sequence of words right. The signal carries various sources of variabilities, such as the variability due to the message itself, speaker characteristics, ambient noise and characteristics of the recording channel. Feature extraction is the process of reducing the variabilities in the signal that are not important to the task and enhancing the variability of interest. In speech recognition we wish to preserve the variability due to the message. Traditionally features used in speech recognition are based on the short term spectral properties of the units. The short term spectrum undergoes a series of transformations before it is finally used as feature in a classifier. Each transformation addresses different sources of variability present in the signal. These transformations cannot suppress all the unwanted sources of variabilities in the signal. The residual sources of variabilities are modeled by training the classifier on large amounts of data containing all the sources of variabilities.

This thesis investigated a data-guided feature extraction method for automatic speech recognition. In data-guided feature extraction a transformation is derived from the data to enhance the variability of interest. This could be a linear or nonlinear transformation. The objective measure used in deriving the transformation is generally Bayes classification error of phonemes or sub-phoneme speech units. Theoretically the Bayes classification error is minimum when posterior probabilities of classes, or functions of them, are used as features. In this thesis we investigated a method of using posterior probability based features in a hidden Markov model based speech recognition system. The thesis is broadly

divided into three parts. In the first part we study the architecture of the transformation, in the second part we investigate the choice of target classes and the in the last part we analyze the effect of training the feature extractor on task independent data.

In chapter 2 we reviewed feature extraction methods used in automatic speech recognition. We reviewed the most commonly used short-term features based on Short Time Fourier Transform of speech signal, various auditory like modifications applied to it before applying to a hidden Markov Model (HMM) based classifier. We described data-guided feature extraction and various design issues involved. Two of the most popular data-guided feature extraction methodologies are explained, one based on parametric representation of input features and the other based on a connectionist approach. We reviewed some of the existing data-guided feature extraction methods such as Linear Discriminant Analysis (LDA), Tandem and TempoRAl PatternS (TRAPS).

The next chapter presented the motivation for Tandem feature extraction. The training methodology was explained in detail. We studied the statistics of Tandem features and how they could be used as features in a HMM based speech recognizer. It was shown that posterior probabilities have a skewed distribution that is difficult to be modeled by Gaussian mixture model based classifiers. Applying a logarithmic nonlinearity makes the features more gaussian. Removing the softmax nonlinearity in the final layer of the MLP during the forward pass gives a smoother distribution that can be easily modeled without loss of discriminability.

Various architectures for Tandem were explored in chapter 4. We investigated discriminative and generative transformations of acoustic feature space. Gaussian mixture models (GMM) trained in a maximum likelihood manner using the expectation maximization algorithm were used as a generative estimator of posterior probabilities. We compared discriminatively trained GMM with its generative version and showed the discriminant training is better. Further we showed that using an MLP to estimate posterior probabilities in a discriminative manner is a better transformation. MLP has the advantage that it makes fewer assumptions about statistics of the input feature space. We investigated a hierarchical architecture for tandem feature extraction. Thus task of discriminating between all the classes has been converted into discriminating subsets and independently

discriminating the classes within each subset. The posterior probability for a specific class was computed by multiplying all the conditional posteriors from root node to the leaf corresponding to the specific class. We compared monolithic versus hierarchical architecture for feature extraction and found marginal improvement in the case of hierarchical features.

HMMs use context dependent phonemes as units. The effect of difference in targets used in feature extraction and the classes modeled by HMM is studied in chapter 5. We studied broad phonetic categories, context independent phonemes, context dependent phoneme based targets. By using context independent phonemes as targets in feature extraction, the context variability in the initial feature space could be reduced. We analyzed the use of targets with context variability. The context dependent targets were derived by clustering the context dependent phoneme to a manageable number. The performance of the features with context dependent targets were only marginally better than the context independent targets.

Finally the effect of training the transformation and stochastic classifier on different types of databases is studied. We investigated the situation where there is a large amount of task independent training data available and only a small amount of task dependent training data. We showed that the feature transformation, when trained on task independent data, performs better than the PLP cepstral features. This showed the generalizability of the feature transformation, though the best performance was obtained when both features and classifier were trained on task specific data.

7.1 Contributions

The contributions of this thesis can be summarized as follows.

- **Nonlinear discriminative feature extraction** We compared various linear and nonlinear discriminative feature extraction methods, such as Linear Discriminant Analysis (LDA), Gaussian Mixture Models (GMM) and Multi-Layer Perceptrons (MLP). Since the speech recognition consisted of two or more classifiers in tandem, the feature extraction method is called tandem feature extraction. LDA is an optimal feature transformation method under the assumptions that features are normally

distributed. This is not usually the case. We showed that nonlinear discriminative feature extraction using a MLP is better than LDA. MLP is not constrained by the statistics of input feature space to derive an optimal feature transformation. This thesis studied various models for estimating posterior probabilities. We show that discriminatively estimating posterior probabilities is better than maximum likelihood estimation. We investigate various architectures for estimating the posteriors. A hierarchical method of computing the features was investigated. The hierarchical tree was constructed based on the phonetic properties of the target classes used. It was shown that computing features in a hierarchical manner provides better discrimination.

- **Effect of training data in feature estimation** We showed that the nonlinear feature transformation generalizes across different databases. We showed reduction in word error rate in a connected digit recognition task when the MLP was trained on a task independent database. When the availability of task specific data is limited, feature extractor trained on large amounts of task independent data is an elegant way of compensating for the lack of task specific knowledge. Adaptation of the classifier, trained on task independent data, using the limited task specific data is an alternative method. The best performance was obtained when we utilized all the available data, both task independent and task specific, to train both the feature extractor and classifier.

- **Target classes in data-guided feature estimation** Hidden Markov model (HMM) based speech recognizers typically model context dependent phonemes. Ideally the feature extractor must be optimizing he separability of the classes modeled by the classifier. Since these recognizers often incorporate many tens or even hundreds of thousands of such units, it is a challenging task to reliably estimate the posteriors. We studied the effect of using context independent and context dependent classes in calculating the features. We derived new target classes by clustering context independent phonemes and show that they perform equally well as context independent phoneme targets when HMMs also use context independent phonemes

as units. Use of clustered context dependent phoneme targets provide only marginal improvement over context independent targets.

7.2 Future Directions

We presented a data-guided feature extraction scheme which consists of a nonlinear discriminative transformation of the input feature space to functions of posterior probabilities of the target classes. We showed that it performed better than the best performing PLP cepstral features. It has the limitation the use of features are limited by the choice of target classes. To use it in a different language, we need to retrain the neural net on a different set of phonemes present in the application. This work can be extended further to train the neural net on a set of targets that are universal across various languages. Another possibility is to derive classes automatically from the data. To test the generalization ability and task independence of the MLP, a big MLP has to be trained on large amounts of data and has to be tested on various tasks.

Appendix A

ICSI 56 Phoneme Set

Symbol	Description	Symbol	Description	Symbol	Description
p	*p*ea	t	*t*ea	k	*k*ey
pcl	*p closure*	tcl	*t closure*	kcl	*k closure*
b	*b*ee	d	*d*ay	g	gay
bcl	*b closure*	dcl	*d closure*	gcl	*g closure*
ch	*ch*oke	dx	dir*t*y	jh	*j*oke
th	*th*in	dh	*th*en	f	*f*ish
v	*v*ote	s	*s*ound	z	*z*oo
sh	*sh*out	zh	a*z*ure	hh	*h*ay
hv	a*h*ead	m	*m*oon	n	*n*oon
ng	si*ng*	em	bottom	en	button
nx	wi*nn*er	l	*l*ike	el	bott*le*
r	*r*ight	er	bird	axr	butt*er*
y	*y*es	w	*w*ire	iy	b*ee*t
ih	b*i*b	ey	b*ai*t	eh	b*e*t
ae	b*a*t	aa	f*a*ther	ao	b*ough*t
ah	b*u*t	ow	b*oa*t	uh	b*oo*k
uw	b*oo*t	ix	deb*i*t	aw	*ou*t
ay	b*i*te	oy	b*oy*	ax	ab*ou*t
h#	*silence*				

Table A.1: ICSI56 phoneme set.

Bibliography

[1] A. K. JAIN. *Fundamental of Digital Image Processing.* Prentice Hall, New Jersey, 1989.

[2] A.P. DEMPSTER, N.M. LAIRD AND D.B. RUBIN. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society 39* (1977), pp. 1–38.

[3] B. CHEN, Q. ZHU AND N. MORGAN. Learning long-term temporal features in LVCSR using neural networks. In *Proceedings of ICSLP 2004* (Jeiju, South Korea, October 2004).

[4] B. LAUNAY, O. SIOHAN, A. SURENDRAN AND C.H. LEE. Towards knowledge-based features for hmm based large vocabulary automatic speech recognition. In *Proceedings of ICASSP* (May 2002), pp. 817–820.

[5] C. M. BISHOP. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, 1999.

[6] D. ELLIS AND J. BILMES. Using mutual information to design feature combinations. In *Proceedings of ICSLP* (Beijing, China, 2000).

[7] D. H. KLATT. Review of the ARPA speech understanding project. *The Journal of the Acoustical Society of America 62* (1977), pp. 1345–1366.

[8] D. LOWE AND A. R. WEBB. Optimized feature extraction and the bayes decision in feed-forward classier networks. *IEEE Transactions on PAMI 13* (April 1991), pp. 355–364.

[9] D. REYNOLDS. Speaker identification and verification using gaussian mixture models. *Speech Communication 17* (1995), pp. 91–108.

[10] D.E.RUMELHART, G.E. HINTON AND R.J. WILLIAMS. *Learning internal representations by error propagation*, vol. 1. MIT Press, 1986, pp. 318–362.

[11] D.LOWE. *The Handbook of Brain Theory and Neural Networks.* MIT Press, 1995, pp. 779–782.

[12] D.P.W. ELLIS AND M.J. REYES GOMEZ. Investigations into tandem acoustic modeling for the aurora task. In *Proceedings of Eurospeech-01, Special Event on Noise Robust Recognition* (Denmark, 2001), pp. 189–193.

[13] D.R. REDDY. Speech recognition by machine: A review. *Proceedings of the IEEE 64* (1976), pp. 502–531.

[14] E. BATLLE, C. NADEU AND J. A. R. FONOLLOS. Feature decorrelation methods in speech recognition. A comparative study. In *Proceedings of International Conference on Spoken Language Processing* (Sydney, Australia, 1998), pp. 951–954.

[15] E. EIDE. Distinctive features for use in an automatic speech recognition system. In *Proceedings of Eurospeech* (September 2001), pp. 1613–1616.

[16] E. ZWICKER, G. FLOTTORP AND S. S. STEVENS. Critical bandwidth in loudness summation. *The Journal of the Acoustical Society of America 29* (1957), pp. 548–557.

[17] G. DODDINGTON. Phonetically sensitive discriminants for improved speech recognition. In *Proceedings of ICASSP'89* (Glasgow, Scotland, 1989), IEEE, pp. 556–559.

[18] G. RIGOLL AND D. WILLETT. A NN/HMM Hybrid for Continuous Speech Recognition with a Discriminant Nonlinear Feature Extraction. In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (Seattle, 1998), pp. 9–12.

[19] H. BOURLARD AND C.J. WELLEKEN. Links between markov models and multilayer perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12* (1990), pp. 1167–1178.

[20] H. BOURLARD AND N. MORGAN. *Connectionist Speech Recognition - A Hybrid Approach.* Kluwer Academic Publishers, Boston, 1994.

[21] H. BOURLARD AND S. DUPONT. A new ASR approach based on independent processing and re-combination of partial frequency bands. In *Proceedings of ICSLP* (Philadephia,PA, 1996), vol. 1, pp. 426–429.

[22] H. FLETCHER. Speech and hearing in communication. In *The ASA Edition of Speech and Hearing in Communication* (New York, 1994), J. Allen, Ed., Acoustical Society of America.

[23] H. G. HIRSCH AND D. PEARCE. The aurora experimentral framework for the performance evaluations of speech recognition systems under noisy conditions. *ISCA, ITRW, ASR 2000* (2000).

[24] H. HERMANSKY. Perceptual linear predictive (PLP) analysis for speech. *The Journal of The Acoustical Society of America 87* (Apr. 1990), pp. 1738–1752.

[25] H. HERMANSKY. The modulation spectrum in automatic recognition of speech. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding* (1997), S. Furui, B.-H. Juang and W. Chou, Ed., IEEE Signal Processing Society, pp. 140–147.

[26] H. HERMANSKY AND N. MALAYATH. Spectral basis functions from discriminant analysis. In *Proceedings of ICSLP* (Sydney, Australia, 1998), pp. 1379–1382.

[27] H. HERMANSKY AND N. MORGAN. RASTA processing of speech. *IEEE Transactions on Speech and Audio 2* (Oct. 1994), pp. 578–589.

[28] H. HERMANSKY AND P. JAIN. Band-independent speech-event categories for TRAP based ASR. In *Proceedings of Eurospeech* (2003), pp. 1013–1016.

[29] H. HERMANSKY AND P. JAIN. Beyond a single critical-band in TRAP based ASR. In *Proceedings of Eurospeech* (2003), pp. 437–440.

[30] H. HERMANSKY AND S. SHARMA. TRAPs Classifiers of Temporal Patterns. In *Proceedings of ICSLP'98* (Sydney, Australia, 1998), vol. 2, pp. 615–618.

[31] H. HERMANSKY, D. ELLIS, AND S. SHARMA. Tandem connectionist feature extraction stream extraction for conventional HMM systems. In *Proceedings of ICASSP* (Istanbul, 2000), pp. 1635–1638.

[32] H. HERMANSKY, S. SHARMA. Temporal Patterns (TRAPS) in ASR of Noisy Speech. In *Proceedings of ICASSP'99* (Phoenix, Arizona, USA, 1999), pp. 663–666.

[33] H. HERMANSKY, S. SHARMA AND P.JAIN. Data driven nonlinear mapping for HMM based ASR. In *Proceedings of ASRU'99* (Colorado, USA, 1999).

[34] J. FRITSCH. *Modular Neural Networks for Speech Recognition.* PhD thesis, Carnegie Mellon University, Pittsburgh, USA, July 1996.

[35] J. MAKHOUL. Linear prediction: A tutorial review. *Proceedings of the IEEE 63* (1975), pp. 561–580.

[36] J. SCHUERMANN AND W. DOSTER. A decision theoretic approach to hierarchical classifier design. *Pattern Recognition 17*, 3 (1984), pp. 359–369.

[37] J.B. ALLEN. Short term spectral analysis, synthesis and modification by discrete fourier transform. *IEEE - ASSP 25* (1977), pp. 235–238.

[38] J.B. ALLEN. How do humans process and recognize speech? *IEEE Transactions on Speech and Audio Processing 2* (1994), pp. 567–577.

[39] K. FUKUNAGA. *Introduction to Statistical Pattern Recognition.* Academic Press, Indiana, 1990.

[40] K. FUKUNAGA. *Statistical Pattern Recognition.* Academic Press, Indiana, 1990.

[41] K. HORNIK, M. STINCHCOMBE AND H. WHITE. Multilayer networks are universal approximators. In *Neural Networks* (1989), vol. 2, pp. 359–378.

[42] K. KIRCHHOFF. Robust speech recognition using articulatory information. Tech. Rep. TR-98-037, ICSI, 1998.

[43] K. WANG AND S. SHAMMA. Representation of acoustic signals in the primary auditory cortex. *IEEE Transactions on Speech and Audio Processing 3* (1995), pp. 382–395.

[44] K. WEBER, S. BENGIO AND H. BOURLARD. HMM2- Extraction of Formant Features and their Use for Robust ASR. In *European Conference on Speech Communication and Technology (Eurospeech 2001)* (Aalborg, Denmark, September 2001), pp. 607–610.

[45] L. K. SAUL, M. G. RAHIM, AND J. B. ALLEN. A statistical model for robust integration of narrowband cues in speech. *Computer Speech and Language 15* (2001), pp. 175–194.

[46] L.R. RABINER. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE 77* (Feb. 1989), pp. 257–285.

[47] M. HUNT. Spectral Signal Processing for ASR. In *Proceedings of ASRU* (Colorado, USA, 1999), pp. 17–25.

[48] M. HUNT AND C. LEFEBVRE. A comparison of several acoustic representations for speech recognition with degraded and undegraded speech. In *Proceedings of ICASSP* (Glasgow, Scotland, 1989), pp. 262–265.

[49] M. HUNT, S. RICHARDSON, D. BATEMAN AND A. PIAU. An investigation of plp and imelda acoustic representations and of their potential for combination. In *Proceedings of ICASSP'91* (Toronto, Canada, 1991), pp. 881–884.

[50] M. PAVEL AND H. HERMANSKY. Temporal masking in automatic speech recognition. *The Journal of The Acoustical Society of America 95* (May 1994), pp. 2876–2880.

[51] M. PORTNOFF. Time-frequency representation of digital signals and systems based on short-time fourier analysis. *IEEE Transactions on Acoustics, Speech and Signal Processing 28* (Feb. 1980), pp. 55–69.

[52] M.I. JORDAN AND R.A. JACOBS. Hierarchical Mixtures of Experts and the EM algorithm. *Neural Computation 6* (1994), pp. 181–214.

[53] M.I. JORDAN AND R.A. JACOBS. *The Handbook of Brain Theory and Neural Networks.* MIT Press, 1995, ch. Modular and Hierarchical Learning Systems, pp. 579–581.

[54] M.J. HUNT. A statistical approach to metrics for word and syllable recognition. *The Journal of The Acoustical Society of America 66* (1979), pp. S1–S35.

[55] M.J.F. GALES AND P.C.WOODLAND . Mean and variance adaptation within the mllr framework. *Computer Speech & Language 10* (1996), pp. 249–264.

[56] M.N. STUTTLE AND M.J.F. GALES. A mixture of gaussians front end for speech recognition. In *Proceedings of Eurospeech* (September 2001), pp. 675–678.

[57] N. MALAYATH. *Data-Driven Methods for Extracting Features from Speech.* PhD thesis, Oregon Graduate Institute of Science and Technology, Portland, USA, Jan. 2000.

[58] N. MERHAV AND C.H. LEE. On the asymptotic statistical behavior of empirical cepstral coefficients. *IEEE Transactions on Signal Processing 41* (May 1993), pp. 1990–1993.

[59] N. MORGAN AND H. BOURLARD. Factoring networks by a statistical method. *Neural Computation 4*, 6 (1992), pp. 835–838.

[60] N. MORGAN AND H. BOURLARD. Continuous speech recognition. *IEEE Signal Processing Magazine* (May 1995), pp. 25–42.

[61] SPINE, Navy Research Laboratory. http://elazar.itd.nrl.navy.mil/spine/, 2001.

[62] P. BROWN. *The Acoustic-Modelling Problem in Automatic Speech Recognition.* PhD thesis, Computer Science Department, Carnegie Mellon University, 1987.

[63] P. GALLINARI, S. THIRIA, F. BADRAN AND F.FOGELMAN-SOULIE. On the relations between discriminant analysis and multilayer perceptrons. *Neural Networks 2* (1991), pp. 349–360.

[64] P. MERMELSTEIN. Distance measures for speech recognition, psychological and instrumental. In *Pattern Recognition and Artificial Intelligence* (New York, 1976), C.H. Chen, Ed., pp. 374–388.

[65] R. A. FISHER. The use of multiple measurements in taxonomic problems. *Annals of Eugenics 7* (1936), pp. 179–188.

[66] R. COLE, M. NOEL AND T. LANDER. Telephone speech corpus development at CSLU. In *Proceedings of ICSLP* (Yokohama, Japan, 1994), pp. 1815–1818.

[67] R. DUDA, P. HART AND D. STORK. *Pattern Classification.* Wiley Interscience, 2001.

[68] R. HAEB-UMBACH AND H. NEY. Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *Proceedings of ICASSP'92* (San Francisco, USA, 1992), vol. 1, IEEE, pp. 13–16.

[69] R.P. LIPPMANN. Recognition by humans and machines: Miles to go before we sleep. *Speech Communication* (May 1996), Volume 18, Issue 3, Pages 247–248.

[70] S. DUPONT, C. RIS, O. DEROO, V. FONTAINE, J.M. BOITE AND L. ZANONI. Context independent and context dependent hybrid HMM/ANN systems for training independent tasks. In *Proceedings of EUROSPEECH'97.* (Rhodes, Greece, 1997), vol. 4, pp. 1947–1950.

[71] S. FURUI. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions of ASSP 34* (1986), pp. 52–59.

[72] S. KAJAREKAR AND H. HERMANSKY. Analysis of information in speech and its application in speech recognition. In *Proceedings of TSD* (Brno, Czech Republic, 2000), pp. 283–288.

[73] S. SHARMA, D. ELLIS, S. KAJAREKAR, P. JAIN AND H. HERMANSKY. Feature Extraction Using Non-linear transformation for robust speech recognition on the AURORA data-base. In *Proceedings of ICASSP* (Istanbul, Turkey, 2000), pp. 1117–1120.

[74] S. TIBREWALA AND H. HERMANSKY. Sub-band based recognition of noisy speech. In *Proceedings of ICASSP'97* (1997), vol. 2, IEEE, pp. 1255–1258.

[75] S. YOUNG. Large vocabulary continuous speech recognition: A review. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding* (1995), pp. 3–28.

[76] S.B. DAVIS AND P. MERMELSTEIN. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustic, Speech and Signal Processing 28* (Aug. 1980), pp. 357–366.

[77] S.J. YOUNG. The HTK Hidden Markov Model Toolkit: Design and Philosophy. Tech. Rep. 153, Cambridge University, UK, 1993.

[78] S.J. YOUNG, J. O., AND WOODLAND, P. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of ARPA Workshop on Human Language Technology* (1994), pp. 307–312.

[79] S.J. YOUNG AND P.C. WOODLAND. The use of state tying in continuous speech recognition. In *Proceedings of Eurospeech* (1993), pp. 2203–2206.

[80] T. EISELE, R. HAEB-UMBACH AND D. LANGMANN. A comparative study of linear feature transformation techniques for automatic speech recognition. In *Proceedings of International Conference on Spoken Language Processing* (Philadelphia, USA, 1996), pp. 252–255.

[81] T. M. COVER AND J. A. THOMAS. *Elements of Information theory.* John Wiley & Sons, Inc., New York, 1991.

[82] T.J.HASTIE AND R.J.TIBSHIRANI. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Series B 58* (1996), pp. 155–176.

[83] V. FONTAINE, C. RIS AND J.M. BOITE. Nonlinear discriminant analysis for improved speech recognition. In *Proceedings of EUROSPEECH'97* (Rhodes, Greece, 1997), pp. 2071–2074.

[84] V.R. LESSER, R.D. FENNEL, L.D. ERMAN AND D.R. REDDY. Organization of the Hearsay-II speech understanding system. *IEEE Transactions on Acoustics, Speech and Signal Processing 23* (1975), pp. 11–23.

[85] X.D. HUANG, F. ALLEVA, H.W. HON, M.Y. HWANG AND R. ROSENFELD. The SPHINX-II speech recognition system: an overview. *Computer Speech and Language 7*, 2 (1993), pp. 137–148.

[86] Y.R. BENGIO, R. DE MORI, G. FLAMMIA AND R. KOMPE. Global optimization of a neural-hidden markov model hybrid. *IEEE Transactions on Neural Networks 3* (1992), pp. 252–258.

[87] ZUE, V. The use of speech knowledge in automatic speech recognition. *Pattern Recognition and Artificial Intelligence, Alex Waibel and Kai-Fu Lee, Eds.* (1990), pp. 200–213.

Biographical Note

Sunil Sivadas was born in Palakkad, Kerala, India, on the 5th of May, 1973. He completed the Bachelor of Technology degree in 1995 from the NSS college of engineering, Palakkad. From 1995 to 1996 he was a Project Assistant in Speech processing lab, Indian Institute of Science (IISc), Bangalore, India. He was responsible for research and implementation of keyword spotting algorithms. In 1998 he obtained Master of Technology degree from Center for Electronic Design & Technology (CEDT). Between 1998 and 1999 he worked on voice based name dialing for mobile phones, an Ericsson USA sponsored project, in Speech processing lab, IISc. In the fall of 1999, he joined Anthropic Signal Processing Group, Oregon Graduate Institute as a Ph.D. student. Under the guidance of Prof. Hynek Hermansky he worked on noise robust feature extraction for speech recognition for Distributed Speech Recognition (DSR), large vocabulary conversation speech recognition and speaker verification. Between 2000 and 2002 he was a member of the Qualcomm-ICSI-OGI team working on the ETSI AURORA advanced front-end for DSR. In 2001 he was a summer intern at International Computer Science Institute (ICSI), Berkeley, Caligornia, where he developed noise compensation algorithms for feature extraction. He has authored several conference papers.

Selected Publications

1. S. Sivadas, P. Jain and H. Hermansky, "Discriminative MLPs In HMM-Based Recognition Of Speech in Cellular Telephony", *in Proc. of ICSLP 2000*, pp. , Beijing, China.

2. C. Benitez, L. Burget, B. Chen, S. Dupont, H. Garudadri, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan and S. Sivadas, "Robust ASR front-end using spectral-based and discriminant features: experiments on the Aurora tasks ", *in Proc. of*

Eurospeech 2001, pp. ,Aalborg, Denmark, September 2001, pp. 429–432.

3. D. Ellis, R. Singh and S. Sivadas, "Tandem Acoustic Modeling In Large-Vocabulary Recognition", *in Proc. of ICASSP 2001*, pp. ,Salt Lake City, USA, pp. 517–520.

4. S. Sivadas and H. Hermansky, "Hierarchical Tandem Feature Extraction", *in Proc. of ICASSP 2002*, pp. ,Orlando, Florida, USA, pp. 809–812.

5. A. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grezl, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan and S. Sivadas, "QUALCOMM-ICSI-OGI Features for ASR", *in Proc. of ICSLP 2002*, pp. ,Denver, Colorado, USA, Sep, 2002, pp. 21–24.

6. S. Sivadas and H. Hermansky, "Generalized Tandem Feature Extraction", *in Proc. of ICASSP 2003*, pp. ,HongKong, April 2003, pp. 56–59.

7. S. Sivadas and H. Hermansky, "In Search Of Target Class Definition In Tandem Feature Extraction", *in Proc. of Eurospeech 2003*, pp. ,Geneva, September 2003, pp. 837–840.

8. B. Chen, S. Chang and S. Sivadas, "Learning Discriminative Temporal Patterns in Speech: Development of Novel TRAPS-Like Classifiers", *in Proc. of Eurospeech 2003*, pp. ,Geneva, September 2003, pp. 853–856.

9. S.Sivadas and H. Hermansky, "On Use of Task Independent Training Data in Tandem Feature Extraction", *in Proc. of ICASSP 2004*, pp. ,Montreal, May 2004, pp. 541–544.

10. G. Aradilla, J. Dines and S. Sivadas, "Using RASTA in task independent TANDEM feature extraction", *to appear in Proc. of ICSLP 2004*, Jeju, Korea, October 2004.

11. S. Ikbal, H. Misra, S. Sivadas, H. Hermansky and H. Bourlard, "Entropy Based Combination of Tandem Representations for Noise Robust ASR", *to appear in Proc. of ICSLP 2004*, Jeju, Korea, October 2004.