ACOUSTIC FEATURE DESIGN FOR SPEECH RECOGNITION, A STATISTICAL INFORMATION-THEORETIC APPROACH

BY

MOHAMED KAMAL MAHMOUD OMAR

BECEngr, Cairo University, 1995
MEngr, Cairo University, 1999

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2003

Urbana, Illinois

ACOUSTIC FEATURE DESIGN FOR SPEECH RECOGNITION,
A STATISTICAL INFORMATION-THEORETIC APPROACH

Mohamed Kamal Mahmoud Omar, Ph.D.
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, 2003
Mark Hasegawa-Johnson, Adviser

Bayesian classifiers rely on models of the *a priori* and class-conditional feature distributions; the classifier is trained by optimizing these models to best represent features observed in a training corpus according to a certain criterion. In many problems of interest, the true class-conditional feature probability density function (PDF) is not a member of the set of PDFs the classifier can represent.

This dissertation addresses this model mismatch problem. We formulate it as the problem of minimizing the relative entropy between the true conditional probability density function and the hypothesized probabilistic model. Based on this formulation, we provide a computationally efficient solution to the problem based on volume-preserving maps; existing linear transform designs are shown to be special cases of the proposed solution. We apply this approach to automatic speech recognition (ASR) systems. We describe an iterative algorithm to estimate the parameters of both a class of nonlinear volume-preserving feature transforms and the hidden Markov model (HMM) that jointly optimize the objective function for an HMM-based ASR system.

In the second part of this work we present a generalization of linear discriminant analysis (LDA) that optimizes a discriminative criterion and solves the problem in the lower-dimensional subspace. We start with showing that the calculation of the LDA projection matrix is a maximum mutual information estimation problem in the lower-dimensional space with some constraints on the model of the joint conditional and unconditional PDFs of the features, and then, by relaxing these constraints, we develop a dimensionality reduction approach that maximizes the conditional mutual information between the class identity and the feature vector in the lower-dimensional space given the recognizer model.

To my mother Nagat, and the soul of my father Kamal.

ACKNOWLEDGMENTS

This dissertation would not have been possible without the the support of my advisor Mark Hasegawa-Johnson. I would like to thank him for his insightful suggestions, comments, and feedback throughout this work. It was an honor having Thomas Huang and Stephen Levinson on my committee. I would like to thank Thomas Huang for his encouragement and advice to investigate possible applications of my work other than speech recognition. Thanks to Stephen Levinson for his suggestions that improved the way my ideas are presented. Thanks also to committee member Minh Do, whose discussions concerning the generality of some ideas of my work emphasized the importance of explaining this issue in the dissertation and the articles published on this work. Finally, thanks to my wife Ingy, whose patience and support allowed me to assign most of my time to the work presented here.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The first stage in many pattern recognition and coding tasks is to generate a good set of features from the observed data. The set should be compact and capture all class-discriminating information in case of recognition and all information needed to reconstruct the observed data with sufficient quality in case of coding. Features that contain little or no information should be avoided since they increase the computational load and the storage and transmission requirements without improving the performance. The features also should satisfy the assumptions imposed on them by the recognizer or the decoder.

The purpose of this work is to construct acoustic features for automatic speech recognition that are optimized based on certain information-theoretic criteria to achieve the goals of compactness, discrimination, and satisfaction of the recognizer's assumptions.

1.1 Automatic Speech Recognition

Many of the approaches presented here can be applied to any recognition or classification problem, but we choose to test these approaches by applying them to the automatic speech recognition problem.

The problem of automatic speech recognition (ASR) is the problem of generating the text that corresponds to a given speech waveform. Our discussion of this problem will be confined to the statistical approach, not only because it recognizes the probabilistic nature both of the waveform we seek to process and of the form in which we should express the

results, but also because it is the most successful approach to the ASR problem up to now. The statistical approach allows also a rigorous formulation of the feature design problems and their solutions.

Current statistical speech recognition systems use many sources of information to accomplish their task. They use acoustic measurements usually at a fixed time-frequency resolution. These measurements are used by the acoustic-phonetic model to achieve a mapping from the speech waveform to some phonetic units that represent the different kinds of sounds that are encountered in a language. Other measurements like visual measurements are sometimes used in addition to acoustic measurements to achieve this mapping. In addition to these measurements, lexical information, and the likelihood of different word sequences are also used to achieve this mapping and also the mapping of the phonetic units sequence to a sequence of words.

Despite a great deal of success as indicated by current commercial products, current systems have significant problems. These problems are caused by a number of different factors, including coarticulation, change in speaking rate, speaker accent, and ambient noise conditions.

In this dissertation, we will focus on the acoustic-phonetic part of the ASR systems. More specifically, the dissertation focuses on the problem of acoustic representation of the speech signal without any assumptions about the underlying probabilistic model. It could be the traditional hidden Markov model (HMM), Bayesian networks, or any other probabilistic model. However, our examples and experiments use the Gaussian mixture HMM as the underlying probabilistic model. This is because no other model has provided significantly better recognition results, and because we wish to compare our results with the best recognition results achieved using HMM. As will be discussed later, the acoustic feature design approaches proposed here may provide solutions to some of the problems that prevented the extensive use of more sophisticated probabilistic models than HMM in speech recognition.

1.2 Feature Extraction for Speech Recognition

The term "features" is sometimes ambiguous specially in hybrid systems, where the output of one system is fed as an input to the other system. To avoid this ambiguity, we always mean by the term "features" in this dissertation the set of measurements estimated from the speech waveform whose conditional probability density functions are modeled by ASR systems to accomplish their recognition task.

The objective of speech signal analysis for ASR systems is to produce a parameterization of the speech signal that reduces the amount of data that is presented to the speech recognizer, separates all information relevant for the recognition task from irrelevant information (e.g., speaker or channel characteristics), discriminates among different phonemes, and finally satisfies the ASR system's assumptions. The principal assumption of most statistical speech recognition systems is that the input features are approximately independent or at least decorrelated given the values of some hidden variable. The importance of this assumption is due to the extreme increase in the number of parameters required to model the joint density of the features, if the features are not approximately independent or decorrelated.

Speech data can be parameterized in many different ways. The two main approaches are some type of coding—usually linear prediction—of the time domain, and direct sampling of domains other than the time domain, usually the frequency or cepstral domains [1]. In both approaches, the input speech samples are windowed and the resulting speech segments termed frames. The data analysis is then executed on each frame, which corresponds to a single observation with regard to a hidden Markov model (HMM). Mel-frequency cepstrum coefficients (MFCC) and perceptual linear prediction cepstrum coefficients (PLPCC) are the current most successful features for speech recognition systems. Both were motivated by the study of human speech production and perception [2]. They try to approximately separate the linguistic information related to the vocal tract shape from other sources of variations due to the excitation source that are speaker-dependent. They try also to use concepts based

on human speech perception like Mel-frequency scaling and critical band filters to simulate the front-end of the human auditory system. Both use discrete cosine transform (DCT) to generate the features because DCT approximates the Karhunen-Loève transform (KLT) for a first-order Gaussian-Markov random process. This means that, under the assumption that frequency samples of the speech log spectrum are a first-order Gaussian-Markov random process, the output coefficients are approximately decorrelated. Since speech is a continuous signal that has continuity constraints due to the human speech production system, it is not enough to represent it with a discrete sequence of features, and it is necessary to include features representing the temporal correlation of the speech frames. This is usually achieved by appending the delta and delta-delta coefficients to the cepstrum coefficients in the feature vector. The delta coefficients are usually generated by implementing a difference equation on the cepstral coefficients of a window of frames centered at the current frame. The delta-delta coefficients are generated from the delta coefficients using the same method. This standard feature vector is more compact and less sensitive to the excitation signal and speaker variations than the speech waveform itself. There are also many techniques that further reduce the sensitivity of cepstral coefficients to speaker variations like vocal tract length normalization and speaker-adaptive training [3]–[5]. The sensitivity of cepstral coefficients to environmental noise is an important subject of such extensive research that many important speech conferences dedicate special sessions for it. The main reason of the difficulty of this problem is that even simple additive noise in the time domain is combined non-linearly with the speech signal in the cepstral domain. Even with these additional techniques for speaker normalization and combating environmental noise, incorporating properties of human speech production and auditory perception is not necessarily the optimal approach to feature extraction for speech recognition, as they do not achieve the goals of discrimination and model satisfaction mentioned before. The feature vector based on cepstral coefficients and their deltas completely violates the decorrelation assumption, as part of the feature vector is an explicit function of the other part. They also completely ignore the need for

discriminative features that is essential for a pattern recognition system. There were recently some attempts to address these problems and they will be described in detail in Chapter 2. The main goal of this dissertation is to describe methods to improve the discrimination and satisfaction of model assumptions by the acoustic features. These methods do not have the limitations of previous methods.

1.3 Motivations of Data-Driven Acoustic Feature Design For ASR

ASR in general and specially acoustic modeling of speech are well-studied problems, so one may wonder whether any research for further improvement is required. Lippmann has gathered in [6] machine recognition results on speaker-independent corpora and compared them with human recognition results. Table 1.1 summarizes the characteristics of the corpora and the corresponding results for human and machine.

The table indicates that humans are clearly superior to machines and the room for improvement is still wide. This indicates that the components of the current speech recognition systems should be improved to get closer to human performance. When designing a machine learning system, there is always a trade-off between the complexity of the recognition algorithm and the complexity of the feature extractor. In this work, we investigate the possibility of improving the ASR systems by using more sophisticated feature extraction modules without changing the recognizer complexity.

We will discuss briefly in the following the importance of designing acoustic features whose true joint conditional probability density function (PDF) can be approximated well by the recognizer, and of optimizing the features to discriminate among different phonemes. Then, we will discuss the advantage of using class-dependent features instead of one global feature vector for all speech units.

Table 1.1 Human versus machine speech recognition.

Corpus and Description	Vocabulary Size	Recognition Perplexity	Machine Error (%)	Human Error (%)
TI Digits: Read Digits	10	10	0.72	0.009
Alphabetic Letters: Read Alphabetic Letters	26	26	5	1.6
Resource Management: Read Sentences (Word-Pair Grammar)	1000	60	3.6	0.1
Resource Management: Read Sentences (Null Grammar)	10000	1000	17	2
Wall Street Journal: Read Sentences	5000	45	7.2	0.9
North American Business News: Read Sentences	Unlimited	160	6.6	0.4
Switchboard: Spontaneous Telephone Conversations	2000-Unlimited	80-150	38.5-43	4

1.3.1 The model enforcement approach to acoustic feature design

An important goal for designers of ASR systems is to achieve a high level of performance while minimizing the number of parameters used by the system, not only because a large number of parameters increases the computational load and the storage requirements, but also because it increases the size of the training data required to estimate the parameters. One way of controlling the number of parameters is to adjust the structure of the conditional joint PDF used by the recognizer. For example, the dimensionality of the acoustic feature vectors in HMM-based ASR systems is too large for their Gaussian conditional joint PDFs to have full covariance matrices. On the other hand, approximating the conditional PDF by a diagonal covariance matrix Gaussian PDF is equivalent to assuming that the elements of

the acoustic feature vector are statistically independent given the HMM state. The use of mixture of Gaussians relaxes this assumption, as it can be considered as a way of modeling the correlations implicitly. However, the mixture of Gaussian components can model discrete sources of variability like speaker variations, gender variations, or local dialect, but cannot model continuous types of variability that account for correlation between the elements of the feature vector. Examples of these continuous sources are coarticulation effects and background noise. Clearly, modeling both continuous and discrete types of variability is important to obtain good models of the speech signal. We present in this dissertation a unified information-theoretic approach to the problem of designing features that satisfy a given joint PDF model. We call this problem the model enforcement problem. We describe the conditions under which the model enforcement approach can be reduced to maximum likelihood estimation problem. We describe also the relation between this approach and maximizing the conditional mutual information between the classes and the features given the HMM model. We describe iterative algorithms to calculate nonlinear maps of the original features to new features that satisfy better the model assumptions using either maximum likelihood estimation or maximum conditional mutual information estimation. We provide also a generalization of these algorithms to calculate class-dependent features.

1.3.2 Class-dependent acoustic feature design

In classification and recognition problems with many classes, it is commonly the case that different classes exhibit wildly different properties. In this case it is unreasonable to expect to be able to summarize these properties by using features designed to represent all the classes. In contrast, features should be designed to represent subsets that exhibit common properties without regard to any class outside this subset. The value of these features for classes outside the subset may be meaningless, or simply undefined.

The class-dependent features can be looked at as a method of dimensionality reduction in classification [7], [8]. Unlike other methods of dimensional reduction, it can be defined in terms of sufficient statistics and in such a way that result in no theoretical loss of performance. There are two conflicting sources of loss of information necessary for classification and recognition. The first is due to reducing the given data to a set of features, and the second is due to approximating the true joint PDFs of the features. The former loss decreases as the dimensionality of the features increases, while the latter increases as the dimensionality of the features increases. Class-dependent features avoid this compromise by allowing more information to be kept for a given maximum feature dimension. This is clearly at the expense of increasing the computational requirements of the system.

Phoneme-dependent feature design for phoneme classification is a well-studied approach that is motivated by the fact that different phonemes have different salient characteristics that may require different features. Using phoneme-dependent features not only simplifies the features design problem but also allows the overall system to benefit from the ability of these streams to reveal discriminant information of the speech signal. However, using multiple observations in statistical speech recognition systems was not a possible attractive choice one or two decades ago. The two main reasons for this fact are the computational complexity and storage requirements associated with this approach, and the lack of a rigorous formulation of how the use of these class-dependent features will integrate with the current statistical approach for speech recognition.

Computational power has doubled roughly every 18 months since the late 1960s, and this trend is expected to continue for more than 10 years. This increase in computing power makes it feasible to move beyond the simple acoustic representation of speech in current recognition systems. Many recent speech recognition systems combine multiple speech recognizers to achieve more robustness and better performance. Using different feature streams within each recognizer allows the overall system to benefit from the ability of these streams to reveal complementary information of the original speech signal. Combination of multiple

recognizers is consistently reported to outperform baseline systems. In [9], for example, a hybrid speech recognition system based on the combination of acoustic and articulatory information achieved better word recognition results than the baseline systems. Choices of the level of the combination and the best feature streams to be combined together remain as main issues for successful combination. These choices are currently made through intuition and empirical comparison.

The mathematical formulation for phoneme-dependent features in the weak sense within the same statistical speech recognition systems was studied in [10], and recently in [11]. In the weak sense, features have observable values for all classes, but the features and some class variables are conditionally independent given a set of classes [12]. This increases the computational and the storage requirements of the system, and results in the introduction of meaningless models that degrade the performance of the recognizer. Features are said to be class-dependent in the strong sense if they are assumed to be observable only for one class or cluster of classes but undefined for the rest of the classes [12]. We will use here the notion of class-dependent features for ASR to represent using different features for different phonemes or different clusters of phonemes that are constructed using some criterion. In Chapter 2, examples of previous attempts to use class-dependent features in the weak sense for speech recognition will be provided. The important requirement to overcome problems in previous formulations is to provide a mathematical framework describing the efficient integration of class-dependent features in the strong sense with current speech recognizers. One of the main goals of this dissertation is to provide a mathematical formulation and a practical solution to this issue. In Chapter 4, two methods for class-dependent feature design for pattern recognition are suggested: one to optimize a discriminative criterion, and the other to maximize the likelihood of the training data. These methods are applied directly to the hidden Markov model speech recognizer and require neither hierarchical and voting schemes nor antiphoneme models that are usually used in previous multiple-observation systems. Experiments presented in Chapter 4 show that systems trained using a discriminative approach

to develop class-dependent features outperform those trained using a maximum likelihood criterion. This discriminative criterion can be achieved by minimizing an estimate of the recognition error or maximizing an estimate of the conditional mutual information between the class identity and the features.

1.3.3 The discriminative approach to acoustic feature design

There are two main categories of current discriminative approaches to the ASR problem. The first is the model-based approach. In this approach, discriminative training algorithms are used to estimate the parameters of the model. Important examples of these algorithms are maximum mutual information estimation (MMIE) algorithms [13] based on an extended Baum-Welch algorithm [14], and minimum classification error (MCE) algorithms based on the generalized probabilistic decent (GPD) algorithm [15]. Clearly, the discrimination power of the models generated using these approaches is limited by the discrimination power of the features used. The second category takes advantage of this fact and therefore is feature-based. Most of the algorithms that belong to this category are variants or extensions of linear discriminant analysis (LDA) [16]. These approaches maximize the likelihood of Gaussian or mixture of Gaussians models of the joint class-conditional PDFs of the feature vector in the original feature space. The main advantage of these approaches is their computational efficiency, but the results reported on their effect on the ASR system performance do not show consistent improvement. After this brief introduction, we can emphasize that both categories have severe limitations on their performance. Model-based approaches are limited by the discrimination power of the features used, and feature-based approaches are limited by being restricted to optimizing a nondiscriminative objective—namely likelihood.

In this dissertation, we first introduce a feature selection algorithm based on MMIE [17], [18], and then describe a maximum conditional mutual information linear projection algorithm that is based on a novel interpretation of LDA [19]. The main advantage of these approaches

is that they maximize a discriminative criterion in the projected feature space instead of maximizing the likelihood in the original feature space as in most previous feature-based methods. In [17] and [18], maximization of mutual information between acoustic features and phoneme identity or phonological features was used to select the acoustic features for speech recognition. In both cases the improvement achieved by using this criterion was marginal. This can be mainly attributed to the suboptimality of the algorithms used in sequential feature selection. In [20], a linear transform is optimized to maximize the conditional mutual information between acoustic features and phoneme identity given Gaussian mixture models in the original feature space. No improvement in word error rate is achieved compared to the baseline system. This can be attributed to the inefficiency of the approximate joint probabilistic models in the high-dimensional original feature space. We suggested a method, [19], to calculate the conditional mutual information given a set of probabilistic models in the low-dimensional projected feature space. We achieved significant improvement in phoneme recognition accuracy using this approach over current LDA-based approaches. This approach will be presented in Chapter 5. Compared to LDA-based approaches, this proposed feature-based approach has three main advantages: optimizing a discriminative criterion, the solution is based on modeling the joint conditional PDF of the features in the lower-dimensional space, and using the model of these PDFs that are used by the recognizer. The goal of training both the map parameters and the model parameters is to improve the recognition accuracy, and both of them can be trained using the same discriminative training algorithm. We devise methods for joint optimization of both types of parameters. These methods are extensions to the existing MMI extension to the Baum-Welch algorithm [14], and the MCE/GPD algorithm [15]. This joint optimization should, in principle, lead to better performance, but at the expense of an increase in the computational complexity of the algorithm.

1.4 Goals and Accomplishments

This dissertation demonstrates how to optimize the acoustic features in statistical ASR systems to satisfy the recognizer assumptions, and to increase its ability to discriminate among phonemes. It demonstrates also how these optimizations can provide possible solutions to some of the problems of current speech recognition systems. In this demonstration, we have both theoretical and practical goals.

The first theoretical goal is to mathematically formulate the problem of model enforcement and prove that it can be reduced to the problem of maximum likelihood estimation of the parameters of a volume-preserving map of the features. We show also that optimizing a map of the features to maximize the conditional mutual information given the recognizer model is a special case of the model enforcement framework that we present. This equivalence takes place when we try to improve the validity of our estimate of the a *posteriori* probability using the recognizer's probabilistic model. This goal requires definitions of the criteria that can be used in optimizing the acoustic features to better satisfy the recognizer's assumptions in estimating the likelihood or the a *posteriori* probability. It also requires definitions of the empirical estimate of these criteria. Once we have empirical estimates of these criteria, optimization algorithms can be described to design the acoustic features. The second theoretical goal is to provide as general a mathematical formulation as possible of the problem of strong-sense phoneme-dependent feature design, or more generally strong-sense cluster-of-phonemes-dependent features that are optimized for discrimination or model enforcement. In this dissertation, we introduce a class-dependent acoustic feature design approach that can be integrated directly with any probabilistic model. This approach avoids the need of having a conditional probabilistic model for each class and feature type pair. This decreases the computational and storage requirements of speech recognizers based on heterogeneous features. The third theoretical goal is to provide a new interpretation of LDA based on a discriminative criterion to allow discriminative generalizations of LDA.

The first practical goal is to provide iterative algorithms that solve these optimization problems efficiently and to describe practical considerations that can decrease the computational complexity of these algorithms. The second practical goal is to describe iterative algorithms to jointly optimize the parameters of the feature extraction module and the parameters of the recognizer using MLE, MMIE, and MCE. The third practical goal is to test the approaches described in this dissertation and investigate the significance of the improvement in recognition accuracy compared to speech recognizers using standard techniques.

The main accomplishment of this dissertation is the introduction of the unified feature transformation framework for classification and recognition to satisfy a given probabilistic model. Not only does this formulation explain the relation between many popular techniques for data analysis and feature transformation in various disciplines like principal component analysis (PCA), independent component analysis (ICA), and maximum likelihood linear transform (MLLT), but it also allows the extension of these approaches to not-necessarily-linear feature transforms. Motivated by computational efficiency, we described a nonlinear volume-preserving features transform based on this framework. However, as the computational capabilities of computers increase with time, many other problem-dependent feature transforms can be designed using our framework. An important accomplishment of this dissertation is describing LDA, the popular technique for dimensionality reduction in classification and recognition, as a special case of maximizing the conditional mutual information between the features vector and the class identity given the classifier's probabilistic model. By relaxing the assumptions needed for the equivalence of the two approaches, we achieve several possible discriminative generalizations of LDA for dimensionality reduction.

1.5 Organization of the Dissertation

In this section, we review briefly the organization of this dissertation. This review may help readers with different backgrounds to focus on different parts. The dissertation is divided

into six chapters. Chapter 1 states the problems to be solved and discusses their importance. It provides also motivations for the approaches presented in the dissertation.

Chapter 2 gives a brief survey of previous work related to our approaches. It starts with a very brief introduction to the history of statistical ASR systems, and the current most widely used features for ASR systems in Sections 2.1 and 2.2, respectively. In Section 2.3, a brief review of previous transformations for approximate normality is provided. Then, we focus on previous work on developing features that can be better modeled using mixture of Gaussians with diagonal covariance matrices in Section 2.4. We also describe the most important ASR systems that used multiple observations in Section 2.5. Finally, discriminant analysis approaches to feature design for ASR systems are provided in Section 2.6.

In Chapter 3, we start with a brief overview of the parametric approach for statistical classification and recognition in Section 3.1. A summary of limitations of previous approaches is given in Section 3.2. In Section 3.3, the unified feature transformation framework to decrease the mismatch between the joint PDF of the features and its model used by the recognizer is introduced. Then we provide some approaches based on this framework. First, a theoretical formulation of the nonlinear independent component analysis approach is introduced in Section 3.4 to solve the problem due to the diagonal-covariance assumption in ASR systems. An algorithm that uses this formulation for speech processing is described also in Section 3.4. A new formulation of the same approach based on MLE is introduced in Section 3.5. An application of the approach to large-vocabulary speech recognition is introduced in Section 3.6. The choice of using global or class-dependent maps is discussed in Section 3.7.

Chapter 4 discusses the strong-sense class-dependent features approach. It starts with a brief introduction to previous approaches to class-dependent features in ASR in Section 4.1. In Section 4.2, the problem of using strong-sense class-dependent features in statistical classification or recognition systems is formulated. A maximum likelihood approach is described in Section 4.3. Finally, discriminative strong-sense class-dependent features design is introduced in Section 4.4.

Chapter 5 discusses discriminative dimensionality reduction and feature selection techniques. It starts with a description of an algorithm for feature selection based on MMI in Section 5.1. This algorithm is applied to features selection for phoneme recognition in Section 5.2. Then, an interpretation of LDA using a constrained maximum conditional mutual information projection is provided in Section 5.3. In Section 5.4, implementation of the maximum conditional mutual information projection approach is described. Finally, a summary and future work directions are described in Chapter 6.

In this dissertation, a superscript is used as an index of a realization of the random vector. Capital letters are used to denote the random variables and the corresponding small letters to denote their realizations. Both vectors and matrices are in boldface to be distinguished from scalars.

CHAPTER 2

BRIEF OVERVIEW OF RELATED WORK

This chapter provides examples of the most important research related to the goals of this dissertation. It starts with a glimpse of the history of ASR systems and a quick introduction to the most important acoustic features used in current ASR systems. Then, we provide examples of previous work in statistical analysis on using feature transformation to generate features that satisfy approximately the normality assumption in Section 2.3. In Section 2.4, important previous solutions to the problem of using diagonal-covariance Gaussian mixture conditional PDFs in ASR systems are described. In Section 2.4, we show how the most significant previous attempts to use class-dependent observations in ASR systems were formulated. Finally, we describe important approaches to dimensionality reduction in feature extraction for ASR systems in Section 2.5.

2.1 Statistical Approaches to ASR Modeling

Early ASR systems were inspired by advances in artificial intelligence (AI) [2]. These systems relied on sets of rules for acoustic-phonetic modeling and language modeling. These systems were knowledge-based systems that used the experiences and knowledge of spectrogram readers, and psychoacoustics. They worked reasonably well for small tasks under controlled environments. The performance of such systems was found to be fragile [2]. Then, stochastic approaches were introduced to both the acoustic-phonetic and language modeling. These stochastic approaches brought the rich mathematical basis that was available in statistical

pattern recognition literature to ASR [21]. In current ASR systems, the acoustic-phonetic modeling is mainly based on hidden Markov models (HMM) [22], or hybrid systems like artificial neural networks and HMM (ANN/HMM) [23], and the language models are in the form of N-grams which are trained using a large text corpus [24]. Stochastic techniques typically use minimal *a priori* assumptions about the nature of the problem. They estimate the parameters of the model directly from the data. This statistical approach improved the quality of ASR systems significantly and extended their applications to new areas. However, applying statistical approaches to the feature extraction module was very limited and did not have the same tremendous impact, as will be discussed in the next section.

2.2 Current Feature Extraction Module in ASR Systems

Most acoustic features that have been successful in speech recognition try to model the speech signal as the convolution of the excitation signal and the vocal tract transfer function, and try to extract the vocal tract transfer function characteristics by linear predictive coding or homomorphic signal processing. There are a large number of studies in the literature which describe and compare various feature extraction algorithms for speech recognition; [25]–[29] are just a few examples.

Over the past few decades, many variants of filter banks, LPC, and cepstral vectors have been used for speech recognition. The majority of the systems have converged to the use of cepstral vectors derived from a filter bank that has been designed according to some model of the auditory system. This model takes into consideration that human speech perception follows a nonlinear frequency scale named the Mel scale, and that the perception of a certain frequency component is affected only by the presence of energy in neighboring frequencies within what is called the critical band. Therefore, a filter bank with center frequencies that are chosen according to the Mel scale is used. Psychoacoustic experiments using simultaneous frequency masking have revealed that the bandwidth of the critical bands increases with

the center frequency. Therefore, the filter's bandwidth is increased as its center frequency increases.

In many speech parameterization schemes, such as filter bank data obtained by sampling the short-time Fourier spectrum (STFS), nearby frequencies within the same observation vector are highly correlated. This is inconsistent with using diagonal-covariance Gaussian mixtures in HMM speech recognizers. To decrease this correlation, cepstral coefficients obtained by taking the inverse Fourier transform of the log of the Fourier transform of the data are used instead of straight filter bank data. This decreases the correlation between the individual parameters of a single observation frame, fitting the data more closely to the diagonal covariance assumption. It was proved that for a first-order Gaussian-Markov random process, the DCT transform approximates the KLT transform [1]. As described in Chapter 1, the delta and delta-delta coefficients are usually appended to the cepstrum coefficients to account for temporal correlation among frames of speech. Despite the widespread use of the cepstrum coefficients as the features for speech recognition, the method used to get these features is completely heuristic. Also, there is no reason to believe that delta and delta-delta coefficients added to the cepstrum coefficients are the optimal features to model the temporal correlation of speech frames.

2.3 Transformations for Approximate Normality

Many important results in statistical analysis and pattern recognition follow from the assumption that the population being sampled or investigated is normally distributed. The assumption of normality is seriously violated in many interesting problems. A frequently discussed solution in the statistical literature is to transform the original measurements to features that better satisfy the normality assumption. In this section, we will give very brief examples of previous approaches to transform multivariate data such that this assumption is better satisfied.

The transformation may be based on theoretical considerations or use a data-driven approach. Univariate examples of the former type are the logistic transformation for binary data [30] and the variance stabilizing transformations for the binomial, the Poisson, and the correlation coefficient [31].

There are many examples of data-driven transformations. Tukey introduced a family of power transformations such that the transformed values are a monotonic function of the observations over some admissible range for univariate analysis [32]. This family was modified in [33], where maximum likelihood and Bayesian methods were used to estimate the transformation parameter. These power transforms were extended to the multivariate case by using a number of scalar transforms equal to the dimension of the observation vector in [34]. Conceptual and computational simplicity were the main reasons to limit the suggested transforms to a family of power transforms.

2.4 Transformations for Redundancy Reduction

As described in Chapter 1, one way of controlling the number of parameters is to adjust the structure of the covariance matrices used by the recognizer. Traditionally, the choice is made between either diagonal or full covariance matrices. Full covariance is an impractical choice in many applications, and diagonal covariance degrades the performance of the recognizer [35], as the acoustic features used in ASR systems are not decorrelated. Recent approaches to this problem that offer new alternatives can be classified into two major categories. The first category tries to decrease the number of parameters required for full covariance matrices by tying the parameters of the unitary eigenvectors matrix that can map any covariance matrix to a diagonal matrix, using the fact that the covariance matrix is a symmetric nonnegative definite matrix that can be diagonalized using a unitary transform. In other words, this category tries to reduce the redundancy in the model parameters. The second category chooses to decorrelate the features or map them to approximately independent features that

can be modeled by a diagonal covariance. In other words, this category tries to decrease the redundancy in the features themselves. In this section, we will describe both approaches and give several examples of each approach.

2.4.1 Redundancy reduction of the model parameters

There are a variety of choices for covariance structure other than diagonal or full. Two examples that can be used in ASR systems are block-diagonal [36] and banded-diagonal matrices. Another method often used by ASR systems is tying, where certain parameters are shared among a number of different models. Accordingly, various matrix decomposition methods of the form $\mathbf{C} = \mathbf{A}^T\mathbf{D}\mathbf{A}$, where $\mathbf{D}$ is a diagonal matrix and $\mathbf{A}$ is a unitary matrix, have been applied to covariance matrices along with different styles of partial parameter tying [37].

2.4.1.1 Semitied covariance matrices

In [37], the semitied covariance matrices approach is introduced. It estimates a transform in a maximum likelihood fashion given the current model parameters. This optimization is performed using an iterative scheme that is guaranteed to increase the likelihood of the training data. An iterative algorithm based on the expectation-maximization algorithm that calculates a linear transform that diagonalizes the covariance matrix of the Gaussian components of each state is provided. Instead of having a covariance matrix for every component in the recognizer, each covariance matrix consists of two elements, a component-specific diagonal covariance element, $\mathbf{\Sigma}_{diag}^{(m)}$, and a semitied class-dependent nondiagonal matrix $\mathbf{H}^{(r)}$, such that

$$\mathbf{\Sigma}^{(m)} = \mathbf{H}^{(r)}\mathbf{\Sigma}_{diag}^{(m)}\mathbf{H}^{(r)T},$$

where m is the Gaussian component index and r is the state index. It was shown that the word error rate decreased from 9.2% to 8.12% on the 1994 ARPA Hub1 Database [37].

The iterative algorithm used in the semitied covariance matrices approach required solving a nonlinear optimization problem for each iteration. In [38], a solution of this problem is provided by enforcing the matrix $\mathbf{H}^{(r)}$ to be a unitary upper-triangular matrix.

2.4.1.2 Factor analysis approach

Factor analysis uses a small number of parameters to model the data in a high-dimensional space. It is a linear Gaussian model that assumes the observed feature vector is related to a set of independent latent variables (factors) by linear transformation, with additive independent white Gaussian noise added to the output of the transformation. It was used in [39] to model the covariance matrix of each Gaussian component of the Gaussian mixture used within each state of the HMM recognizer. The parameters of the factor analysis model were derived using an expectation-maximization to maximize the likelihood and using gradient based method to minimize an empirical estimate of the recognition error as in [40]. We will discuss it in more detail in Chapter 3.

2.4.2 Redundancy reduction of the features

In these approaches, the original feature space is transformed to a new feature space that satisfies the diagonal-covariance models better. This is achieved by optimizing the transform based on a criterion that measures the validity of the assumption. All these methods used linear transforms and tried to solve the problem due to the diagonal covariance assumption only. In Chapter 3, we will introduce a framework that allows using a nonlinear transformation and can deal with any assumptions by the probabilistic models.

2.4.2.1 Principal component analysis

Principal component analysis [16] and the closely related Karhunen-Loève transform are classic techniques in statistical data analysis, feature extraction, and data compression. Given a

random vector $\mathbf{X}$ and a number of observations from this random vector, no explicit assumptions on the probability density of the vectors are made in PCA, as long as the first- and second-order statistics can be estimated from the observed data. Also, no generative model is assumed for the vector $\mathbf{X}$, but there are extensions to PCA like probabilistic principal component analysis (PPCA) [41] and the approach in [42] that associate a generative model with PCA. In the PCA transform, the vector $\mathbf{x}$ of length n is first centered by subtracting its mean. Next, $\mathbf{x}$ is linearly transformed to another vector $\mathbf{y}$ with m elements, $m < n$, so that the redundancy induced by correlation is removed. This is done by finding a rotated orthogonal coordinate system such that the elements of $\mathbf{x}$ in the new coordinate system become uncorrelated. The vector is projected in this new coordinate system to the subspace that consists of the directions along which the vector has maximum variance. The transform is constructed from the eigenvectors of the sample covariance matrix with maximum corresponding eigenvalues. This transform is the unique unitary transform of dimension m such that the elements of $\mathbf{y}$ are uncorrelated and the variance of $\mathbf{y}$ is maximized. PCA is a linear technique, so computing $\mathbf{y}$ from $\mathbf{x}$ is not computationally expensive, which makes real-time processing possible. Since there are many sources of variability in speech features and some of them are irrelevant to linguistic information, selecting the direction of maximum variance for projection does not always minimize the recognition error [43]. Therefore, PCA was mainly used in speech parameterization to calculate the principal components of the Fisher covariance matrix of the classes corresponding to the speech units [44], [45],

$$\mathbf{S}_{wb} \quad = \quad \mathbf{W}^{-1}\mathbf{B}, \tag{2.1}$$

where $\mathbf{W}$ is the matrix of the mean of the within-class variance, and $\mathbf{B}$ is the matrix of the variance of the means of the classes [16].

A state-specific rotation approach was introduced in [35]. It calculates the full covariance matrix for each state in the system. All data from that state is then decorrelated using the

eigenvectors matrix corresponding to the estimated covariance matrix. Multiple diagonal covariance matrix Gaussian components are then trained for each state. In other words, it is a state-specific PCA of the acoustic features.

2.4.2.2 Independent component analysis

ICA defines a generative model for the observed multivariate data [46], [47]. This data is typically given as a large database of samples. In the model, the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed non-Gaussian and mutually independent, and they are called the independent components of the observed data. These independent components, also called sources or factors, can be found by ICA.

ICA can be seen as an extension to principal component analysis and factor analysis. ICA is a much more powerful technique, however, capable of finding the underlying factors or sources when the assumptions assumed by these classic methods are not valid.

The goal of ICA is to estimate the independent sources and the mixing coefficients given only observations that are a linear mixture of the latent independent source signals. In contrast to PCA, ICA not only decorrelates the sources but also reduces higher-order statistical dependencies, attempting to make the components as independent as possible.

The data analyzed by ICA could originate from many different kinds of application fields, including digital images and document databases, as well as economic indicators and psychometric measurements. In many cases, the measurements are given as a set of parallel signals or time series; the term blind source separation is used to characterize this problem. Typical examples are mixtures of simultaneous speech signals that have been picked up by several microphones, brain waves recorded by multiple sensors, interfering radio signals arriving at a mobile phone, or parallel time series obtained from some industrial process.

There are many approaches to solving the ICA problem, including information maximization approach, maximum likelihood estimation, negentropy maximization, higher-order

moments and cumulants approximations of differential entropy, and nonlinear PCA. In [48], it is shown that all these different approaches lead to the same iterative learning algorithm.

ICA has been used in speech recognition applications when there is a background auditory source other than the speaker, [49], [50]. It was used also in developing features for speaker recognition [51] and speech recognition [52]–[54]. Factor analysis also was used to model the covariance matrix of the Gaussian mixtures of HMM recognizers in [39].

2.4.2.3 Maximum likelihood linear transformation

The maximum likelihood linear transform (MLLT) was introduced in [55]. It is based on the idea that the diagonal covariance models impose a constraint on the likelihood of the features which results in underestimating its value, and by trying to maximize the value of the likelihood by the introduction of a linear transformation of the data, we will get features that are better represented by the model. In [56], heteroscedastic discriminant analysis (HDA) made no improvement in word recognition, but made a significant improvement when used in combination with MLLT. We will describe the HDA transform in Section 2.5. In [57], MLLT was used also after HDA and improved the word recognition error by 10-15% relative to the original results. The nonlinear independent component analysis introduced in Chapter 3 can be shown as a generalization of the MLLT to nonlinear transforms. This is due to the fact that the empirical estimate of the objective function to be minimized in our work is actually the negative of the empirical estimate of the likelihood in MLLT approach. The proof will be provided in Chapter 3 and is based on the equivalence of minimizing the mutual information of the output components and maximizing the likelihood of the outputs under the statistical independence assumption.

2.5 ASR Systems with Class-Dependent Features

The approach of using class-dependent features in speech recognition and verification has been suggested many times before [58]. Its main problem, due to the statistical nature of the recognizer, was how to compare *a posteriori* probabilities conditioned on different sets of features to decode a given utterance. Actually, the number of papers that ignored this problem and used class-dependent features with statistical recognizers is really surprising [59]–[63]. However, there were many suggestions recently to solve this problem. The approaches can be classified as model-based approaches and feature-based approaches. In model based approaches, the problem was solved by completely abandoning the statistical structure of the recognizer, or by adding extra reference models that have no physical meaning but are used to normalize the likelihoods to be comparable statistically. The feature-based approach restricted the class-dependent features to features generated by class-dependent linear transforms from an original set of features. In the following, we introduce brief examples of both approaches.

2.5.1 Model-based approaches

To avoid the problem of having to compare likelihoods based on different observation spaces, many researchers [58] suggested using hierarchal approaches for recognition or verification. The author of this dissertation [64], for example, proposed clustering the phonemes of the language to a certain number of clusters, and then building HMM models for these clusters. These models are then used in an HMM-based verification system to verify that the correct sequence of clusters was pronounced. If the utterance passes this test, another test verifies that the correct phoneme of each cluster was pronounced based on a cluster-specific set of features. However, hierarchal systems added complexity to ASR systems, and their performance was worse than purely statistical approaches like HMM-based systems. They also have the implicit unjustified assumption that features at a certain level are independent of

classes and features at lower levels given the value of the cluster at this level. Other systems just used likelihoods based on different observations, ignoring that these likelihoods cannot be compared together [59]–[63]. This problem was addressed for segmental ASR systems that use different set of features for different segments in [10]. This class of recognizers processes the speech frames to produce a segment-based network and represent each segment by fixed-dimensional features. In such a feature-based recognizer the observation space takes the form of a temporal network of feature vectors, so that a single segmentation of an utterance will use a subset of all possible feature vectors. This approach was motivated by the ability to incorporate knowledge of the speech signal by using these different sets of features. This approach was used in the SUMMIT speech recognizer developed by the Spoken Language System Group at MIT [62]. In [10], a probabilistic framework for this recognizer was provided. This framework was motivated by the need to account for incomplete knowledge in the system. This probabilistic framework will be discussed in Chapter 4. It introduced the notion of antiphone to model all the segmentations of the utterance that compete with the segmentation corresponding to the phonetic units. This results in replacing the likelihood estimation by estimating the likelihood ratio of the features given the phone and the antiphone, respectively. The main problem with this approach is how to train the antiphone models. They are synthetic entities that have no physical meaning at all, so there have been a variety of suggestions to train these models. They range from taking all other phones in the phone set to train the antiphone model to taking a very small set of similar phones in the phone set. By the introduction of these synthetic models, statistical ASR systems can—at least theoretically—deal with class-dependent features in the weak sense described before. Many discriminative training algorithms [65] for HMM parameters for speech recognition used the likelihood ratio of the sequence of phones and their corresponding antiphone as the objective function to be maximized instead of traditional maximum likelihood approach [66]. In [67], an extension of the Baum-Welch algorithm to class-dependent features was introduced. The development was based on a statistical hypothesis testing approach. However,

it replaces the antiphone models requirement by a noise-only model. Then, the likelihood is replaced by the likelihood ratio of the phoneme sequence and the noise-only model. The relation between maximizing the new likelihood function, and the original likelihood function is provided there, but this framework has not solved the main problems of the previous methods which is the definition of this noise-only model, and how well we can approximate the PDF of these class-dependent features under this noise-only hypothesis.

2.5.2 Feature-based approaches

The relation between the likelihood of the features and the likelihood of the new features generated by class-dependent linear transformation of the original features was described in [55]. The class-dependent linear transformations were estimated by maximizing the likelihood of the original features. In [11], class-dependent subspace projection of the features for ASR systems was suggested. The problem of likelihoods based on different projections was approached by ensuring that all the feature transforms span the same original feature space. This was achieved by defining clusters of classes and defining a common PDF shared by all members of the class over the complement of the projection subspace.

2.6 Dimensionality Reduction Techniques

To achieve high recognition accuracy, the feature extractor is required to capture salient characteristics suited for discriminating among different classes. Therefore, the linear discriminant analysis (LDA) approach was borrowed from statistical pattern recognition techniques [16] and applied to ASR systems. Recently, discriminant analysis approaches are generalized to more powerful techniques and tailored specifically to the ASR problem. Although using a specific discriminant analysis technique for feature extraction in ASR systems is not an agreed-upon issue now, there are a growing number of ASR systems that use dis-

criminant analysis in their feature extraction module. In this section, we will introduce briefly the most important techniques.

2.6.1 Linear discriminant analysis

The linear discriminant analysis (LDA) technique tries to improve the separability of the classes by finding the linear transform that maximizes the ratio of the determinant of between-class covariance and the determinant of the average within-class covariance [16]. Given a set of N independent observation vectors $\{\mathbf{x}_i\}_{1\leq i\leq N}, \mathbf{x}_i \in \Re^N$, each of them belongs to only one class $j \in 1, \cdots, J$. Let each class j be characterized by its sample mean μ_j, sample covariance matrix $\mathbf{\Sigma}_j$, and observation count N_j. The within-class scatter is given by

$$\mathbf{W} = \frac{1}{N}\sum_{j=1}^{J} N_j \mathbf{\Sigma}_j, \tag{2.2}$$

and the between-class scatter is given by

$$\mathbf{B} = \frac{1}{N}\sum_{j=1}^{J} N_j \mu_j \mu_j^T - \mu\mu^T, \tag{2.3}$$

where μ is the global mean of the observations. The goal of LDA is to find a linear transformation characterized by the matrix θ such that

$$J(\theta) = \frac{\left|\theta \mathbf{B} \theta^T\right|}{\left|\theta \mathbf{W} \theta^T\right|} \tag{2.4}$$

is maximized. The maximization can be formulated as principal component analysis of the Fisher covariance matrix or as a maximum likelihood estimation problem [68].

To obtain features suitable for syllable classification, Hunt proposed the use of linear dis-

criminant analysis (LDA) to derive features that improve the separability of the models of syllables [69]. Brown, almost a decade latter, experimented with both principal component analysis (PCA) [16] and LDA to project the features in subspaces of reduced dimensions [70]. His experiments showed that the LDA transform is superior to the PCA transform. He incorporated context information by applying LDA on an augmented feature vector formed by concatenating the features from a number of frames around the observation vector. By doing so, context is incorporated selectively based on the best linear combination of observation vectors, and thus all the components of the feature vector are likely to contribute to better classification. LDA has been employed successfully to reduce the feature dimensions from high-dimensional acoustic representations for speech recognition [29]. In [45], using PCA analysis had no effect on the phoneme classification task on OGI numbers database. LDA improved the phoneme classification on the same task by 0.7%. Since the LDA solution is insensitive to any nonsingular linear transforms before it, many researchers suggested removing the discrete cosine transform from the calculation of the cepstral coefficients and replacing it with the LDA transform [56], [71]. However, this replacement gave no improvement in word error rate in [71]. On the other hand, removing the Mel-scale filter bank stage from the feature extraction module gave 1% improvement in the word error rate, when LDA and a maximum likelihood diagonalization transform were used. LDA has been applied to discrete [72] and continuous [45] HMM speech recognition systems. Applying LDA to mixture of Gaussians HMM is more complicated than the discrete density HMM as there are many choices of the sample class assignment. Various techniques for class assignment have been proposed and used with different degrees of success with continuous density HMMs [73]–[75]. Adaptive forms of LDA have also been proposed with encouraging results, taking into account mismatch between the assumed class distributions and the actual data [76]. Despite its popularity and promise for significant improvements to speech recognition, LDA has not always improved the performance of speech recognition systems. This is due to lack of robustness in the widely used model-free formulation of LDA. In the original Fisher-Rao

model-free formulation [16], LDA projections are best suited to classifier models where class distributions have equal variance. LDA's assumption that all the within-class covariance matrices are approximately the same, makes it inappropriate for problems of unequal covariance classes like speech recognition. It has improved the performance on small vocabulary tasks, but the results were not conclusive for large vocabulary phoneme-based systems [77], [78].

Campbell [79] has shown that linear discriminant analysis is related to the maximum likelihood estimation of parameters for a Gaussian model, with *a priori* assumptions on the structure of the model. The first assumption is that all the class discrimination information resides in a p-dimensional subspace of the n-dimensional feature space where the LDA mapping is represented by $p \times n$ matrix. The second assumption is that the within-class variances are equal for all classes. Hastie and Tibshirani [80] further generalized this result by assuming that class distributions are a mixture of Gaussians. However, the constraint of common covariance matrices is maintained in both [79] and [80]. Kumar [68] generalized LDA to the case of classes of different covariance matrices and referred to this generalization as heteroscedastic discriminant analysis (HDA).

2.6.2 Heteroscedastic discriminant analysis

Heteroscedastic discriminant analysis (HDA) is an extension to LDA that removes the equal covariance constraint [68]. HDA was first formulated as a maximum likelihood estimation problem for normal populations with common covariance matrix in the rejected subspace. An alternative interpretation of HDA as a constrained maximum likelihood projection for a full-covariance Gaussian model is introduced in [56]. It maximizes the objective function

$$J(\theta) = \frac{\left|\theta \mathbf{B} \theta^T\right|^N}{\prod_{j=1}^{J} \left|\theta \mathbf{\Sigma}_j \theta^T\right|^{N_j}}. \tag{2.5}$$

In [56], HDA made no improvement in word recognition, but made a significant improvement when used in combination with MLLT. In [57], HDA combined with MLLT were reported to improve the performance over MFCC by 10-15% relative on large vocabulary conversational speech tasks using Voicemail and Switchboard databases. It was noted in [11] that MLLT is a special case of HDA when the dimension of the generated feature vector equals the dimension of the original feature vector.

2.6.3 Information-theoretic approaches

Extracting linguistic information related to speech recognition based on a given probabilistic model can be achieved by using information-theoretic measures like mutual information between the classes and the feature vector as the criterion to be optimized by the features.

An approach for selecting the level of the combination of several speech recognizers based on conditional mutual information of the feature streams given the underlying phoneme identity was suggested in [81]. In [82], the mutual information was used to estimate the distribution of partial phonetic information in the time-frequency plane relative to acoustic landmarks. A framework for defining the theoretically optimal method for feature subset selection was presented in [83]. It proves that for a feature to be unnecessary to model a certain property, it should have a Markov blanket within the complete feature set. However, this optimal feature selection approach is computationally intractable. In [17], the speech signal is modeled as a combination of independent phonological factors. These phonological factors are represented by the best fixed-length subset of the available acoustic feature space. An algorithm that calculates a good approximation of the acoustic features subset that has the maximum mutual information with each phonological factor is presented in [17]. The algorithm was applied to maximize the mutual information of the feature vector with the phoneme identity in [18].

An expression of the mutual information between the features and the class identity was used to learn a discriminative linear feature transform in [20]. It was based on Renyi entropy and nonparametric Parzen estimates of the the conditional PDFs of the features. The use of Renyi entropy was motivated by computational efficiency. An implementation that assumes a Gaussian mixture model of the PDFs in the original feature space was also introduced. No improvement in the word error rate on the AURORA2 task [84] is achieved by using this approach compared to the baseline system.

CHAPTER 3

THE MODEL ENFORCEMENT APPROACH

Given a set of realizations of a random vector and a hypothesized model of its probability density function, the purpose of this work is to find a transform of this random vector and a set of model parameters that jointly minimize an empirical estimate of the relative entropy between its true probability density function and the hypothesized model. The first stage in many pattern recognition and coding tasks is to generate a good set of features from the observed data. The set should be compact and capture all class discriminating information. This set of features is usually chosen based on the available knowledge about the problem, or based on data-driven approaches to achieve compactness and discrimination goals. In both cases, the features also should satisfy the assumptions imposed on them by the recognizer or the decoder.

Statistical pattern recognition and classification systems are based on the assumption that the conditional probability density functions of the features can be approximated. Many probabilistic models in statistical recognition and classification systems approximate the features' joint PDF by a Gaussian PDF or a mixture of Gaussian PDFs. Since the measurements are not necessarily jointly normal, power transforms are used in statistical analysis to get features that better satisfy the normality assumption [34]. Moreover, in many high-dimensional applications, the values of the correlation between different features are ignored. This is achieved by assuming that the observations are conditionally independent given some intermediate class label (e.g., given the Gaussian component label in a diagonal-covariance Gaussian mixture model [85], or given the class label in a naive Bayes classifier [86]). The

computational efficiency requirements often motivate this assumption, although it is known to be unjustified in many applications of interest, e.g., in speech [35], image [87], and text [86] applications. This makes the problem of finding the features that are best represented with these models equivalent to the problem of finding the conditionally independent components of the original features for each one of these intermediate class labels. Previous approaches to this problem formulated it as a redundancy reduction problem that can be solved by using a more relaxed model or by using a linear transform of the data. In [88], we formulated the problem as a nonlinear independent component analysis (NICA) problem. We showed that using the features generated using NICA in speech recognition increased the phoneme recognition accuracy compared to the baseline system and compared to systems that used linear transforms like linear ICA [47], linear discriminant analysis (LDA) [16], and maximum likelihood linear transform (MLLT) [55]. We showed also that the NICA algorithm described in [88] can be formulated as a generalization of the MLLT.

In this chapter, we will introduce a unified information-theoretic approach to feature transformation that makes no assumptions about the true probability density function of the original data and can be applied for any probabilistic model with arbitrary constraints. Both power transforms and redundancy reduction approaches can be formulated as special cases of what we call a model enforcement approach: the model enforcement approach estimates a nonlinear transform and the parameters of the probabilistic model that jointly minimize the relative entropy between the true joint features PDF and its hypothesized model.

In the next section, we will give a brief overview of parametric approaches for Statistical modeling. We will describe the main problems with previous approaches to redundancy reduction techniques and transformation for normality in Section 3.2, then we will formulate the model enforcement problem and provide a unified feature transformation framework that has most previous approaches as special cases of it in Section 3.3 [89]. Then, we apply this framework to extend ICA to the case of nonlinearly mixed sources in Section 3.4 as presented in [88]. An iterative algorithm is described also in Section 3.4 to estimate features for ASR

based on nonlinear ICA. In Section 3.5, we describe a special case of the model enforcement problem that reduces to a maximum likelihood estimation of the parameters of a volume-preserving transform and the model. In Section 3.6, a large-vocabulary conversational speech recognition application of the algorithm discussed in Section 3.5 is introduced. Finally, in Section 3.7 the level at which this technique for feature transformation is applied in ASR systems is discussed.

3.1 Parametric Approach for Statistical Modeling

Bayes rule is the optimal classification rule if the underlying distribution of the data is known. In practice, we do not know the underlying distribution. There are two main approaches to this problem: parametric and nonparametric [16]. In nonparametric approaches like kernel-based approaches the decision boundaries between the classes are estimated directly instead of trying to estimate the conditional density of the classes while parametric approaches estimate a parametric model of the conditional PDFs. In this chapter, we will limit our discussion to the parametric approaches.

In parametric statistical modeling for classification and recognition, a probabilistic model is chosen and its parameters are trained to optimize a certain criterion under the assumption that the true PDF of the features can be approximated well by the model. Parameter optimization takes place without questioning the validity of this assumption. Since the features are usually chosen based on prior knowledge about the task using heuristic approaches, this assumption is in most cases unjustified.

Information theory provides a measure by which we can say how well a PDF is approximated by another PDF [90]. This measure is called the divergence, Kullback-Liebler distance, or the relative entropy and is defined by

$$R(P,\hat{P}) \;=\; E_P\left[\log\left(\frac{P}{\hat{P}}\right)\right], \tag{3.1}$$

where P is the true PDF and $\hat{P}$ is the approximate PDF. An important property of the relative entropy is that

$$R(P,\hat{P}) \geq 0$$

with equality if and only if

$$\hat{P} = P$$

in the expectation sense.

Most parametric statistical classification systems use maximum likelihood estimation (MLE) or Bayesian methods to estimate the parameters of the model. The popularity of MLE is attributed to the existence of efficient algorithms to implement it, like the expectation-maximization (EM) algorithm, and to its consistency and asymptotic efficiency, if the true PDF belongs to the admissible set of parameterized PDF models [91].

In the MLE method, the parameters λ^* are estimated given a set of i.i.d observations $\{\mathbf{x}^i\}_{i=1}^N$ by maximizing the functional

$$L_{emp} \;=\; \sum_{i=1}^{N} \log \hat{P}(\mathbf{x}^i, \lambda) \tag{3.2}$$

with respect to the parameters λ.

Maximizing this empirical functional is equivalent to minimizing an empirical estimate of the relative entropy between the true PDF and the hypothesized PDF model

$$R_{emp}(P, \hat{P}) \quad = \quad -H(\mathbf{x}) - \frac{1}{N} \sum_{i=1}^{N} \log \hat{P}(\mathbf{x}^i, \lambda), \tag{3.3}$$

where $H(\mathbf{x})$ is the differential entropy of the random vector $\mathbf{x}$.

Vapnik and Chervonenkis show that the necessary and sufficient condition of the consistency of this maximization problem is [92]

$$Pr\left(\sup_{\lambda \in \mathbf{\Lambda}} |R(P, \hat{P}) - R_{emp}(P, \hat{P})| \geq \epsilon\right) \rightarrow 0 \tag{3.4}$$
$$\text{for } N \rightarrow \infty \text{ and } \forall \epsilon > 0,$$

where $\{\mathbf{x}^i\}_{i=1}^N$ are generated by any admissible PDF $\hat{P}(\mathbf{x}, \lambda_0)$, $\forall \lambda_0 \in \mathbf{\Lambda}$.

However, as we do not know the true PDF, we cannot guarantee small approximation error. A small approximation error can be achieved by using a complex structure of the hypothesized models that can approximate a large set of PDFs. On the other hand, this increases the computational and conceptual complexity of the system, and increases the required amount of training data to get a good estimate of the model parameters.

An important property of any classification or recognition model that is related to consistency is its generalization ability. The generalization ability is a monotonically increasing function of the ratio of the number of available training vectors and the VC dimension of the family of the hypothesized PDFs [92]. This means that the requirements of generalization ability conflict with the requirements of decreasing the approximation error.

One way of controlling the complexity of the model is by using a relatively simple probabilistic model and a transform of the observation vector to a new feature vector whose PDF is better modeled by the hypothesized PDF based on certain criterion. Many previous approaches to feature transformation show improvement in classification and recognition accuracy compared to using more complex probabilistic models for the same number of

parameters [55], [87], and [88]. Most of these methods, as will be discussed in the next section, are linear transformations that use a number of parameters equal to the square of the dimension of the feature vector. Our approach provides a generalization to nonlinear transformations that is more flexible in selecting the number of the parameters of the transform, as it is linear in the dimension of the input features.

3.2 Limitations of Previous Approaches

Transformations to achieve normality, described in Chapter 2, were constrained to using a restricted family of power transforms and to a Gaussian hypothesized model. These transforms were scalar transforms, i.e., each transformed feature is obtained from a single input measurement.

In previous chapters, the importance of modeling the covariance structure of the probabilistic models in an efficient way, or generating a set of features that satisfy the diagonal covariance assumptions, was discussed. The brief review provided in Chapter 2 shows that all model-based and feature-based approaches were based on linear transformation of the parameters of the model or the feature space, respectively. We will concentrate here on factor analysis (FA) and independent component analysis (ICA), because principal component analysis (PCA) [16] and maximum likelihood linear transform (MLLT) [55] can be shown to be special cases of these approaches. In factor analysis, the observed feature vector is assumed to be related to latent variables (factors) by the relation

$$\mathbf{x} \;=\; \mathbf{\Lambda z} + \mathbf{v}, \tag{3.5}$$

where $\mathbf{x} \in \Re^n$ is the observation vector, $\mathbf{\Lambda}$ is an $n \times p$ matrix where $p \leq n$, $\mathbf{z} \in \Re^p$ is the factor vector that denotes a Gaussian random vector with zero mean and identity covariance matrix, and $\mathbf{v} \in \Re^n$ is an independent Gaussian noise vector with diagonal covariance ψ.

Using this model for the acoustic feature vector in speech recognition, the covariance matrix of this feature vector can be written as [39]

$$\mathbf{\Sigma} = \mathbf{\Lambda}\mathbf{\Lambda}^T + \psi. \tag{3.6}$$

The parameters of this model are then optimized to maximize the likelihood using the EM algorithm [93] or to minimize an empirical estimate of the recognition error using the minimum classification error (MCE) approach [40].

On the other hand, the goal of ICA algorithms can be formulated as finding the linear transformation $\mathbf{W}$ of the dependent observation vector $\mathbf{X}$ that makes the outputs as statistically independent as possible. This means minimizing the mutual information of the output vector $\mathbf{Y}$, since

$$I(\mathbf{Y}) \geq 0,$$

with equality if and only if the output vector components are statistically independent.

Both FA and ICA algorithms assume that the factors are mixed linearly to generate the observations data. In many interesting applications, this assumption is unjustified or unacceptable. An example is the speech recognition problem, as all acoustic features used in speech recognition cannot be modeled as a linear mixture of independent sources of variations in the speech signal. To be more specific, let us concentrate, for example, on the standard form of these features as coefficients in the cepstral domain. Coarticulation effects and additive noise are examples of independent sources in the speech signal that are nonlinearly combined in the cepstral domain with the information about the vocal tract shape that is important for recognition. The source-filter model proposes that the excitation signal and the vocal tract filter are linearly combined in the cepstral domain, but the source-filter model is unrealistic in many cases, especially for consonants. Time-varying filters and filter-dependent sources result in a nonlinear source-filter combination in the cepstral

domain [94]. In image and face recognition also, there are deformations like bending which result in correlations that cannot be compensated for by a linear transform. In case of Gaussian or mixture of Gaussian hypothesized PDF, this sufficiency of linear transformation assumption is equivalent to assuming that the true conditional joint PDFs of the features are Gaussian or mixture of Gaussian PDFs, respectively. This is due to the fact that any linear transformation of a Gaussian random vector results in a Gaussian random vector. This limitation was alleviated in the nonlinear independent component analysis approach proposed in [88]. However, the statistical independence constraint is only one of many possible constraints that may be imposed on the probabilistic models used in classification and recognition systems. For HMM recognizers with diagonal-covariance Gaussian mixtures, the statistical independence constraint is conditional on the Gaussian component of the mixture. This problem can be solved by using a different map for each Gaussian component, but this solution may be impractical in a large vocabulary speech recognition system with hundreds of thousands of Gaussian PDFs. There is another problem not addressed by these previous approaches, a problem actually common to all previous approaches to the problem of incorrect probabilistic model assumptions in speech recognition described in Chapter 2. The approaches neglect the effect of having an incorrect parametric model of the PDF. For example, the features can be statistically independent, but their PDF is different from the Gaussian PDF imposed by the model. This mismatch affects both model-based and feature-based approaches. In model-based approaches, a model with more relaxed constraints is assumed, and the additional parameters of this model are trained by MLE or a discriminative training algorithm. This takes place without questioning the validity of the new relaxed model of the features, or discussion of how the MLE or discriminative training of these parameters will improve the overall performance [11], [39]. In feature-based approaches like MLLT, an intuitive argument was presented in [55] proving that, for a diagonal covariance Gaussian PDF, using a full-rank linear map that is estimated by maximizing the likelihood in the new feature space will increase the likelihood of the training data, but it was not

generalized to other PDFs or more general transformations. In the work presented in [88] and described in Section 3.4, we proved that for any PDF that assumes statistical independence of the features and a volume-preserving mapping of the features, maximizing the likelihood in a new feature space will converge under some consistency constraints to the true likelihood in the original feature space. This convergence cannot be achieved by MLE of the parameters of the model in the original feature space unless the original features are already statistically independent. A generalization to other model assumptions is needed and will be provided in the next section.

In the recent subspace-constrained precision matrices and means (SPAM) approach to the redundancy reduction problem, the precision matrices of the model (i.e., the inverse covariance matrices) are constrained to lie in a subspace of the space of all symmetric $n \times n$ matrices and the mean vectors are constrained to lie in a subspace of R^n, where n is the features vector dimension. The percesion matrices are represented by the basis expansion

$$\mathbf{\Sigma}_j^{-1} = \sum_{k=1}^{K} \lambda_k^j \mathbf{\Lambda}_k, \tag{3.7}$$

where j is the Gaussian component index and $\{\lambda_k^j\}_{k=1}^K$ are the untied parameters for the jth Gaussian component, the basis symmetric matrices $\{\mathbf{\Lambda}_k\}_{k=1}^K$ are tied across all Gaussian components, and K is the number of basis matrices [95]. This approach assumes that a linear subspace projection of the full-covariance models can provide good performance for fewer parameters than those needed by the full-covariance model. It is not clear, however, what feature space conditions are required for this assumption to be valid. It suffers also from the problems of not accounting for nonlinear sources of the correlation in the feature vector, and using a probabilistic model with relaxed constraints without questioning the validity of the new relaxed model. Also, the relaxation of the diagonal-covariance constraint on the Gaussian components achieved by the SPAM approach can be combined with any feature transformation using the model enforcement approach described in the next section.

3.3 A Unified Information-Theoretic Approach to Model Enforcement

Bayesian classifiers rely on models of the *a priori* and class-conditional feature distributions; the classifier is trained by optimizing these models to best represent features observed in a training corpus according to certain criteria. In many problems of interest, the true class-conditional feature probability density function (PDF) is not a member of the set of PDFs the classifier can represent. Previous research has shown that the effect of this problem may be reduced either by improving the models, or by transforming the features used in the classifier. This section addresses this model mismatch problem in statistical identification, classification, and recognition systems. In the previous section, we described many serious limitations of previous techniques. We formulate the problem as the problem of minimizing the relative entropy, also known as the Kullback-Liebler distance, between the true conditional probability density function and the hypothesized probabilistic model.

The goal of this section is to generalize feature transformation in two ways. First, we will provide a feature transformation framework that makes no assumptions about the probabilistic model and the constraints imposed on it. This provides us with the flexibility needed to address problems in which the model is not necessarily Gaussian and does not assume the features are uncorrelated or independent, but assumes a certain parametric form of the features' conditional PDFs. Second, we will provide a nonlinear transform, as opposed to previous linear transforms, that is based on this framework. This nonlinear transform is a vector-based transform, as opposed to previous scalar power transforms. The number of parameters of this transform is linear in the dimension of the input feature vector, while it is quadratic for linear transforms. We will show also how all previous transforms to normality and redundancy reduction approaches discussed in Chapter 2 and the previous section are special cases of the information-theoretic model enforcement approach proposed here. Based on this formulation, we provide a computationally efficient solution to the problem based on

volume-preserving maps; existing linear transform designs are shown to be special cases of the proposed solution. Using this result, we provide a nonlinear extension of ICA and present the symplectic maximum likelihood transform (SMLT), a nonlinear volume-preserving extension of the maximum likelihood linear transform (MLLT). This approach has many applications in statistical modeling, classification, and recognition. We apply it to the maximum likelihood estimation of the joint probability density function (PDF) of order statistics and show a significant increase in the likelihood for the same number of parameters. We provide also phoneme recognition experiments that show recognition accuracy improvement compared to using the baseline Mel-frequency cepstrum coefficient (MFCC) features or using MLLT. Then, we present an iterative algorithm to jointly estimate the parameters of the symplectic map and the probabilistic model for both applications.

3.3.1 Problem formulation

Motivated by the discussion of the previous sections, we will choose any hypothesized parametric family of distributions to be used in our probabilistic model, and search for a map of the features that improves the validity of our model. To do that, we will need the following theorem.

Theorem 3.1 *Let* $\mathbf{y} = f(\mathbf{x})$ *be an arbitrary one-to-one map of the random vector* $\mathbf{x}$ *in* $\Re^n$ *to* $\mathbf{y}$ *in* $\Re^n$*, and let* $\hat{P}_{\mathbf{\Lambda}}(\mathbf{Y})$ *be a hypothesized parametric family of density functions. The map* $f^*(.)$ *and the set of parameters* $\mathbf{\Lambda}^*$ *minimize the relative entropy between the hypothesized and the true PDFs of* $\mathbf{y}$ *if and only if they also maximize the objective function*

$$V = E_{P(\mathbf{y})}\left[\log\left(|det\left(\mathbf{J}_f\right)|\right) + \log \hat{P}_{\mathbf{\Lambda}}(\mathbf{y})\right], \tag{3.8}$$

where $\mathbf{J}_f$ *is the Jacobian matrix of the map* $f(.)$*.*

Proof:

We will rewrite the expression for the relative entropy after an arbitrary transformation $\mathbf{y} = f(\mathbf{x})$ of the input random vector $\mathbf{x}$ in $\Re^n$, as

$$R(P(\mathbf{y}), \hat{P}(\mathbf{y})) \quad = \quad -H(P(\mathbf{y})) - E_{P(\mathbf{y})}\left[\log\left(\hat{P}(\mathbf{y})\right)\right], \tag{3.9}$$

where $H(P(\mathbf{y}))$ is the differential entropy of the random vector $\mathbf{y}$ based on its true PDF $P(\mathbf{y})$.

The relation between the output differential entropy and the input differential entropy is in general [96]

$$H(P(\mathbf{y})) \quad \leq \quad H(P(\mathbf{x})) + \int_{\Re^n} P(\mathbf{x}) \log\left(\left|det\left(\mathbf{J}_f\right)\right|\right) \mathbf{dx}, \tag{3.10}$$

where $P(\mathbf{x})$ is the probability density function of the random vector $\mathbf{x}$, for an arbitrary transformation $\mathbf{y} = f(\mathbf{x})$ of the random vector $\mathbf{x}$ in $\Re^n$, with equality if $f(\mathbf{x})$ is invertible.

Therefore the relative entropy can be written as

$$R(P(\mathbf{y}), \hat{P}(\mathbf{y})) \quad = \quad -H(P(\mathbf{x})) - E_{P(\mathbf{x})}\left[\log\left(\left|det\left(\mathbf{J}_f\right)\right|\right)\right] - E_{P(\mathbf{y})}\left[\log \hat{P}(\mathbf{y})\right] \tag{3.11}$$

for an invertible map $\mathbf{y} = f(\mathbf{x})$.

The expectation of a function $g(\mathbf{x})$ for an arbitrary one-to-one map $\mathbf{y} = f(\mathbf{x})$ can be written as [96]

$$E_{P(\mathbf{x})}\left[g(\mathbf{x})\right] \quad = \quad E_{P(\mathbf{y})}\left[g(f^{-1}(\mathbf{y}))\right], \tag{3.12}$$

where $f^{-1}(.)$ is the inverse map.

Therefore,

$$R(P(\mathbf{y}), \hat{P}(\mathbf{y})) \;=\; -H(P(\mathbf{x})) - E_{P(\mathbf{y})}\left[\log\left(|det\left(\mathbf{J}_f\right)|\right) + \log \hat{P}(\mathbf{y})\right]. \qquad (3.13)$$

Equation (3.13) proves the theorem. ■

Theorem 3.1 states that minimizing the relative entropy is equivalent to maximizing the sum of the expected log likelihood and a cost function; the cost function is determined by the determinant of the Jacobian matrix of the transform. This cost function guarantees that maximizing the likelihood of the transformed features will not be at the expense of their information content measured by their differential entropy. It should be noted that the objective function is the likelihood in the original feature space given the probabilistic model in the new feature space.

3.3.2 A maximum likelihood approach to model enforcement

For a nonlinear feature transformation, the Jacobian matrix of the transformation is a function of the values of the feature vectors. This makes the maximization of the objective function for a high-dimensional input feature vector computationally expensive. A significant reduction in the computational complexity is achieved by an important special case. This special case that reduces the problem to maximum likelihood estimation (MLE) of the model and map parameters is given in the following lemma, but first we need to define volume-preserving maps in $\Re^n$, where n is an arbitrary positive integer [97].

Definition 3.1 A C^∞ map $f : S_{\mathbf{x}} \rightarrow S_{\mathbf{y}}$, where $S_{\mathbf{x}} \subset \Re^n$ and $S_{\mathbf{y}} \subset \Re^n$ is said to be volume-preserving if and only if $|det\left(\mathbf{J}_f\right)| = 1 \;\forall \mathbf{x} \in S_{\mathbf{x}}$.

Lemma 3.1 *Let* $\mathbf{y} = f(\mathbf{x})$ *be an arbitrary one-to-one volume-preserving map of the random vector* $\mathbf{x}$ *in* $\Re^n$ *to* $\mathbf{y}$ *in* $\Re^n$*, and let* $\hat{P}_{\mathbf{\Lambda}}(\mathbf{y})$ *be a hypothesized parametric family of density functions. The map* $f^*(.)$ *and the set of parameters* $\mathbf{\Lambda}^*$ *jointly minimize the relative entropy between the hypothesized and the true PDFs of* $\mathbf{y}$ *if and only if they also maximize the expected log likelihood based on the hypothesized PDF,* $E_{P(\mathbf{y})}\left[\log \hat{P}_{\mathbf{\Lambda}}(\mathbf{y})\right]$.

Using the definition of the volume-preserving maps, the proof of the lemma is straightforward. The lemma proves that the maximum likelihood criterion is the appropriate model enforcement criterion for any volume-preserving transform. By reducing the problem to MLE, efficient algorithms based on the incremental EM algorithm can be designed [98].

3.3.3 Generality of the model enforcement approach

Theorem 3.1 generalizes the previous approaches in two ways. First, transforms can be designed to satisfy arbitrary constraints on the hypothesized PDF, not necessarily those that impose an independence or decorrelation constraint on the features, and the hypothesized PDF is not necessarily Gaussian or mixture of Gaussians. Second, the feature transformation is not necessarily linear. To show the generality of Theorem 3.1 and its wide range of applications, we relate it to previous methods.

Transformations to normality described in Chapter 2 are a special case of Theorem 3.1 by constraining the PDF model to be Gaussian and the transform to be a power transform.

PCA may be viewed as a special case of Theorem 3.1 under two equivalent constraints. First, if the transform is constrained to be linear and the model PDF is constrained to be a diagonal-covariance Gaussian, then Theorem 3.1 reduces to PCA. Equivalently, if the true feature PDF is assumed to be Gaussian, and the model PDF is constrained to be a diagonal-covariance Gaussian, Theorem 3.1 reduces to PCA. Probabilistic PCA (PPCA) is a generalization of PCA that can be shown as an application of Theorem 3.1 when the

hypothesized model of the joint PDF assumes that the features are uncorrelated but not necessarily Gaussian.

ICA also can be shown as a special case of Theorem 3.1 when the hypothesized model assumes statistical independence of the transformed features and the transform is constrained to be linear. Nonlinear ICA removes the constraint that the transform must be linear. Factor analysis is also a special case of Theorem 3.1 by assuming that the hypothesized joint PDF is Gaussian with special covariance structure.

MLLT is a special case of Theorem 3.1 by using a linear map of the features and assuming the hypothesized joint PDF is Gaussian or a mixture of Gaussians. As we highlighted before, these two assumptions of linearity and Gaussianity together are equivalent to the assumption that the original features are Gaussian.

It should be noted that all linear maps designed to improve the satisfaction of the features of a given model are special cases of Lemma 3.1, as any linear map is equivalent to a linear volume-preserving map multiplied by a scalar.

3.4 A Nonlinear Independent Component Analysis Application

In the previous section, we showed that by using a volume-preserving map, the model enforcement problem is reduced to maximizing the likelihood of the output components. In this section, an extension of the ICA algorithms to nonlinearly mixed sources is introduced. Our goal is to find the mixing functions and the independent components given the observations. By restricting the mixing function to a class of volume-preserving transforms, we will show that this approach is a direct application of Lemma 3.1 in the previous section. This section therefore develops a maximum likelihood volume-preserving nonlinear transform algorithm for the case when the probabilistic model assumes statistical independence of the elements of the feature vector. The resulting algorithm may be considered a nonlinear generalization of ICA with a more flexible parameter count than ICA; experiments in the next section

show that the algorithm outperforms ICA with fewer trainable parameters. The maximum likelihood approach using volume-preserving maps is a good compromise between the two extremes of previous linear approaches with their simplicity and computational efficiency but inadequacy in many applications, and the nonlinear approaches with their generality but computational complexity associated with calculating the determinant of the Jacobian matrix.

3.4.1 Problem formulation

Since the components are assumed to be statistically independent, we have to find the solution that minimizes the mutual information of the output components $I(\mathbf{Y})$ [90]. The mutual information is a function of the output differential entropy,

$$I(\mathbf{Y}) = \sum_{i=1}^{n} H(Y_i) - H(\mathbf{Y}), \tag{3.14}$$

where n is the number of components of the output vector, and Y_i is the ith component of the vector $\mathbf{Y}$.

To have a well-defined optimization problem, we need some restrictions on the nonlinear mixing function or the criterion that the solution should optimize. For a continuous random vector $\mathbf{Y} \in \Re^n$, the mutual information is invariant to scaling but differential entropy is sensitive to it. To avoid this scale-sensitivity problem, and the need of having an estimate of the joint probability density function to calculate the differential entropy of the output vector, we choose to keep the output differential entropy equal to the input differential entropy $H(\mathbf{Y}) = H(\mathbf{X})$, while minimizing $\sum_{i=1}^{n} H(Y_i)$ to minimize the mutual information of the output vector.

It will be shown also that this choice leads to minimizing the negative of the empirical function used in maximizing the likelihood of the output vectors. This means that this

approach produces a maximum likelihood transform as described in Lemma 3.1 under the constraint of the output components' independence.

As was shown in the previous section, the input and output differential entropy are equal if and only if the map is volume-preserving. Symplectic maps are a class of volume-preserving maps with useful properties. An interesting property of any nonreflecting symplectic transformation from $\mathbf{x}$ to $\mathbf{y}$ is that it can be represented using a scalar function $g(.)$ such that [99]

$$\mathbf{y} = \mathbf{x} - \mathbf{Q}^{-1}\frac{\partial}{\partial \mathbf{u}}g(\mathbf{u}); \tag{3.15}$$

$$\mathbf{u} = \frac{\mathbf{x}+\mathbf{y}}{2};$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{0} & -\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix};$$

$$\mathbf{Q} = -\mathbf{Q}^{-1}, \tag{3.16}$$

where $\mathbf{I}$ denotes the identity matrix in $\Re^{n/2}$. The gradient is to be taken with respect to the argument $\mathbf{u}$.

Now the nonlinear ICA problem can be formulated as the problem of finding the function $g(\mathbf{u})$ that minimizes $\sum_{i=1}^{n} H(Y_i)$ under the constraint that $H(\mathbf{Y}) = H(\mathbf{X})$ guaranteed by the symplectic map. The minimum of this sum , $H(\mathbf{X})$, is independent of the symplectic map parameters, as for any random vector $\mathbf{Y}$,

$$\sum_{i=1}^{n} H(Y_i) \geq H(\mathbf{Y}). \tag{3.17}$$

We use a multilayer feed-forward neural network to get a good approximation of the scalar function $g(\mathbf{u})$ [100]. The parameters of this network are optimized to minimize $\sum_{i=1}^{n} H(Y_i)$ under the constraint $H(\mathbf{Y}) = H(\mathbf{X})$, i.e.,

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \sum_{i=1}^{n} H(Y_i), \tag{3.18}$$

where

$$\mathbf{W} = (\mathbf{A}, \mathbf{B}),$$

$$g(\mathbf{u}, \mathbf{A}, \mathbf{B}) = \sum_{j=1}^{M} b_j S(\mathbf{a}_j \mathbf{u}), \tag{3.19}$$

where $S(.)$ is a nonlinear function like sigmoid or hyperbolic tangent, $\mathbf{a}_j$ is the jth row of the $M \times n$ matrix $\mathbf{A}$, and b_j is the jth element of the $M \times 1$ vector $\mathbf{B}$. The constant offset term that is usually used was omitted.

3.4.2 Efficient estimation of the objective function

The objective function to be minimized is

$$V = \sum_{i=1}^{n} H(Y_i). \tag{3.20}$$

The differential entropy of a random variable is by definition the negative of the expectation of the logarithm of its probability density function

$$H(Y_i) = -E[\log P(Y_i)], \text{ for } i = 1, 2, \cdots, n. \tag{3.21}$$

Since we do not have the true probability density function of the random variable Y_i, and

all we can calculate is a finite set of realizations of this random variable $\{y_i^1, y_i^2, \cdots, y_i^N\}$ of size N, the expectation will be approximated by the sample mean of the given values of the random vector. This is justified by the weak law of large numbers, which states that if the set $\{y^1, y^2, \cdots, y^N\}$ are independent and identically distributed, then [96]

$$\frac{1}{N}\sum_{i=1}^{N} y_j^i \rightarrow E[Y_j], \text{ in probability as } N \rightarrow \infty. \tag{3.22}$$

This gives the empirical estimate of the objective function as

$$V_{emp} = -\sum_{i=1}^{N}\sum_{j=1}^{n} \log P(Y_j = y_j^i), \tag{3.23}$$

where N is the number of samples used to estimate V_{emp}. Again, we do not have the true probability density functions of each component; therefore, we use a maximum likelihood parameterized estimate of these probability density functions.

This gives the final form of the empirical estimate of the objective function as

$$V_{emp} = -\sum_{i=1}^{N}\sum_{j=1}^{n} \log P_{\mathbf{\Lambda_j}}(Y_j = y_j^i), \tag{3.24}$$

where $P_{\mathbf{\Lambda}_j}(Y_j)$ is the parameterized estimate of $P(Y_j)$ defined by the parameters $\mathbf{\Lambda}_j$ for $j = 1, 2, \cdots, n$.

Minimizing this expression is equivalent to maximizing the estimated log likelihood of the output vectors, under the assumption that the features are independent. This means that this approach can be considered as a generalization of maximum likelihood approaches to ICA to the nonlinear mixing case. Maximum likelihood approaches to ICA are closely related to the MLLT introduced in [55]. The difference is mainly in replacing the output coefficients' independence constraint of ICA by the diagonal-covariance constraint of the

Gaussian mixture model in MLLT.

To calculate the gradient of the objective function with respect to the symplectic transformation parameters, we need to calculate its derivative with respect to each parameter.

In general,

$$\frac{\partial V_{emp}}{\partial a_{qr}} = -\sum_{i=1}^{N}\sum_{j=1}^{n}\frac{\partial P_{\mathbf{\Lambda}_j}(Y_j = y_j)}{\partial y_j}\frac{\partial y_j}{\partial a_{qr}}(\log P_{\mathbf{\Lambda}_j}(Y_j = y_j) + 1)|_{y_j = y_j^i}, \tag{3.25}$$

$$\frac{\partial V_{emp}}{\partial b_q} = -\sum_{i=1}^{N}\sum_{j=1}^{n}\frac{\partial P_{\mathbf{\Lambda}_j}(Y_j = y_j)}{\partial y_j}\frac{\partial y_j}{\partial b_q}(\log P_{\mathbf{\Lambda}_j}(Y_j = y_j) + 1)|_{y_j = y_j^i}, \tag{3.26}$$

for

$$q = 1, 2, \cdots, M,$$

and

$$r = 1, 2, \cdots, n.$$

Therefore,

$$\frac{\partial y_j}{\partial a_{qr}} = -h(j)\frac{\partial^2 g(\mathbf{u})}{\partial u_{j+h(j)\frac{n}{2}}\partial a_{qr}}, \tag{3.27}$$

$$\frac{\partial y_j}{\partial b_q} = -h(j)\frac{\partial^2 g(\mathbf{u})}{\partial u_{j+h(j)\frac{n}{2}}\partial b_q}, \tag{3.28}$$

$$h(j) = \begin{cases} -1 & \text{if } j \geq \frac{n}{2} \\ 1 & \text{if } j < \frac{n}{2}\ . \end{cases}$$

This formulation of the symplectic parameters evaluation as a minimization problem is ill-posed, as very small changes in the values of the parameters may lead to drastic changes in the objective function. The instability of the solution arises from the fact that the output is related to the input using an implicit form. Once the instability for a given set of data samples causes the absolute value of the symplectic parameters to be large, the output of the

feed-forward network saturates and becomes less dependent on the values of the symplectic parameters. This means that the algorithm will not converge and the effect of any additional input data sets on the final values of the symplectic parameters will be minimal. To make the problem well posed, the map from $\mathbf{X}$ to $\mathbf{Y}$ should be continuous, and the map from $\mathbf{X} \times \mathbf{Y}$ to $g(.)$ should be continuous also. If $\mathbf{Y}$ was represented as an explicit function of $\mathbf{X}$ using the feed forward neural network, the continuity of both maps will be forced by this representation. But due to the implicit function representation of the symplectic map, the value of $\mathbf{y}$ for a given $\mathbf{x}$ is estimated by an optimization problem and the map is no longer guaranteed to be continuous. The problem becomes well-posed by restricting the set from which $g(\mathbf{u})$ is chosen to some compact set $\mathbf{G}$. One can show by virtue of the operator inversion lemma [101] that in this case the problem of empirical risk minimization becomes well posed. One can show also that if $g(\mathbf{u})$ is sufficiently well-behaved, i.e, has a finite covering number, the empirical objective function will converge to the actual objective function for increasing sample size N, i.e.,

$$Pr\left(\sup_{g\in\mathbf{G}}|V[g]-V_{emp}[g]|\geq\epsilon\right)\rightarrow 0 \qquad \text{for } N\rightarrow\infty \text{ and } \epsilon>0. \tag{3.29}$$

Vapnik and Chervonenkis show that such a condition is necessary and sufficient to give uniform convergence bounds [102]. Classical regularization theory provides a solution to this type of problem in which a function is to be approximated from sparse data [103]. It formulates the regression problem as a variational problem of finding the function $g(\mathbf{u}) \in \mathbf{G}$ that minimizes the functional

$$E_g = \frac{1}{N}\sum_{i=1}^{N} V_{emp}(\mathbf{x}^i, \mathbf{y}^i, g) + \lambda\|g\|_K^2, \tag{3.30}$$

where $\|g\|_K^2$ is a norm in a reproducing kernel Hilbert space $\mathbf{G}$ defined by the positive definite

function K, N is the number of data samples. The functionals of classical regularization lacked a rigorous justification for a finite set of training data. Vapnik has provided a general theory that justifies regularization functionals for learning from a finite set of data [92]. In the framework of structural risk minimization (SRM) suggested by Vapnik, [92], [104] , we can define a structure using a nested sequence of hypothesis spaces $\mathbf{G}_1 \subset \mathbf{G}_2 \subset \cdots \subset \mathbf{G}_{l(N)}$ with $\mathbf{G}_m$ being the set of functions $g(\mathbf{u})$ in the reproducing kernel Hilbert space (RKHS) with

$$\|g\|_K \quad \leq \quad C_m, \tag{3.31}$$

where $\{C_m\}_{m=1}^{l(N)}$ is a monotonically increasing sequence of positive constants. For each m, we are supposed to minimize the empirical objective function subject to this constraint. This in turn leads to using the Lagrange multiplier λ_m and to minimizing

$$\frac{1}{N}\sum_{i=1}^{N} V_{emp}(\mathbf{x}^i, \mathbf{y}^i, g) + \lambda_m(\|g\|_K^2 - C_m^2),$$

with respect to the symplectic parameters and maximizing with respect to $\lambda_m \geq 0$. The solution of this optimization problem is the same as the solution for minimizing

$$\frac{1}{N}\sum_{i=1}^{N} V_{emp}(\mathbf{x}^i, \mathbf{y}^i, g) + \lambda^*(N)(\|g\|_K^2 - C_m^2),$$

with respect to the symplectic maps, where $\lambda^*(N)$ is the optimal Lagrange multiplier corresponding to the optimal element of the structure $C_{l^*(N)}$.

In practice, this structure is formulated by imposing a convex penalty term on some quantity $K(g)$ related to $g(\mathbf{u})$, which is not necessarily the norm of the function in the reproducing kernel Hilbert space [105]. This functional has to be convex and continuous. The value of $\lambda^*(N)$ is usually chosen from a finite set of possible values or set to a constant

value.

In this work, we used the square of the ℓ_2 norm of the symplectic parameters vector $\mathbf{W} = vect(\mathbf{A}, \mathbf{B})$,

$$\|\mathbf{W}\|_2^2 = \sum_{i=1}^{m} |w_i|^2, \tag{3.32}$$

where w_i is the ith element of the vector $\mathbf{W}$, and m is the length of the vector, as the convex penalty and selected the optimal Lagrange multiplier $\lambda^*(N)$ from a finite set of 10 values. The value of $C_{l^*(N)}$ also was selected from a finite set of four values.

3.4.3 Maximum likelihood formulation of the problem

As shown before, the problem of nonlinear independent component analysis is reduced by using volume-preserving maps to the problem of minimizing the objective function

$$V_{emp} = -\sum_{i=1}^{N}\sum_{j=1}^{n} \log P_{\mathbf{\Lambda}_j}(Y_j = y_j^i), \tag{3.33}$$

where $P_{\mathbf{\Lambda}_j}(Y_j)$ is the parameterized estimate of $P(Y_j)$ defined by the parameters $\mathbf{\Lambda}_j$ for $j = 1, 2, \cdots, n$. This is clearly equivalent to maximizing the following objective function

$$L_{emp} = \sum_{i=1}^{N}\sum_{j=1}^{n} \log P_{\mathbf{\Lambda}_j}(Y_j = y_j^i). \tag{3.34}$$

Let the joint PDF of the output components under their independence constraint be $P_{ind}(\mathbf{y})$; then

$$P_{ind}(\mathbf{y}) = \prod_{j=1}^{n} P(y_j). \tag{3.35}$$

This means that our objective function to be maximized is

$$L_{emp} = \sum_{i=1}^{N} \log P_{ind}(\mathbf{y}^i). \tag{3.36}$$

This concludes the proof that the volume-preserving independent component analysis problem is a maximum likelihood problem in the new feature space under the independence constraint. This means that the symplectic parameters that make the output components as statistically independent as possible are the parameters that maximize the likelihood of the output components under the independence constraint that is imposed explicitly on the joint PDF. There is an implicit assumption here that $\prod_{j=1}^{n} P(y_j) = \prod_{j=1}^{n} P_{\mathbf{\Lambda}_j}(y_j)$, which is not necessarily true. The necessary and sufficient conditions for this to be true are the same conditions for consistency of the maximum likelihood estimate provided in [92] and [102]. Since maximum likelihood estimation is the most popular approach for estimating the parameters of the speech recognizer due to the existence of efficient algorithms to train the parameters like the expectation-maximization algorithm (EM) [24], this result allows us to jointly optimize the symplectic parameters and the recognizer parameters.

3.4.4 Implementation of the algorithm for speech processing

Initially, both the values of the symplectic map parameters $\mathbf{W}$ and the output vectors $\mathbf{y}$ are unknown, so we choose an initial value of the symplectic map parameters, then we solve the symplectic map equation for the output vectors. Given the output vectors corresponding to the input data, we use the EM algorithm to calculate the parameters of the probabilistic model. Based on this model, the empirical objective function is estimated, and the symplectic map parameters are updated using a conjugate gradient based method. This sequence is repeated until a local minimum of the empirical estimate of the objective function is achieved.

3.4.4.1 Estimation of the output vectors

To solve the symplectic transformation relation for the output vector given the input vector and the symplectic map parameters, the problem is formulated as an optimization problem. The output of the symplectic mapping is calculated using the conjugate gradient algorithm. The conjugate gradient algorithm [106], is used to calculate the output vector $\mathbf{y}$ that achieves the unconstrained minimum of

$$D(\mathbf{y}) = \left\| \mathbf{y} - \mathbf{x} + Q^{-1} \nabla g \left(\frac{\mathbf{x}+\mathbf{y}}{2} \right) \right\|^2 . \tag{3.37}$$

The updating rule at each iteration is

$$\mathbf{y}^{k+1} = \mathbf{y}^k + \alpha^k \mathbf{d}^k. \tag{3.38}$$

The directions of the conjugate gradient algorithm are generated by

$$\mathbf{d}^0 = -\nabla D(\mathbf{y}^0), \tag{3.39}$$

$$\mathbf{d}^k = -\nabla D(\mathbf{y}^k) + \zeta^k \mathbf{d}^{k-1}, \tag{3.40}$$

where ζ^k is given by

$$\zeta^k = \frac{\nabla D(\mathbf{y}^k)^T \nabla D(\mathbf{y}^k)}{\nabla D(\mathbf{y}^{k-1})^T \nabla D(\mathbf{y}^{k-1})}. \tag{3.41}$$

The scaling factor α^k of the direction in each iteration is selected based on the limited minimization rule on the interval $[0, h]$

$$D(\mathbf{y}^k + \alpha^k \mathbf{d}^k) = \min_{\alpha \in [0,h]} D(\mathbf{y}^k + \alpha \mathbf{d}^k), \tag{3.42}$$

using the golden-section search method. The algorithm is guaranteed to converge to a local minimum like all gradient-based optimization algorithms; because $D(\mathbf{y})$ is in general not a convex function of $\mathbf{y}$, convergence to a global minimum is not guaranteed. In practice, for about 90% of the input vectors, the algorithm converged in less than five iterations to a value of $D(\mathbf{y})$ less than 0.0001. Before using the regularization term in the objective function, the convergence of this algorithm was slow, and it sometimes failed to converge, when the input data consisted of time-domain speech samples.

The computational complexity of the algorithm for updating the output vectors in each iteration is $O((n + (n + 1)M)N)$, where n is the input vector length, M is the number of hidden nodes in the neural network, and N is the number of input vectors.

3.4.4.2 Evaluation of the parameters of the symplectic map

After calculating the output vectors corresponding to the initial map parameters, we use the conjugate gradient algorithm to find the set of the mapping parameters that minimize the regularized objective function E_g.

To be able to calculate the differential entropy of each component of the output $\mathbf{y}$ and its gradient, we have to define a parametric form of the PDF of the output components. In our experiments, we used both the mixture of Gaussians and the generalized Gaussian probabilistic model for each component. The motivation of choosing these specific forms is that both are general enough to approximate any PDF from the exponential family, while the mixture of Gaussians is the better choice to approximate a multimodal PDF. The mixture of Gaussians is usually used to model the conditional PDF of MFCC coefficients in speech recognition that is known to be multimodal [107], while the generalized Gaussian is known to approximate well the PDF of the time-domain speech samples that is known to be unimodal [51].

In all experiments described in this work, we used both parametric forms and reported the one that gave the best results. The generalized Gaussian PDF gave better results for

direct time-domain processing of the speech signal, while the mixture of Gaussians PDF gave better results for cepstral-domain experiments.

The mixture Gaussian model is given by

$$P(y_j) = \sum_{k=1}^{K} H_{jk} \frac{1}{\sqrt{2\pi}\sigma_{jk}} \exp\left(-\frac{(y_j - \mu_{jk})^2}{2\sigma_{jk}^2}\right), \tag{3.43}$$

$$\sum_{k=1}^{K} H_{jk} = 1$$

for all $j = 1, 2, \cdots, n$, where H_{jk} is the weight of the kth Gaussian PDF in the mixture of Gaussians, K is the number of Gaussian PDFs in the mixture of Gaussians, μ_{jk} is the mean of the kth Gaussian PDF in the mixture, σ_{jk}^2 is the variance of the kth Gaussian PDF in the mixture, and the generalized Gaussian probability distribution model for each component is

$$P(y_j) = \frac{\omega(\beta_j)}{\sigma_j} \exp\left[-c(\beta_j)\left|\frac{y_j - \mu_j}{\sigma_j}\right|^{2/(1+\beta_j)}\right] \tag{3.44}$$

for all $j = 1, 2, \cdots, n$, where

$$c(\beta_j) = \frac{\Gamma\left[\frac{3}{2}(1+\beta_j)\right]}{\Gamma\left[\frac{1}{2}(1+\beta_j)\right]^{1/(1+\beta_j)}}, \tag{3.45}$$

and

$$\omega(\beta_j) = \frac{\Gamma\left[\frac{3}{2}(1+\beta_j)\right]^{1/2}}{(1+\beta_j)\Gamma\left[\frac{1}{2}(1+\beta_j)\right]^{3/2}}, \tag{3.46}$$

where μ_j is the mean of y_j, σ_j^2 is the variance of y_j, and β is a measure of the kurtosis and a parameter that controls the distribution's deviation from normality. In the case of the

mixture Gaussian probabilistic model, the derivative of the probability density function is

$$\frac{\partial P(y_j)}{\partial y_j} = \sum_{k=1}^{K} -H_{jk} \frac{1}{\sqrt{2\pi}\sigma_{jk}} \frac{(y_j - \mu_{jk})}{\sigma_{jk}^2} \exp\left(-\frac{(y_j - \mu_k)^2}{2\sigma_k^2}\right), \tag{3.47}$$

and in the case of the generalized Gaussian distribution model, the derivative of the probability density function is

$$\frac{\partial P(y_j)}{\partial y_j} = -c(\beta_j) \frac{2}{1+\beta_j} \left|\frac{y_j - \mu_j}{\sigma_j}\right|^{\frac{2}{1+\beta_j}-1} P(y_j). \tag{3.48}$$

The parameters of these probabilistic models are calculated from the output data using the expectation-maximization (EM) algorithm [93].

We used the hyperbolic tangent function as the nonlinear function in the feed forward neural network approximation of the scalar function that is used in the symplectic map,

$$S(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \tag{3.49}$$

Therefore, the derivatives of the output components with respect to the symplectic map parameters become

$$\frac{\partial y_j}{\partial a_{qr}} = \begin{cases} 2h(j) b_q a_{qj+h(j)\frac{n}{2}} g(\mathbf{a}_q \mathbf{y})(1 - g^2(\mathbf{a}_q \mathbf{y})) \frac{x_r + \mathbf{y}_r}{2} & \\ & \text{if } r \neq j + h(j)\frac{n}{2} \\ 2h(j) b_q a_{qj+h(j)\frac{n}{2}} g(\mathbf{a}_q \mathbf{y})(1 - g^2(\mathbf{a}_q \mathbf{y})) \frac{x_r + \mathbf{y}_r}{2} & \\ -h(j) b_q (1 - g^2(\mathbf{a}_q \mathbf{y})) & \text{if } r = j + h(j)\frac{n}{2} \end{cases} \tag{3.50}$$

$$\frac{\partial y_j}{\partial b_q} = -h(j) a_{qj+h(j)\frac{n}{2}} (1 - g^2(\mathbf{a}_q \mathbf{y})). \tag{3.51}$$

Substituting the derivatives of the output components with respect to the symplectic map parameters and the derivatives of the probability density function with respect to the output components for both parametric PDF forms in Equations (3.25), (3.26), (3.27), and (3.28), we get the derivatives of the empirical objective function with respect to the symplectic parameters. Adding to these derivatives, the derivatives of the regularization term, we get the derivatives of the regularized objective function with respect to the symplectic parameters. Given these derivatives, we can use any gradient-based algorithm to update the values of the symplectic parameters. We chose the conjugate gradient algorithm due to its fast convergence compared to other gradient based methods. The computational complexity of the algorithm for updating the symplectic parameters in each iteration is $O((3nK + (n+1)M + n^2M)N)$, where n is the input vector length, M is the number of hidden nodes in the neural network, K is the number of Gaussian components in the mixture, and $K = 1$ for generalized Gaussian PDF, and N is the number of input vectors.

3.4.5 Experiments and results

Our approach to nonlinear ICA was applied to the speech signal. First, it was applied to the speech samples directly in the time domain, and then it was applied to the MFCC coefficients in the cepstral domain. The time domain processing of speech has applications in speech coding, prosody recognition, and speaker recognition, while processing of MFCC can be used in speech recognition.

In the direct time domain processing, the TIMIT speech database, with sampling rate at 16 KHZ, is downsampled to 8 KHZ and preemphasized. Each utterance of speech is divided into fixed-size frames of length 20 samples. Then 1000 of these frames are used at a time to update the values of the parameters of the symplectic transformation and the marginal probability density functions of the output components.

In the cepstral domain processing, the Mel-frequency cepstrum coefficients are calculated for 4500 utterances from the TIMIT database. The overall feature vector consists of 12 MFCC coefficients in the first two experiments in the cepstral domain. The last experiment uses 12 MFCC coefficients, energy, and their deltas. In both cases, this MFCC based feature vector is used as the input to our symplectic nonlinear independent component analysis.

In each iteration, the output components are calculated using the current symplectic transformation parameters by using the symplectic mapping equation, then the maximum likelihood estimates of the marginal probability density functions of the output components are calculated using the EM algorithm. Then, the sum of the differential entropy of the output components is calculated and its gradient and the symplectic mapping parameters are updated such that this sum is minimized. After the iterative algorithm converges to a set of locally optimal symplectic parameters, the training data are transformed by the symplectic map yielding corresponding output coefficients. The output coefficients are compared to LPCC and MFCC coefficients in their coding efficiency, and to LDA, linear ICA, and MLLT in their recognition accuracy.

3.4.5.1 Coding efficiency and sparseness of output coefficients

Coding efficiency of acoustic features that are used in speech recognition is receiving much more attention recently. This is due to the growing interest in distributed speech recognition systems, especially over limited bandwidth networks like wireless networks. We used the empirical estimate of the differential entropy, V_{emp}, as a measure of the number of bits required to code each coefficient. Table 3.1 compares the empirical estimate of the differential entropy of the coefficients obtained using the nonlinear ICA algorithm in the time domain and the cepstral domain to the empirical estimate of the differential entropy of MFCC and LPCC coefficients. The table shows that coefficients that are generated by nonlinear ICA can be more efficiently coded than MFCC and LPCC coefficients.

Table 3.1 An estimate of the differential entropy of the features per coefficient.

Acoustic Features	**Average Number of Bits**
ICA in Cepstral Domain	1.52
ICA in Time Domain	1.64
MFCC	1.77
LPCC	1.85

Another important feature of the output coefficients that are generated by nonlinear ICA is the sparseness of the output feature set. Sparseness is related to reducing the redundancy in the representation of the input signal. Given a dictionary of basis functions $S_1(u), S_2(u), \cdots, S_m(u)$, sparse approximation techniques seek an approximation of a function $g(\mathbf{u})$ as a linear combination of the smallest number of elements of the dictionary, that is, an approximation of the form

$$f_{\mathbf{w}}(\mathbf{u}) = \sum_{q=1}^{m} w_q S_q(\mathbf{u}), \tag{3.52}$$

with the smallest number of nonzero coefficients w_q [108].

The problem can be formulated as minimizing the following cost function

$$E[\mathbf{w}] = D(g(\mathbf{u}), \sum_{q=1}^{m} w_q S_q(\mathbf{u})) + \epsilon \|\mathbf{w}\|_{\ell_0}, \tag{3.53}$$

where D is a cost measuring the distance in some predefined norm between the true function $g(\mathbf{u})$ and our approximation, the ℓ_0 norm of a vector counts the number of elements of that vector which are different from zero, and ϵ is a parameter that controls the trade off between the sparseness and the goodness of the approximation. Unfortunately, it can be shown that minimizing this cost function is NP-hard because of the ℓ_0 norm. Therefore, the ℓ_0 norm is usually approximated by some other kind of norm like the ℓ_2 norm.

In our work, we choose $g(\mathbf{u})$ to be the scalar function in the symplectic mapping that generates the independent components of the input data, and D is taken as the sum of the differential entropy of these output components. We choose also to use the ℓ_2 norm. Comparing this optimization problem with the one we adopted in our algorithm, we find that they are identical and therefore our algorithm is expected to provide a relatively sparse representation of the scalar function $g(\mathbf{u})$. A sparse representation of $g(\mathbf{u})$ does not necessarily imply a sparse representation of the speech signal itself, but the two types of sparseness are related. To evaluate the sparseness of the signal representation using nonlinear ICA, we compare the output coefficients with coefficients of other transforms (MFCC, LPCC) by computing a measure of the sparseness of the feature vector itself.

One of the important measures of the sparseness of the output components is the kurtosis measure defined by

$$K(Y) \quad = \quad E\left[\frac{(Y-\mu_y)^4}{\sigma_y^4}\right] - 3, \tag{3.54}$$

where μ_y is the mean of the random variable Y, and $\sigma_{\mathbf{y}}^2$ is its variance. Kurtosis is proportional to the peakiness of the probability density function of the random variable [109]. The average value of the parameter β of the generalized Gaussian probabilistic model can be used as a measure of the average kurtosis of the output components. In Figure 3.1, the average value of β for the output coefficients that result from processing the time-domain samples of speech is shown as a function of the number of iterations of the algorithm. In Figure 3.2, the average value of β when the speech signal is processed in the cepstral domain is shown as a function of the number of the iterations of the algorithm. The figures show that the nonlinear ICA algorithm tends to converge to output components with high kurtosis and therefore to increase the sparseness of the output coefficients. The figures also show that the nonlinear ICA algorithm with cepstral inputs tends to converge to output components with kurtosis higher than those obtained from nonlinear ICA in the time domain.

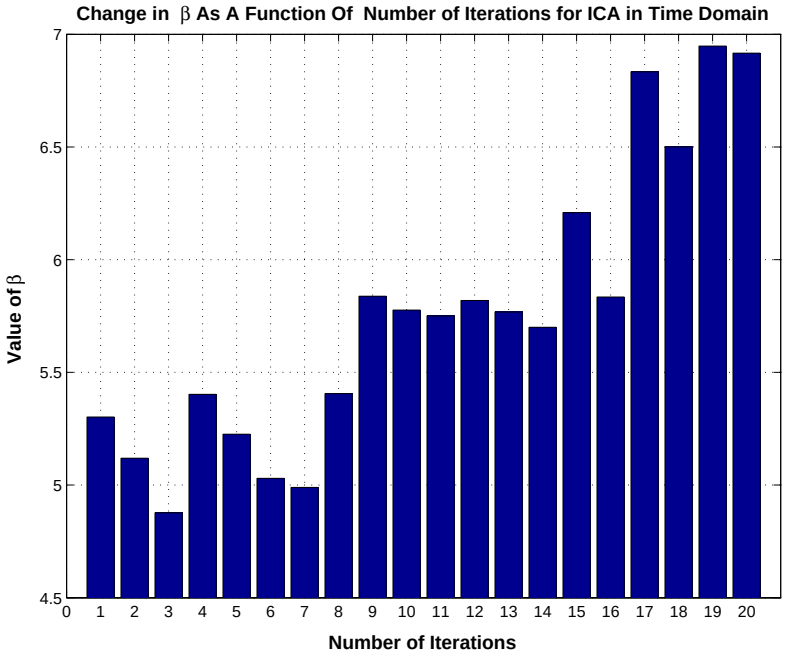

Figure 3.1 The Average Value of β versus Number of Iterations of Nonlinear ICA in Time Domain

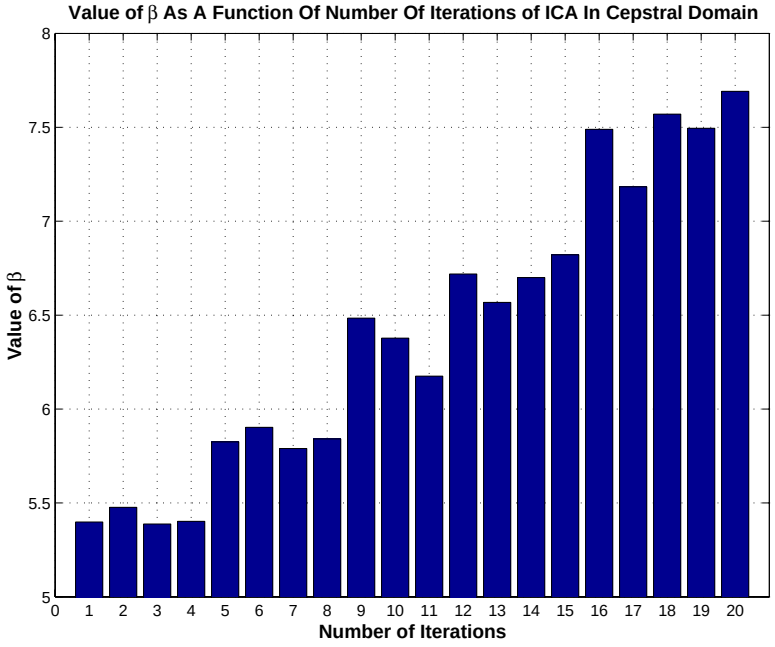

Figure 3.2 The Average Value of β versus Number of Iterations of Nonlinear ICA in Cepstral Domain

3.4.5.2 Recognition accuracy of output coefficients

There are many sources of variability in the speech signal, including linguistic information content, but also including speakers with different dialects and speaking styles, and environmental noise. Most acoustic features that have been successful in speech recognition try to model the speech signal as the convolution of the excitation signal and the vocal tract transfer function, and try to extract the vocal tract transfer function characteristics by linear predictive coding or homomorphic signal processing [1]. If linguistic and nonlinguistic information in the speech signal are independently distributed, nonlinear ICA is capable in principle of learning a mapping that approximately separates them, without the use of an explicit convolutional speech production model. In order to evaluate the success of nonlinear ICA in finding such a mapping, we performed many speech recognition experiments on the TIMIT database. The phoneme recognition accuracy achieved on TIMIT using the SUMMIT segment-based system with different features for different segments and boundaries was 75.6% as reported in [110]. The HMM speech recognizer in [111] used 12 MFCC coefficients, energy and their deltas as the acoustic feature vector. It achieved a 73.7% phoneme recognition accuracy on TIMIT. In [112], the phoneme recognition accuracy for the context-dependent models was 73.8% on TIMIT. A segment-based recognizer that was tested on TIMIT achieved a phoneme recognition accuracy of 69.5% in [10]. A speech recognizer based on recurrent networks achieved phoneme recognition accuracy of 73.4% on TIMIT in [113]. The results reported in this work like all previous results are recognition results that do not use the time alignments provided with the test data. On the other hand, phoneme classification experiments that use this time-alignment data can get classification results on TIMIT up to 81.7% as reported in [110] using heterogeneous measurements.

In our experiments, the 61 phonemes defined in the TIMIT database are mapped to 48 phoneme labels for each frame of speech as described in [112]. These 48 phonemes are collapsed to 39 phonemes for testing purposes as in [112]. A three-state left-to-right model for

each triphone is trained using the EM algorithm. The number of mixtures per state was fixed to five. After training the overall system and obtaining the symplectic map parameters, the approximately independent output coefficients of the symplectic map are used as the input acoustic features to a Gaussian mixture hidden Markov model speech recognizer [114]. The parameters of the recognizer are trained using the training portion of the TIMIT database. The parameters of the triphone models are then tied together using the same approach as in [115].

To compare the performance of the proposed algorithm with other approaches, we generated acoustic features using LDA, linear ICA, and MLLT. We used the maximum likelihood approach to LDA [68] and kept the dimensions of the output of LDA the same as the input. We used also the maximum likelihood approach to linear ICA as described in [48] and briefly overviewed in Chapter 2. Finally, we implemented MLLT as described in [55] and briefly overviewed in Chapter 2. All these techniques used a feature vector that consists of 12 MFCC coefficients, the energy, and their deltas as their input.

Testing this recognizer, using the test data in the TIMIT database, we get the phoneme recognition results in Table 3.2. These results are obtained by using a bigram phoneme language model and by keeping the insertion error around 10% as in [112]. The table compares these recognition results to the ones obtained by MFCC, LDA, linear ICA, and MLLT.

It is clear that searching for the independent components of the speech signal cannot separate information in the speech signal due to linguistic variations from information due to other variations in the time domain.

To improve the phoneme recognition accuracy in the time domain, we used a linear map that maximizes an empirical estimate of the mutual information between the phoneme identities and the output coefficients [116]. These linear maps were used on each component separately and therefore preserved the approximate independence property of the components generated by the nonlinear symplectic map. Using these features, generated by trying

to maximize the mutual information, we trained the previously described HMM recognizer. As shown in Table 3.2, the phoneme recognition accuracy is improved by using this linear map, but it still fails to match the phoneme recognition accuracy achieved by MFCC features.

These results encouraged us to perform the nonlinear independent component analysis on the MFCC coefficients instead of the time-domain signal directly.

Table 3.2 Phoneme recognition accuracy (%) on TIMIT for MFCC features and features generated with ICA, LDA, MLLT, or NICA.

Acoustic Features	**Recognition Accuracy**
MFCC	73.7%
Linear ICA	73.5%
LDA	73.8%
MLLT	74.6%
nonlinear ICA (NICA) in Time Domain	61.2%
NICA in Time Domain After MMI Mapping	64.4%
NICA(Static MFCC) +Energy	68.7%
NICA(Static MFCC) +△NICA+Energy+△Energy	71.2%
NICA(Static MFCC +Energy+△MFCC+△Energy)	75.6%

Three different kinds of experiments were done to test the phoneme recognition results based on the nonlinear ICA coefficients generated with MFCC inputs. First, the 12 cepstrum coefficients were used as the input vector to the nonlinear component analysis algorithm, and the energy was added to the output coefficients. The resultant 13-coefficient feature vector was used to train the HMM recognizer. In the second experiment, we added the delta of the output coefficients and the energy to the acoustic vector that is used in the first experiment. The resultant 26-coefficient feature vectors were used to train the HMM recognizer. Finally, we used the twelve cepstrum coefficients, the energy, and their deltas as the input to the nonlinear independent component analysis algorithm, and used the 26 output coefficients

as the acoustic vector that is used in phoneme recognition. As shown in Table 3.2, the best results were achieved by using the cepstrum coefficients, the energy, and their deltas as the input to the nonlinear symplectic map and using the output of the map as the acoustic feature vector for the phoneme recognizer.

Comparing the phoneme recognition results of the symplectic map in the cepstral domain to the results obtained using the symplectic map on the time-domain data, we find that the features obtained from the mapping of the MFCC features outperform those obtained from the time-domain data. Also, adding the delta coefficients to the MFCC coefficients increases the phoneme recognition accuracy by about 7%. As shown in Table 3.2, the MLLT performed the best among linear transforms with about 0.9% improvement over the MFCC-based feature vector. Comparing these results with the nonlinear ICA algorithm in the cepstral domain, we find that nonlinear ICA outperforms the best linear approach by 1% using the same length of the feature vector.

3.4.5.3 Discussion of the experiments

In this work, we introduced a nonlinear symplectic independent component analysis algorithm. This algorithm can provide the maximum likelihood transform of the features under the independence constraint on the transformed features. This algorithm was applied to the speech signal in two different ways. First, it was applied to the time-domain speech data and the output coefficients' coding efficiency and phoneme recognition accuracy were evaluated. The coding efficiency was found to be improved by this nonlinear mapping compared to MFCC and LPCC coefficients. Our objective function was compared to the objective function of the sparse approximation approaches and the proximity of the two solutions was highlighted. However, the phoneme recognition accuracy based on these coefficients was clearly less than that based on MFCC. This means that blindly searching for the independent components of speech is not enough to be able to extract information correlated to the linguistic information contained in the speech signal, and, in case of the speech signal,

independence is not the best criterion to extract meaningful components of the speech signal that are related to the actual sources of variations. This is, at least in part, because linguistic and nonlinguistic information are not entirely independent.

Second, we applied our algorithm to the MFCC features of the speech signal and its energy. Again, we compared the coding efficiency of the output coefficients to MFCC and LPCC coefficients, and the phoneme recognition accuracy of the output coefficients to LDA, linear ICA, and MLLT. In this case, the coding efficiency is improved also compared to MFCC and LPCC coefficients and even compared to nonlinear ICA on time-domain data. Not only the coding efficiency but also the phoneme recognition accuracy is improved compared to MFCC, LDA, linear ICA, and MLLT. The best phoneme recognition accuracy is achieved when the MFCC, energy, and their deltas are used as input to the nonlinear ICA algorithm. This can be attributed to the ability of the algorithm to find a better representation of the acoustic clues of different phonemes when provided with input features that have proved to be efficient in coding the acoustic information that is related to phonemes. The improvement due to this different representation over the input MFCC features that have the same amount of information about phonemes, is due to the approximate independence property of the new features that allow a more efficient probabilistic modeling of the conditional probabilities with the same model complexity. We can conclude from these results that starting with well-defined features for our goal, like MFCC for phoneme recognition, our nonlinear independent component analysis can provide us with a more sparse representation that improves both the coding efficiency of the coefficients and also the recognition accuracy. The work done here supports the idea that blind information-theoretic approaches for signal analysis cannot replace signal processing techniques tailored for certain application, but it can improve the performance and increase the efficiency if used to augment traditional signal processing techniques.

3.5 The Symplectic Maximum Likelihood Transform

In the previous section, we introduced an iterative algorithm that reduces the mutual information of the features to achieve as approximately independent components as possible. We showed also that by using a volume-preserving map, the problem is reduced to maximizing the likelihood of the output components. In return of the generality of the symplectic map introduced before, we had to use regularization to guarantee the convergence of our algorithm, and we also had to use an optimization algorithm to calculate the output vectors for each iteration of the algorithm. These requirements increase the computational requirements of the algorithm, and therefore limit its applications. In this section, we use an explicit representation of the symplectic map to avoid these problems. Although it is a more restricted representation than the implicit relation used before, the restrictions on the original feature space by this representation are sometimes naturally satisfied as will be shown later.

By using this explicit relationship, we can calculate the output directly without optimization, and we can optimize our objective function directly without adding a regularization term. Not only this, but also the optimization problem now can be solved using a generalized form of the well-known expectation-maximization algorithm [93]. In the following, we will first discuss the explicit representation of the symplectic map, and then provide Lemma 3.2 of Theorem 3.1 to formulate the problem as a maximum likelihood estimation of the parameters of the symplectic map and an HMM model of a variable-length pattern. This very interesting result allows us to jointly optimize the parameters of the symplectic map and the parameters of the recognizer.

3.5.1 An alternative generating function of the symplectic map

Although the previous presentation of the symplectic map reduced the nonlinear ICA problem to the problem of estimating the parameters of a scalar function, its implementation is not computationally efficient due to the implicit definition of the map. This implicit

definition requires solving an optimization problem to calculate the corresponding output components. To solve this problem, we use a reflecting symplectic transformation [117] that uses explicit functions to define the symplectic map. The limitation of this new form is the assumption that the input vectors can be partitioned into two halves. It is common also in applications of symplectic maps in dynamical systems that one half is the derivative of the other half with respect to time. It is interesting that our acoustic feature vector that consists of cepstral coefficient, energy, and their deltas naturally appears in this form. Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$, and $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2)$, with $\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_2 \in \Re^{\frac{n}{2}}$, then the symplectic map can be represented as

$$\mathbf{y}_1 = \mathbf{x}_1 - \frac{\partial V(\mathbf{x}_2)}{\partial \mathbf{x}_2}, \tag{3.55}$$

$$\mathbf{y}_2 = \mathbf{x}_2 - \frac{\partial T(\mathbf{y}_1)}{\partial \mathbf{y}_1}, \tag{3.56}$$

where $V(.)$ and $T(.)$ are two scalar functions that can be chosen arbitrarily. We use two multi-layer feed-forward neural networks to get a good approximation of these scalar functions [100]

$$V(\mathbf{u}, \mathbf{A}, \mathbf{C}) = \sum_{j=1}^{H} c_j S(\mathbf{a}_j \mathbf{u}), \tag{3.57}$$

$$T(\mathbf{u}, \mathbf{B}, \mathbf{D}) = \sum_{j=1}^{H} d_j S(\mathbf{b}_j \mathbf{u}), \tag{3.58}$$

where $S(.)$ is a nonlinear function such as the sigmoid or hyperbolic tangent, $\mathbf{a}_j$ is the jth row of the $H \times n$ matrix $\mathbf{A}$, c_j is the jth element of the $H \times 1$ vector $\mathbf{C}$, $\mathbf{b}_j$ is the jth row of the $H \times n$ matrix $\mathbf{B}$, and d_j is the jth element of the $H \times 1$ vector $\mathbf{D}$. The parameters of

the neural networks and the parameters of the model are jointly optimized to maximize the likelihood of the training data. By using these explicit functions to represent the symplectic map, we no longer need to regularize our objective function, as the map from $\mathbf{X}$ to $\mathbf{Y}$ is explicitly continuous.

3.5.2 Joint optimization of the map and model parameters

We will explain in this section how the parameters of the volume-preserving map and the recognizer model can be jointly optimized to maximize the likelihood of the estimated features. We will assume that the recognizer is HMM-based. However, this approach can be applied to any statistical recognizer. We will give this lemma to account for modeling dynamic patterns with HMM.

Lemma 3.2 *Let* $\mathbf{y}^t = f(\mathbf{x}^t)$ *be an arbitrary one-to-one volume-preserving map of the random vector* $\mathbf{X}^t$ *at time* t *in* $\Re^n$ *to* $\mathbf{Y}^t$ *in* $\Re^n$*, and let* $\hat{P}_{\mathbf{\Lambda}}(\mathbf{y})$ *be the estimated likelihood using an HMM, where* $\mathbf{y} = \mathbf{y}^1 \cdots \mathbf{y}^t \cdots \mathbf{y}^T$*, and* T *is the length of the pattern. The map* $f^*(.)$ *and the set of HMM parameters* $\mathbf{\Lambda}^*$ *jointly minimize the relative entropy between the hypothesized and the true likelihoods of* $\mathbf{Y}$ *if and only if they also maximize the expected log likelihood based on the model* $E_{P(\mathbf{Y})}[\log \hat{P}_{\mathbf{\Lambda}}(\mathbf{Y})]$.

Define $\mathbf{\Phi}^k = (\mathbf{\Lambda}^k, \mathbf{W}^k)$ to be the set of the recognizer parameters, $\mathbf{\Lambda}^k$, and the symplectic parameters, $\mathbf{W}^k$, at iteration k of the algorithm. Using the EM algorithm, the auxiliary function [118] to be maximized, with respect to $\mathbf{\Phi}^{k+1}$, is

$$Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1}) \;=\; E_{\hat{P}(\xi|\mathbf{Y},\mathbf{\Phi}^k)}[\log \hat{P}(\mathbf{y}, \zeta|\mathbf{\Phi}^{k+1})|\mathbf{y}, \mathbf{\Phi}^k], \tag{3.59}$$

where $\zeta \in \xi$ is the state sequence corresponding to the sequence of observations $\mathbf{x} \in \Re^{n \times T}$ that are transformed to the sequence $\mathbf{y} \in \Re^{n \times T}$, and T is the sequence length in frames. In this case, the hidden variables for the EM algorithm are the HMM states, ζ^t for $1 \leq t \leq T$, and the complete data is the set of features and HMM states $(\mathbf{y}^t, \zeta^t)$ at each instance t. The transformed features $\mathbf{y}^t$ are observable variables as they are obtained from the observed feature vector $\mathbf{x}^t$ by an invertible transformation $\mathbf{y}^t = f(\mathbf{x}^t)$. The auxiliary function can be written as

$$Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1}) = \sum_{\zeta \in \xi} \frac{\hat{P}(\mathbf{y}, \zeta | \mathbf{\Phi}^k)}{\hat{P}(\mathbf{y} | \mathbf{\Phi}^k)} \log \hat{P}(\mathbf{y}, \zeta | \mathbf{\Phi}^{k+1}). \tag{3.60}$$

Given a particular state sequence ζ, $\hat{P}(\mathbf{y}, \zeta | \mathbf{\Phi}^k)$ can be written as

$$\hat{P}(\mathbf{y}, \zeta | \mathbf{\Phi}^k) = \pi_{\zeta^0} \prod_{t=1}^{T} \hat{P}(\zeta^t | \zeta^{t-1}, \mathbf{\Phi}^k) \hat{P}(\mathbf{y}^t | \zeta^t, \mathbf{\Phi}^k), \tag{3.61}$$

where π_{ζ^0} is the probability of starting the sequence in state ζ^0, $\hat{P}(\zeta^t | \zeta^{t-1}, \mathbf{\Phi}^k)$ is the state transition probability from ζ^{t-1} to ζ^t given the current parameters $\mathbf{\Phi}^k$, and $\hat{P}(\mathbf{y}^t | \zeta^t, \mathbf{\Phi}^k)$ is the probability of the observation vector $\mathbf{y}^t \in \Re^n$ given the state ζ^t and the current parameters $\mathbf{\Phi}^k$.

Then, the auxiliary function becomes

$$Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1}) = \sum_{\zeta \in \xi} \frac{\hat{P}(\mathbf{y}, \zeta | \mathbf{\Phi}^k)}{\hat{P}(\mathbf{y} | \mathbf{\Phi}^k)} \left(\log \pi_{\zeta^0} + \sum_{t=1}^{T} \left(\log \hat{P}(\zeta^t | \zeta^{t-1}, \mathbf{\Phi}^{k+1}) + \log \hat{P}(\mathbf{y}^t | \zeta^t, \mathbf{\Phi}^{k+1}) \right) \right). \tag{3.62}$$

The updating equations for the HMM parameters based on this formulation are the same as mentioned in [85], and therefore will not be derived here. To calculate the updating

equations of the symplectic parameters, we note that

$$\sum_{\zeta \in \xi} \frac{\hat{P}(\mathbf{y}, \zeta | \mathbf{\Phi}^k)}{\hat{P}(\mathbf{y} | \mathbf{\Phi}^k)} \sum_{t=1}^{T} \log \hat{P}(\mathbf{y}^t | \zeta^t, \mathbf{\Phi}^{k+1}) = \sum_{l=1}^{L} \sum_{t=1}^{T} \frac{\hat{P}(\mathbf{y}, \zeta^t = l | \mathbf{\Phi}^k)}{\hat{P}(\mathbf{y} | \mathbf{\Phi}^k)} \log \hat{P}(\mathbf{y}^t | \zeta^t = l, \mathbf{\Phi}^{k+1}), \tag{3.63}$$

where L is the total number of states.

Therefore, the derivative of the auxiliary function with respect to y_j for $j = 1, 2, \cdots, n$ is given by

$$\frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial y_j} = \sum_{l=1}^{L} \sum_{t=1}^{T} \frac{\hat{P}(\mathbf{y}, \zeta^t = l | \mathbf{\Phi}^k)}{\hat{P}(\mathbf{y} | \mathbf{\Phi}^k)} \frac{\partial \log \hat{P}(\mathbf{y}^t | \zeta^t = l, \mathbf{\Phi}^{k+1})}{\partial y_j}. \tag{3.64}$$

If a mixture of densities is used to model each state, then the derivative of the auxiliary function becomes

$$\frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial y_j} = \sum_{l=1}^{L} \sum_{m=1}^{K_l} \sum_{t=1}^{T} \frac{\hat{P}(\mathbf{y}, \zeta^t = l, \rho_{\zeta^t t} = m | \mathbf{\Phi}^k)}{\hat{P}(\mathbf{y} | \mathbf{\Phi}^k)} \frac{\partial \log \hat{P}(\mathbf{y}^t | \zeta^t = l, \rho_{\zeta^t t} = m, \mathbf{\Phi}^{k+1})}{\partial y_j}, \tag{3.65}$$

where $\rho_{\zeta^t t}$ is the mixture component at time t in the mixture of the state ζ^t, and K_l is the number of densities in each mixture.

These equations are written for one input sequence of observations, and a summation over all training patterns, i.e., sequences of observations, is excluded to simplify the equations. Since the update equations for the symplectic parameters do not need to explicitly mention the structure of the recognizer, we will merge the summation over all states and densities to a summation over densities. These reductions are only to improve the tractability of the following equations and have no effect on the derivation. After modifying the notation,

$$\frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial y_j} = \sum_{t=1}^{T} \sum_{m=1}^{K} \frac{\hat{P}(\mathbf{y}, m|\mathbf{\Phi}^k)}{\hat{P}(\mathbf{y}|\mathbf{\Phi}^k)} \frac{\partial \log \hat{P}(\mathbf{y}^t|m, \mathbf{\Phi}^{k+1})}{\partial y_j}, \tag{3.66}$$

where K is the total number of Gaussian PDFs in all HMM states.

We will assume that the recognizer models the conditional PDF of the observation as a mixture of diagonal-covariance Gaussians and therefore

$$\frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial y_j} = \sum_{t=1}^{T} \sum_{m=1}^{K} \frac{\hat{P}(\mathbf{y}, m|\mathbf{\Phi}^k)}{\hat{P}(\mathbf{y}|\mathbf{\Phi}^k)} \frac{(\mu_{mj} - y_j^t)}{\sigma_{mj}^2}, \tag{3.67}$$

where μ_{mj} and σ_{mj}^2 are the mean and the variance of the jth element of the mth PDF, respectively.

In the following, we will derive the updating equation for the four sets of parameters used in the symplectic map, namely $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and $\mathbf{D}$. Let the nonlinear function used in both feed-forward neural networks be the hyperbolic tangent as stated before. Starting with $\mathbf{A}$ and $\mathbf{B}$, to calculate the update equation for a symplectic parameter a_{qr} and b_{qr} for $q = 1, 2, \cdots, H$, and for $r = 1, 2, \cdots, \frac{n}{2}$, we have to calculate the partial derivative of the auxiliary function with respect to these parameters. These partial derivatives are related to the partial derivatives of the auxiliary function with respect to the features by the following relation:

$$\frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial a_{qr}} = \sum_{j=1}^{\frac{n}{2}} \frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial y_{1j}} \frac{\partial y_{1j}}{\partial a_{qr}} + \sum_{j=1}^{\frac{n}{2}} \frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial y_{2j}} \frac{\partial y_{2j}}{\partial a_{qr}}, \tag{3.68}$$

and

$$\frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial b_{qr}} = \sum_{j=1}^{\frac{n}{2}} \frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial y_{2j}} \frac{\partial y_{2j}}{\partial b_{qr}}, \tag{3.69}$$

where

$$\frac{\partial y_{1j}}{\partial a_{qr}} = \begin{cases} 2x_{2r} \sum_{h=1}^{H} (c_h a_{hj} S(\mathbf{a}_h \mathbf{x}_2)[1 - S^2(\mathbf{a}_h \mathbf{x}_2)]) & \text{for } r \neq j \\ 2x_{2r} \sum_{h=1}^{H} (c_h \mathbf{a}_{hj} S(\mathbf{a}_h \mathbf{x}_2)[1 - S^2(\mathbf{a}_h \mathbf{x}_2)]) - c_q[1 - S^2(\mathbf{a}_q \mathbf{x}_2)] & \text{for } r = j \end{cases} \tag{3.70}$$

$$\frac{\partial y_{2j}}{\partial a_{qr}} = \sum_{k=1}^{\frac{n}{2}} \frac{\partial y_{1k}}{\partial a_{qr}} \frac{\partial y_{2j}}{\partial y_{1k}}, \tag{3.71}$$

$$\frac{\partial y_{2j}}{\partial y_{1k}} = -\sum_{h=1}^{H} \left(d_h b_{hj} b_{hk} S(\mathbf{b}_h \mathbf{y}_1)[1 - S^2(\mathbf{b}_h \mathbf{y}_1)] \right), \tag{3.72}$$

and

$$\frac{\partial y_{2j}}{\partial b_{qr}} = \begin{cases} 2y_{1r} \sum_{h=1}^{H} (c_h b_{hj} S(\mathbf{b}_h \mathbf{y}_1)[1 - S^2(\mathbf{b}_h \mathbf{x}_2)]) & \text{for } r \neq j \\ 2y_{1r} \sum_{h=1}^{H} (c_h b_{hj} S(\mathbf{b}_h \mathbf{y}_1)[1 - S^2(\mathbf{b}_h \mathbf{x}_2)]) - d_q[1 - S^2(\mathbf{b}_q \mathbf{y}_1)] & \text{for } r = j \, . \end{cases} \tag{3.73}$$

For $\mathbf{C}$ and $\mathbf{D}$, the derivation will follow the same procedure, but the resulting equations are much simpler. The partial derivative of the auxiliary function with respect to the sym-

plectic parameter c_q and d_q for $q = 1, 2, \cdots, H$, are related to the partial derivatives of the auxiliary function with respect to the features by the following relation:

$$\frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial c_q} = \sum_{j=1}^{\frac{n}{2}} \frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial y_{1j}} \frac{\partial y_{1j}}{\partial c_q} + \sum_{j=1}^{\frac{n}{2}} \frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial y_{2j}} \frac{\partial y_{2j}}{\partial c_q}, \tag{3.74}$$

and

$$\frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial d_q} = \sum_{j=1}^{\frac{n}{2}} \frac{\partial Q(\mathbf{\Phi}^k, \mathbf{\Phi}^{k+1})}{\partial y_{2j}} \frac{\partial y_{2j}}{\partial d_q}, \tag{3.75}$$

where

$$\frac{\partial y_{1j}}{\partial c_q} = a_{qj}[1 - S^2(\mathbf{a}_q \mathbf{x}_2)], \tag{3.76}$$

$$\frac{\partial y_{2j}}{\partial c_q} = \sum_{k=1}^{\frac{n}{2}} \frac{\partial y_{1k}}{\partial c_q} \frac{\partial y_{2j}}{\partial y_{1k}}, \tag{3.77}$$

and

$$\frac{\partial y_{2j}}{\partial d_q} = b_{qj}[1 - S^2(\mathbf{b}_q \mathbf{y}_1)]. \tag{3.78}$$

To update the symplectic parameters in each iteration, the symplectic parameters that maximize the likelihood can be estimated at each iteration using gradient based optimization algorithms. Equations (3.68)-(3.78) can be used for updating the symplectic parameters iteratively until the value of the likelihood is maximized.

The steps of the generalized EM iterative algorithm to update the symplectic parameters and the HMM parameters are as follows:

1. Initialize the symplectic parameters and the HMM parameters.
2. Calculate the transformed feature vectors $\mathbf{y}$ using the current symplectic maps and the input feature vectors as in Equations (3.55) and (3.56).
3. Using the current value of the parameters $\mathbf{\Phi}^k$, estimate the auxiliary function.
4. Using the current HMM parameters, estimate the symplectic parameters that maximize the auxiliary function by using a gradient-based optimization algorithm.
5. Update the transformed feature vectors $\mathbf{y}$ using the current symplectic maps and the input feature vectors as in Equations (3.55) and (3.56).
6. Estimate the HMM parameters that maximize the auxiliary function using the current symplectic parameters.
7. Iterate (starting from 3) until convergence.

In our experiments, we used the conjugate gradient algorithm to update the symplectic parameters at each iteration. The computational complexity of updating the symplectic parameters using the conjugate gradient algorithm is $O(2(n+1)HN + nH^2N)$, which compares favorably to $O(n^2N)$ for linear approaches for large n, where n is the dimension of the feature vector, H is the number of hidden units in the neural network, and N is the number of feature vectors in the training data.

3.5.3 Experiments and results

We will apply the symplectic maximum likelihood transform (SMLT) to two different problems of high-dimensional probabilistic model estimation. The first is the estimation of the

joint PDF of an example of order statistics, and the second is the estimation of the joint PDF of the Mel-frequency cepstrum coefficients of a speech utterance using Gaussian mixture hidden Markov model as the hypothesized probabilistic model. In the first set of experiments, we compare the likelihood obtained at each iteration to the likelihood obtained without using any transformation of the measurements, and the likelihood obtained by using maximum likelihood linear transformation (MLLT) of the measurements with all methods having approximately the same number of total parameters. In the second set of experiments, the phoneme recognition accuracies obtained by the three methods are compared. In both sets of experiments, the conjugate gradient algorithm was used to update the symplectic parameters in each iteration. The number of hidden nodes of the neural network used in constructing the symplectic map is three in all experiments. Therefore, the total number of symplectic parameters in each experiment is $3n + 6$, where n is the dimension of the feature vector. In all experiments, initializing the symplectic parameters by very small values compared to the dynamic range of the original features gave the best results that are reported here.

3.5.3.1 Order statistics

Order statistics are important features that are usually used in classification and coding. Examples of order statistics are the five largest wavelet coefficients, or the median of a given set of values. The joint distribution of a collection of order statistics obtained from a set of i.i.d. random variables can be calculated exactly given the probability density function of these random variables [119]. Given N realizations of the random vector $\mathbf{x}$ of length n with $\{x_i\}_{i=1}^{n}$ being iid random variables, let $y_i = G(x_i)$. Define $\mathbf{y} = [y_1 \cdots y_n]'$. Let $\mathbf{z} = [z_1 \cdots z_M]'$ be obtained from $\mathbf{y}$ by sorting into ascending order and selecting the first M values. Let $C_{Y_i}(y_i)$ and $P_{Y_i}(y_i)$ be the cumulative distribution function (CDF) and PDF of y_i $\forall i$, respectively. Then, the joint PDF of $\mathbf{Z}$ is given by

$$P_{\mathbf{Z}}(z_1, z_2, \cdots, z_M) = \frac{N!}{(N-M)!} \prod_{i=1}^{M} P_{Y_i}(z_i)[1 - C_{Y_i}(z_M)]^{(N-M)}. \tag{3.79}$$

In this experiment, we generated a set of N iid realizations of Gaussian random vectors $\{\mathbf{x}^j\}_{j=1}^N$ of length $n = 100$ with zero mean and identity covariance matrix, and transformed each component to $y_i^j = |x_i^j|$. After sorting the one hundred transformed components of each random vector in ascending order, we took the first 30 components, i.e., $M = 30$. These 30 components of each realization were used to estimate the symplectic parameters and the parameters of a Gaussian mixture (GM) probabilistic model of the joint probability density function of these 30 components. The parameters are estimated to maximize the likelihood of the training data using the algorithm described before. The log likelihood of the training data using (SMLT+GM) is compared to the log likelihood achieved using the (MLLT+GM) approach as described in [55] and discussed briefly in Chapter 2, and to the log likelihood achieved using the EM algorithm to train a Gaussian mixture model using the same data without transformation (GM). The hidden variables in this experiment are the identity of the Gaussian PDF in the mixture. The Gaussian mixture model in the three methods is initialized using the Linde-Buzo-Gray (LBG) algorithm [120]. The MLLT transform was initialized with a matrix very close to the identity matrix by using very small off-diagonal values. The symplectic parameters are initialized by very small values compared to the dynamic range of the original features. We considered four other random initializations for the MLLT and the SMLT transforms, and the resulting log likelihoods were the same as or less than those reported here for both methods. The number of training vectors N was chosen to be equal to 2×10^7. The comparison of the three methods is shown in Figure 3.3. The figure shows significant increase in the log likelihood by using the symplectic map. Since an increase in the likelihood can be achieved by increasing the number of parameters of the model, e.g., by increasing the number of Gaussian densities in the mixture, a comparison of

the number of parameters used in each method is provided in Table 3.3. The table shows that the increase in the likelihood using SMLT is achieved using fewer parameters than both GM and MLLT. To compensate for the additional number of transformation parameters needed by SMLT and MLLT, we used a different number of Gaussian PDFs in the mixture for each method. The number of Gaussian PDFs used by each method is provided in Table 3.4.

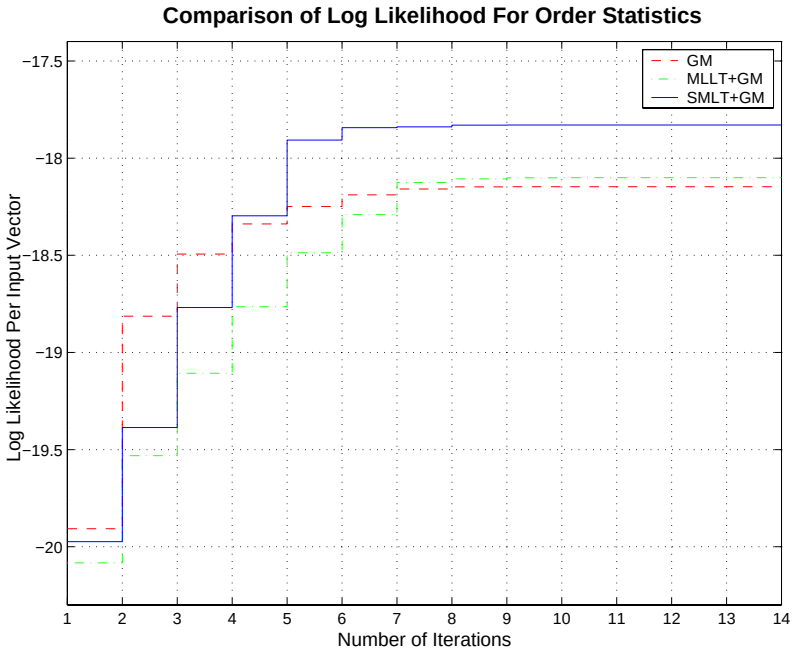

Figure 3.3 Comparison of Log Likelihoods for Order Statistics

Table 3.3 Total number of parameters for each method.

Method	**Number of Parameters**
GM	1952
MLLT+GM	1937
SMLT+GM	1926

Table 3.4 Number of Gaussian PDFs in the mixture for each method.

Method	Number of PDFs
GM	32
GM+MLLT	17
GM+SMLT	30

3.5.3.2 Modeling of dynamic patterns using HMM

To test the performance of our approach on modeling patterns of variable length, we take the speech signal as an example. Most speech recognition systems use a Gaussian mixture HMM-based recognizer and use the Mel-frequency cepstrum coefficients (MFCC) and their deltas as the input acoustic features to the recognizer [25]. In our experiments, the 61 phonemes defined in the TIMIT database are mapped to 48 phoneme labels for each frame of speech as described in [112]. A three-state left-to-right model of each phoneme is trained using the EM algorithm. The number of mixtures per state ranged from 4 to 13 based on the number of frames of training data assigned to the state. The SMLT approach is applied to an input feature vector that consists of 12 MFCC coefficients, energy, and their deltas. These acoustic features are calculated for the whole training subset of the TIMIT database, and the parameters of the symplectic map and the HMM models are jointly optimized to maximize the likelihood as described in the previous section. The SMLT and MLLT transforms are initialized the same way as in the previous set of experiments. We considered four other random initializations for the MLLT, and the SMLT transforms and the resulting phoneme recognition accuracies were the same as or less than those reported here for both methods. The parameters of the triphone models are tied together using the same approach as in [115].

The phoneme recognition results and the total number of parameters for the three methods are provided in Table 3.5. It shows an improvement in the recognition accuracy using the SMLT approach as compared to MLLT and the baseline system. This improvement is significant compared to previous phoneme recognition results on the TIMIT database [88].

Table 3.5 Phoneme recognition accuracy (%) on TIMIT for MFCC features and features generated by MLLT or SMLT.

Recognizer	Recognition Accuracy	Number of Parameters
MFCC+GM	73.7%	25407324
MLLT+GM	74.6%	25407311
SMLT+GM	75.6%	25407302

3.5.4 Discussion

A useful special case of the model enforcement approach is that of the symplectic maximum likelihood transform (SMLT), in which a volume-preserving map is optimized jointly with the model parameters to minimize the relative entropy. A computationally efficient EM-based iterative algorithm for SMLT optimization is described. This iterative algorithm was applied to two important statistical modeling problems: estimation of the joint PDF of order statistics using a Gaussian mixture, and modeling the MFCC coefficients of the speech signal using an HMM. In the first application, an improvement in the log likelihood is achieved using the SMLT approach compared to MLLT and compared to using the original features. This improvement is achieved with a total number of parameters less than other methods in both cases. Phoneme recognition experiments also show significant improvement in recognition accuracy achieved by SMLT compared to the other two methods.

The model enforcement approach is intended to provide a general framework for many interesting feature transformations to reduce inaccuracy of statistical models. This section provides two example applications; several other special cases can be defined by the choice of the parametric form of the map, constraints on the determinant of its Jacobian matrix, and the form of the parameterized likelihood function. The choice of a certain solution is related to the complexity of the problem and the nature of the features used in the system. The main advantage of this general formulation is the avoidance of strict assumptions about the features or the model as in previous approaches.

3.6 Large-Vocabulary Conversational Speech Recognition Using SMLT

In this section, we test the performance of the SMLT on two large-vocabulary, conversational speech recognition tasks: IBM's Superhuman test [121] and the DARPA 2003 Rich Transcription (RT03) test. Conversational speech recognition is a significantly more difficult task than TIMIT phone recognition. Also, the current work uses features computed by a linear projection of spliced cepstra, while the earlier work used static and delta MFCC features. As will be discussed further, this is an important difference because the current implementation of SMLT imposes a partition of the input feature space into two half-spaces. A natural partition exists for static and delta features, but no such partition exists for the projected features.

3.6.1 Experiments

We tested the SMLT in a number of configurations on two large-vocabulary, conversational tasks: IBM's Superhuman test [121] and the DARPA 2003 Rich Transcription (RT03) test. The Superhuman test comprises data from five sources of conversational American English, namely the Switchboard portion of the 1998 Hub 5e test (*swb98*), one meeting from the ICSI meeting corpus [122] (*mtg*), two collections of call center data (*cc1* and *cc2*), and the test set from the IBM Voicemail corpus [123] (*vm*). The RT-03 test material is two-party telephone conversations, like the *swb98* portion of the Superhuman test, but some of the material was collected more recently, and it is about three times longer than *swb98*.

The raw features for the recognition system used in the tests were 18-dimensional MFCC features computed every 10 ms from 25-ms frames with a Mel filter bank that spanned 0.125–3.8 kHz. The recognition features were computed from the raw features by splicing together nine frames of raw features (± 4 frames around the current frame), projecting the 162-dimensional spliced features to 60 dimensions using an LDA projection, and then option-

ally transforming the 60-dimensional projected features with one or more transforms intended to reduce the mismatch between the statistics of the final features and the constraints of the diagonal-covariance Gaussian mixtures that model the HMM observation densities. We tested four different configurations of the LDA projection and subsequent transforms:

L the LDA projection alone,

L+S the LDA projection followed by a nonlinear SMLT,

L+M the LDA projection followed by a linear MLLT, and

L+M+S the LDA projection, then a linear MLLT, then a nonlinear SMLT.

We tested these configurations in order to answer two questions. First, because the SMLT is a nonlinear transform, will an LDA+SMLT cascade match or improve on the performance of an LDA+MLLT cascade? Second, will an LDA+MLLT+SMLT cascade outperform an LDA+MLLT cascade, given the flexibility that the nonlinear SMLT offers?

The acoustic model training data were 315 hours of material from the Switchboard, Switchboard Cellular, and Callhome English corpora. For all five feature sets, an acoustic model comprising 4807 context-dependent states and 156-K diagonal-covariance Gaussian mixtures was used. The states were clustered using decision trees that could ask questions about phone identity within the current word in a ± 5-phone window. The number of Gaussian mixtures assigned to a state was chosen by maximizing the Bayesian Information Criterion (BIC). The decision trees and allocation of mixture components to states were based on the **L+M** feature space.

The Superhuman test was run using an interpolation of four back-off trigram language models (LMs) using modified Kneser-Ney smoothing. The data used to train the four component LMs were 3-M words from Switchboard, 160-M words from Broadcast News, 1-M words from Voicemail, and 600-K words of call center data [121]. The RT03 test was run using an interpolation of four back-off 4-gram LMs using modified Kneser-Ney smoothing.

Table 3.6 Word error rates (%) on the IBM Superhuman test data and the RT-03 test data for features generated with an LDA transform (L), LDA+SMLT transform (L+S), LDA+MLLT transform (L+M), or LDA+MLLT+SMLT transform (L+M+S).

	Superhuman						
transform	swb98	mtg	cc1	cc2	vm	all	RT03
L	47.6	58.0	68.1	48.5	40.2	52.5	–
L+S	46.9	57.4	68.2	47.9	39.3	51.9	–
L+M	43.8	51.7	65.6	41.7	35.4	47.6	39.9
L+M+S	43.5	51.8	65.6	41.4	35.4	47.5	39.7

The component LMs were trained on 3-M words of Switchboard, 58-M words of web data collected and distributed by the University of Washington, 3-M words of Broadcast News relevant to Switchboard topics, and 7-M words from the English Gigaword corpus [124]. Decoding was done using a Viterbi decoder operating on a statically compiled decoding graph and employing a hierarchical Gaussian acoustic model [124].

3.6.2 Results and discussion

The results for our tests of the various transform configurations on the Superhuman and RT03 tests are presented in Table 3.6. A comparison of the **L**, **L+S**, and **L+M** results shows that in almost all cases, the use of an SMLT or MLLT transform improves performance over using only the LDA projection, and that the LDA+MLLT cascade consistently outperforms the LDA+SMLT cascade. This can be partially attributed to the fact that MLLT has roughly 20 times more parameters than the current implementation of SMLT. It should also be noted that the LDA solution is invariant to full-rank linear transforms such as the MLLT, but that no such invariance exists for nonlinear transforms such as the SMLT. A comparison of the **L+M** and **L+M+S** results shows a small advantage for the LDA+MLLT+SMLT cascade, especially on the *swb98* and RT03 tasks — tasks that are well matched to the training data. A number of factors may account for the relatively small improvement obtained with the SMLT: (1) the limited number of parameters in the SMLT, (2) the lack of a natural partition

of the LDA+MLLT feature space into two half-spaces (recall that the implementation used for the reflecting symplectic transform imposes a partition of the feature space), and (3) optimization of the decision trees and mixture allocation to the LDA+MLLT feature space.

3.7 Global versus Local Maps

We will consider here the problem we touched on in the introduction of this chapter, which is the level at which the inaccurate model problem should be solved. We can think of the overall HMM recognizer, for example, as an estimator of a single PDF that describes the likelihood of the utterance, and the assumptions of the HMM like the Markovian property and independence of current observations on all previous and coming states as constraints on this PDF. In this case, the observations corresponding to this utterance should be mapped to a new feature space that better satisfies these constraints. On the other extreme, we can consider each Gaussian PDF in each state's mixture Gaussian PDF as imposing constraints on the observations that we should find a new feature space to satisfy. In large vocabulary speech recognition systems, there are tens of thousands of Gaussian PDFs, thus this approach is impractical. Between these two extremes, there are many options that we would like to discuss here. But before discussing the advantages and disadvantages of each option, we should discuss a very important issue, namely, the comparison of likelihoods based on different observations. In general, we cannot compare likelihoods based on different observations, but if these observations are generated by volume-preserving maps from the original feature space, then we can compare them without the need for any scaling or normalization. This important advantage is due to the relation

$$P(\mathbf{y})|_{\mathbf{y}=f(\mathbf{x})} \quad = \quad P(\mathbf{x}) \tag{3.80}$$

for any volume-preserving map $\mathbf{y} = f(\mathbf{x})$. Clearly, this relation is valid also for conditional PDFs.

Returning to the number of maps and the level at which the map should be used, we recall the traditional tradeoff between increasing the number of parameters in a model to get good approximation results, and minimizing them to avoid the increase of the computational and storage requirements, and the necessity of a large amount of data to get reliable estimates of these parameters. Also the ability of the model to generalize to unseen data may be affected by increasing the number of the parameters above a certain limit. These issues, like the tradeoff between the approximation error and the generalization error and the complexity of the model, are also application dependent. For example if the differences between the testing data and the training data for a certain application are minimal, then this may favor concentrating on the approximation error and trying to use more parameters in the model. There are three reasonable choices, depending on the task, the size of the available training data, the environment, and many other factors. The first is to try to find a global map that will decrease the error due to the conditional PDFs' constraints. The main advantage of this approach is the smaller number of parameters which we need to optimize, and therefore the smaller size of the required training data. On the other hand, the main disadvantage is that training data belonging to different phonemes or allophones may need different maps to satisfy the model constraints. Given sufficiently large training data, we should expect an improvement in the performance by using class-dependent maps. The second choice is to use different maps for different clusters of phonemes or allophones. Phonemes can be clustered based on different manner of articulation, or based on the divergence between their probabilistic models. Finally, we can use phoneme-dependent maps or allophone-dependent maps, if we have large training data to train thousands of parameters.

CHAPTER 4

CLASS-DEPENDENT FEATURES DESIGN

In statistical classification and recognition problems with many classes, it is commonly the case that different classes exhibit wildly different properties. In this case it is unreasonable to expect to be able to summarize these properties by using features designed to represent all the classes. In contrast, features should be designed to represent subsets that exhibit common properties without regard to any class outside this subset. The value of these features for classes outside the subset may be meaningless, or simply undefined. The main problem, due to the statistical nature of the recognizer, is how to compare likelihoods conditioned on different sets of features to decode an input pattern.

Here we will introduce a class-dependent feature design that uses MLE or discriminative training algorithms like MCE and MMI to design a nonlinear mapping of the original feature space for each class. It should be noted that class here could mean a single state, phoneme, or a cluster of phonemes. The formulation of the approach is completely independent of the class choice. An important property of our approach is the class-dependency in the strong sense, which means that we do not need to define meaningless models to compensate for the likelihood differences. This approach avoids the need of having a conditional probabilistic model for each class and feature type pair, and therefore decreases the computational and storage requirements of using heterogeneous features. We present in this work algorithms to calculate the class-dependent features that generalize the model enforcement approach introduced in Chapter 3 to the case of using class-dependent transforms instead of a global transform. We apply our approach to a hidden Markov model (HMM) automatic speech

recognition (ASR) system. We use a nonlinear class-dependent volume-preserving transformation of the features to optimize the objective function. We test two criteria as our objective function—namely maximum likelihood and maximum conditional mutual information. In Section 4.1, we give a brief overview of previous approaches to using class-dependent features in ASR problems. In Section 4.2, the problem is formulated and a solution based on volume-preserving maps is introduced. A maximum likelihood criterion for optimizing the class-dependent features is discussed in Section 4.3. A maximum conditional mutual information criterion is described in Section 4.4 to jointly estimate the parameters of the feature transform and the parameters of the model. Finally, using a generalized probabilistic decent algorithm is suggested for discriminative training of the parameters of the class-dependent transforms in Section 4.4.

4.1 Introduction

The class-dependent features can be looked at as a method of dimensionality reduction in classification [7]. Unlike other methods of dimensional reduction, it is based on sufficient statistics and results in no theoretical loss of performance. Statistical classifiers lose information necessary for classification and recognition in two ways. The first is due to reducing the given data to a set of features, and the second is due to approximating the true joint PDFs of the features. The former loss decreases as the dimensionality of the features increases, while the latter increases as the dimensionality of the features increases. Class-dependent features avoid this compromise by allowing more information to be kept for a given maximum feature dimension. This is clearly at the expense of increasing the computational requirements of the system. Class-dependent features are motivated by the fact that different classes have different salient characteristics that may require different features.

Many recent speech recognition systems use class-dependent feature streams to achieve more robustness and better performance [58]. Using different feature streams within each

recognizer allows the overall system to benefit from the ability of these streams to reveal complementary information about the original speech signal. The main problem, due to the statistical nature of the recognizer, is how to compare likelihoods conditioned on different sets of features to decode a given utterance [10]. The previous approaches to this problem include model-based approaches and feature-based approaches. In model-based approaches, the problem is solved by either completely abandoning the statistical structure of the recognizer, or by adding extra reference models that have no physical meaning but are used to normalize the likelihoods to be comparable statistically [10]. The main problem with the latter approach is how to train these reference models. They are synthetic entities that have no physical meaning at all, so there have been a variety of suggestions to train these models. They range from taking all other phones in the phone set to train the reference model to taking a very small set of similar phones in the phone set. The feature-based approach restricted the class-dependent features to features generated by class-dependent linear transforms of an original set of features [11].

The majority of previous work on discriminant analysis [68] of acoustic features for speech recognition focused on finding a single projection of the features that maximizes the discrimination among the phonemes. The discrimination among phonemes was traditionally measured by the ratio of the within-class covariance and the between-class covariance as in LDA, and more recently [56], [68], [79], the linear discriminant analysis approaches like LDA and HDA are formulated as maximum likelihood problems. Gales [11] recently focused on generalizing the existing projection approaches to multiple subspace projections and called them multiple LDA and multiple HLDA. Realizing the importance of keeping the likelihoods based on these multiple observations comparable, he generalized the idea of sharing a model of the PDF of the complementary subspaces between different classes in the same cluster. His work, like all previous work on discriminant analysis of acoustic features, is based on linear projection of the original feature space. It shares with all previous work [10, 67] on multiple observations the need for "noise-only" models to represent the PDFs of unimportant

dimensions. In other words, the class-dependence of the observations is in the weak sense.

In the weak sense class-dependency, features have observable values for all classes, but the features and some class variables are conditionally independent given a set of classes [12]. This increases the computational and the storage requirements of the system, and results in the introduction of meaningless models that degrade the performance of the recognizer. Features are said to be class-dependent in the strong sense if they are assumed to be observable only for one class or cluster of classes but undefined for the rest of the classes. The need for class-dependence in the strong sense is motivated not only by the difficulty of building the "noise-only" or the "antiphoneme" models and the possible degradation in performance due to building such explicit models, but also by the research in acoustic phonetics that define distinctive features that can characterize any phoneme [125]. The values of most of these distinctive features are not defined for every phoneme. In other words, these distinctive features are class-dependent in the strong sense. Developing a mathematical framework and algorithms for designing statistical recognition systems that use class-dependent features in the strong sense is one of the first requirements for developing a speech recognition system based on these distinctive features.

In this chapter, a nonlinear strong-sense class-dependent feature transformation for pattern recognition is described. It is applied to an HMM speech recognizer. The feature streams are optimized to minimize an estimate of the relative entropy between the actual conditional likelihood and its estimation based on the model, or the *a posteriori* probability and its estimate using HMM and the language model. We will use here the notion of class-dependent features for ASR to represent using different features for different phonemes or different clusters of phonemes that are constructed using some criterion.

4.2 Problem Formulation

The Bayes classification rule minimizes the probability of error if the underlying distribution of the data is known. In its original format, it assumes that the same features are used for all classes; the Bayes classification rule for a set of classes c_i for $i = 1, 2, \cdots, K$ is

$$\hat{c} = \arg \max_{c \in \{1, \cdots, K\}} P_{C|\mathbf{X}}(c|\mathbf{x}), \tag{4.1}$$

where K is the number of classes, $\mathbf{x}$ is the observation vector, and $P_{C|\mathbf{X}}(c|\mathbf{x})$ is the *a posteriori* probability of the classes. This maximization can be reduced to

$$\hat{c} = \arg \max_{c \in \{1, \cdots, K\}} P_{\mathbf{X}|C}(\mathbf{x}|c)P(c). \tag{4.2}$$

Let us now define a set of functions $\{f_i(.)\}_{i=1}^{K}$ such that $\mathbf{y}_i = f_i(\mathbf{x})$ is an arbitrary one-to-one map of the random vector $\mathbf{X}$ in $\Re^n$ to $\mathbf{Y}_i$ in $\Re^n$.

The relation between the joint class-conditional probability of $\mathbf{X}$ and $\mathbf{Y}_i$ is

$$P_{\mathbf{Y}_i|C}(f_i(\mathbf{x})|c_i) = \frac{P_{\mathbf{X}|C}(\mathbf{x}|c_i)}{|det(\mathbf{J}_{f_i}(\mathbf{x}))|}, \tag{4.3}$$

where $det(\mathbf{J}_{f_i}(\mathbf{x}))$ is the determinant of the Jacobian matrix of the map $f_i(.)$ at $\mathbf{x}$ [96].

Therefore, the Bayesian classification rule for the classifiers that use a set of class-dependent features, $\{\mathbf{y}_i\}_{i=1}^{K}$, becomes

$$\hat{c} = \arg \max_{c \in \{1, \cdots, K\}} P_{\mathbf{Y}_i|C}(f_i(\mathbf{x})|c)P(c) \left|det\left(\mathbf{J}_{f_i}(\mathbf{x})\right)\right|. \tag{4.4}$$

Equation (4.4) shows that we can design strong-sense class-dependent features for any

statistical recognition or classification system by taking the determinant of the Jacobian matrix into consideration in the decision rule.

An important special case that simplifies the decision rule is volume-preserving maps. The Bayesian classification rule for the classifiers that use a set of class-dependent features $\{\mathbf{y}_i\}_{i=1}^K$ generated using a set of volume-preserving maps becomes

$$\hat{c} \;=\; \arg \max_{c \in \{1,\cdots,K\}} P_{\mathbf{Y}_i|C_i}(f_i(\mathbf{x})|c_i)P(c_i). \tag{4.5}$$

This means that the decoding is unaffected by using class-dependent volume-preserving transforms.

4.3 A Maximum Likelihood Approach

To train the parameters of these class-dependent transforms to minimize the relative entropy between the hypothesized and the true likelihood, the following lemma generalizes Theorem 3.1 in Chapter 3 for the case of strong-sense class-dependent features.

Lemma 4.1 *Let* $\mathbf{y}_i^t = f_i(\mathbf{x}^t)$ *for* $i = 1, \cdots, K$ *be arbitrary one-to-one volume-preserving maps of the random vector* $\mathbf{X}^t$ *at time* t *in* $\Re^n$ *to* $\mathbf{Y}_i^t$ *in* $\Re^n$*, and let* $\mathbf{y}^t = \mathbf{y}_i^t$ *if* $c^t = c_i$*,* $\mathbf{y} = \mathbf{y}^1 \cdots \mathbf{y}^t \cdots \mathbf{y}^T$*,* T *is the utterance length in frames, and* $\hat{P}_{\mathbf{\Lambda}}(\mathbf{Y}|C)$ *be the estimated likelihood using an HMM, where* $\mathbf{\Lambda} = \{\mathbf{\Lambda}_i\}_{i=1}^K$*. The set of maps* $\{f_i^*(.)\}_{i=1}^K$ *and the set of parameters* $\{\mathbf{\Lambda}_i^*\}_{i=1}^K$ *jointly minimize the expected relative entropy between the hypothesized and the true likelihood of* $\mathbf{Y}$ *if and only if they also maximize the expected log likelihood based on the model,* $E_{P(\mathbf{Y},C)}[\log \hat{P}_{\mathbf{\Lambda}}(\mathbf{Y}|C)]$.

Proof: The expression for the expected relative entropy after an arbitrary transformation $\mathbf{y}_i^t = f_i(\mathbf{x}^t)$ of the input random vector $\mathbf{X}$ in $\Re^n$ is

$$R(P(\mathbf{Y}|C), \hat{P}_{\mathbf{\Lambda}}(\mathbf{Y}|C)) \;=\; -H(P(\mathbf{Y}|C)) - E_{P(\mathbf{Y},C)}\left[\log\left(\hat{P}(\mathbf{Y}|C)\right)\right], \tag{4.6}$$

where $H(P(\mathbf{Y}|C))$ is the conditional differential entropy of the random vector $\mathbf{Y}$ [90].

The relation between the output conditional differential entropy and the input conditional differential entropy is in general

$$H(P(\mathbf{Y}|C)) \;\leq\; H(P(\mathbf{X}|C)) + \sum_{c=1}^{K} \int_{\Re^n} P(\mathbf{x}, c) \log\left(|det\left(\mathbf{J}_{f_c}(\mathbf{x})\right)|\right) \mathbf{dx}, \tag{4.7}$$

where $P(\mathbf{X}|C)$ is the conditional probability density function of the random vector $\mathbf{X}$, for an arbitrary transformation $\mathbf{y}_i^t = f_i(\mathbf{x}^t)$ of the random vector $\mathbf{X}^t$ in $\Re^n$, with equality if $f_i(\mathbf{x}^t)$ is invertible [96].

Therefore, for a volume-preserving map $\mathbf{y}_i^t = f_i(\mathbf{x}^t)$, the expected relative entropy can be written as

$$R(P(\mathbf{Y}|C), \hat{P}_{\mathbf{\Lambda}}(\mathbf{Y}|C)) \;=\; -H(P(\mathbf{X}|C)) - E_{P(\mathbf{Y},C)}\left[\log \hat{P}_{\mathbf{\Lambda}}(\mathbf{Y}|C)\right]. \tag{4.8}$$

Equation (4.8) proves the lemma. ■

The problem of maximizing $E_{P(\mathbf{Y}|C)}[\log \hat{P}_{\mathbf{\Lambda}}(\mathbf{y}|c)]$ can be solved using efficient algorithms based on the incremental expectation maximization (EM) algorithm. Our approach differs from previous approaches to class-dependent feature transform for speech recognition in two ways. First, the feature transform is not necessarily linear. Second, it is class-dependent in the strong sense: the conditional PDF of each features stream is calculated for the corresponding class only. It should be noted that all linear class-dependent maps are special cases of Lemma 4.1, as any linear map is a volume-preserving map multiplied by a scalar.

We showed that by using a set of volume-preserving maps, the problem of minimizing the relative entropy between the actual class-conditional PDFs and their parametric models is reduced to maximizing the likelihood of the training data in the new strong-sense class-dependent feature spaces. In the following, we use the reflecting symplectic map described in Chapter 3 to generate the new set of features. Using the algorithm described in Section 3.4, the values of the symplectic map parameters can be updated in each iteration using any gradient-based optimization algorithm [126].

4.3.1 Results on the TIMIT database

The symplectic maximum likelihood algorithm described in Chapter 3 is used to study the optimal class-dependent feature space for diagonal-covariance Gaussian mixture HMM modeling of the TIMIT database. The phoneme set is divided into three clusters: silence, vowel-like, and consonants. We associated with each cluster a symplectic map that is trained to maximize the likelihood of the training data that correspond to the phonemes member of the cluster.

The baseline 26-feature vector consists of 12 Mel-frequency cepstrum coefficients (MFCC), energy, and their deltas. In each iteration, the new feature vector is calculated using the current symplectic transformation parameters, then the maximum likelihood estimates of the HMM model parameters are calculated. Then, the maximum likelihood estimates of the symplectic map parameters are calculated using the conjugate-gradient algorithm. After the iterative algorithm converges to a set of locally optimal HMM and symplectic parameters, the training data are transformed by the symplectic map yielding the final symplectic maximum likelihood transform (SMLT) feature vector for each cluster of phonemes.

In our experiments, the 61 phonemes defined in the TIMIT database are mapped to 48 phoneme labels for each frame of speech. These 48 phonemes are collapsed to 39 phoneme for testing purposes as in [112]. A three-state left-to-right model for each triphone is trained.

The number of mixtures per state was varied between 3 and 13 based on the number of observations in the training data assigned to the state. The parameters of the recognizer and the symplectic map are trained using the training portion of the TIMIT database. The parameters of the triphone models are then tied together using the same approach as in [115].

To compare the performance of the proposed algorithm with other approaches, we generated acoustic features using independent component analysis (ICA) and maximum likelihood linear transform (MLLT). We tested both approaches using the same three-cluster categorization of the phoneme set used with SMLT. A cluster-dependent feature vector is designed for each cluster using maximum likelihood ICA and MLLT. We used the linear ICA approach described in [52] and implemented MLLT as described in [55].

Testing this recognizer, using the test data in the TIMIT database, we get the phoneme recognition results in Table 4.1. These results are obtained by using a bigram phoneme language model and by keeping the insertion error around 10%. The table compares SMLT recognition results to the ones obtained by MFCC, ICA, and MLLT.

Table 4.1 Phoneme recognition accuracy (%) on TIMIT for MFCC features and class-dependent features generated by ICA, MLLT, or SMLT.

Acoustic Features	**Recognition Accuracy**
MFCC	73.7%
ICA	73.9%
MLLT	74.5%
SMLT	75.5%

4.3.2 Results on the Superhuman and RT03 databases

We tested the class-dependent SMLT also in a number of configurations on two large-vocabulary, conversational tasks: IBM's Superhuman test [121] and the DARPA 2003 Rich Transcription (RT03) test. The Superhuman test comprises data from five sources of conversational American English, namely the Switchboard portion of the 1998 Hub 5e test (*swb98*),

one meeting from the ICSI meeting corpus [122] (*mtg*), two collections of call center data (*cc1* and *cc2*), and the test set from the IBM Voicemail corpus [123] (*vm*). The RT-03 test material is two-party telephone conversations, like the *swb98* portion of the Superhuman test, but some of the material was collected more recently, and it is about three times longer.

The raw features for the recognition system used in the tests were 18-dimensional MFCC features computed every 10 ms from 25-ms frames with a Mel filter bank that spanned 0.125–3.8 kHz. The recognition features were computed from the raw features by splicing together nine frames of raw features (± 4 frames around the current frame), projecting the 162-dimensional spliced features to 60 dimensions using an LDA projection, and then optionally transforming the 60-dimensional projected features with one or more transforms intended to reduce the mismatch between the statistics of the final features and the constraints of the diagonal-covariance Gaussian mixtures that model the HMM observation densities. We tested three different configurations of the LDA projection and subsequent transforms:

L+M the LDA projection followed by a linear MLLT;

L+M+S the LDA projection, then a linear MLLT, then a nonlinear SMLT; and

L+M+S2 the LDA projection, then a linear MLLT, then two class-dependent SMLTs, one for speech states and one for nonspeech states.

We tested these configurations in order to answer the question: Can we obtain additional improvements in recognition performance with multiple, class-dependent SMLTs, taking full advantage of the SMLT's volume-preserving property?

The acoustic model training data were 315 hours of material from the Switchboard, Switchboard Cellular, and Callhome English corpora. For all three feature sets, an acoustic model comprising 4807 context-dependent states and 156-K diagonal-covariance Gaussian mixtures was used. The states were clustered using decision trees that could ask questions about phone identity within the current word in a ± 5-phone window. The number of Gaussian mixtures assigned to a state was chosen by maximizing the Bayesian Information

Table 4.2 Word error rates (%) on the IBM Superhuman test and the RT-03 test for features generated with LDA+MLLT transform (L+M), LDA+MLLT+SMLT transform (L+M+S), or with an LDA+MLLT transform followed by one of two class-dependent SMLTs (L+M+S2).

	Superhuman						
transform	swb98	mtg	cc1	cc2	vm	all	RT03
L+M	43.8	51.7	65.6	41.7	35.4	47.6	39.9
L+M+S	43.5	51.8	65.6	41.4	35.4	47.5	39.7
L+M+S2	43.5	51.9	65.4	41.4	35.3	47.5	39.7

Criterion (BIC). The decision trees and allocation of mixture components to states were based on the **L+M** feature space.

The Superhuman test was run using an interpolation of four back-off trigram language models (LMs) using modified Kneser-Ney smoothing. The data used to train the four component LMs were 3-M words from Switchboard, 160-M words from Broadcast News, 1-M words from Voicemail, and 600-K words of call center data [121]. The RT03 test was run using an interpolation of four back-off 4-gram LMs using modified Kneser-Ney smoothing. The component LMs were trained on 3-M words of Switchboard, 58-M words of web data collected and distributed by the University of Washington, 3-M words of Broadcast News relevant to Switchboard topics, and 7-M words from the English Gigaword corpus [124]. Decoding was done using a Viterbi decoder operating on a statically compiled decoding graph and employing a hierarchical Gaussian acoustic model [124].

The results for our tests of the various transform configurations on the Superhuman and RT03 tests are presented in Table 4.2. We see no significant improvement with the two class-dependent SMLTs over the **L+M+S** results. This result is consistent with results on the TIMIT database reported here and in [126] for the SMLT, results reported for class-dependent MLLTs in [55], and results reported on multiple LDA (MLDA) and multiple HDA (MHDA) in [11]. We argue that transforms trained using MLE on observations corresponding to specific classes are less likely to reduce recognition error compared to MLE global

transforms and a discriminative criterion should be used to estimate the class-dependent transforms.

4.4 Discriminative Class-Dependent Features Design

To train the parameters of these class-dependent transforms to minimize the expected relative entropy between the hypothesized and the true *a posteriori* probability, the following lemma generalizes Theorem 3.1 in Chapter 3 for the case of strong-sense class-dependent features.

Lemma 4.2 *Let* $\mathbf{y}_i^t = f_i(\mathbf{x}^t)$ *for* $i = 1, \cdots, K$ *be arbitrary one-to-one maps of the random vector* $\mathbf{X}^t$ *at time* t *in* $\Re^n$ *to* $\mathbf{Y}_i^t$ *in* $\Re^n$*, and let* $\mathbf{y}^t = \mathbf{y}_i^t$ *if* $c^t = c_i$*,* $\mathbf{y} = \mathbf{y}^1 \cdots \mathbf{y}^t \cdots \mathbf{y}^T$*,* T *is the utterance length in frames, and* $\hat{P}_{\mathbf{\Lambda}}(c|\mathbf{y})$ *be the a posteriori probability estimated using the HMM and the language model, where* $\mathbf{\Lambda} = \{\mathbf{\Lambda}_i\}_{i=1}^K$*. The set of maps* $\{f_i^*(.)\}_{i=1}^K$ *and the set of parameters* $\{\mathbf{\Lambda}_i^*\}_{i=1}^K$ *jointly minimize the expected relative entropy between the hypothesized and the true a posteriori probability if and only if they also maximize the conditional mutual information between the classes and the features given the model* $I(\mathbf{Y}, C|\mathbf{\Lambda})$*.*

Proof: The expression for the expected relative entropy after an arbitrary transformation $\mathbf{y}_i^t = f_i(\mathbf{x}^t)$ of the input random vector $\mathbf{X}^t$ in $\Re^n$ is

$$R(P(C|\mathbf{Y}), \hat{P}_{\mathbf{\Lambda}}(C|\mathbf{Y})) \;\; = \;\; -H(P(C|\mathbf{Y})) - E_{P(C,\mathbf{Y})}\left[\log\left(\hat{P}_{\mathbf{\Lambda}}(C|\mathbf{Y})\right)\right], \qquad (4.9)$$

where $H(P(C|\mathbf{Y}))$ is the conditional differential entropy of the random variable C [90].

The relation between the output conditional differential entropy and the input conditional differential entropy is in general

$$H(P(C|\mathbf{Y})) \;\leq\; H(P(C|\mathbf{X})), \tag{4.10}$$

where $P(c|\mathbf{x})$ is the *a posteriori* probability density function given the random vector $\mathbf{X}$, for an arbitrary transformation $\mathbf{y}_i^t = f_i(\mathbf{x}^t)$ of the random vector $\mathbf{X}^t$ in $\Re^n$, with equality if $f_i(\mathbf{x}^t)$ is invertible [96].

Therefore, for a one-to-one map $\mathbf{y}_i^t = f_i(\mathbf{x}^t)$, the relative entropy can be written as

$$R(P(C|\mathbf{Y}), \hat{P}_{\mathbf{\Lambda}}(C|\mathbf{Y})) \;=\; -H(P(C|\mathbf{X})) - E_{P(C,\mathbf{Y})}\left[\log\left(\hat{P}_{\mathbf{\Lambda}}(C|\mathbf{Y})\right)\right]. \tag{4.11}$$

But

$$I(\mathbf{Y}, C|\mathbf{\Lambda}) \;=\; E_{P(C,\mathbf{Y})}\left[\log\left(\hat{P}_{\mathbf{\Lambda}}(C|\mathbf{Y})\right) - \log\left(\hat{P}(C)\right)\right]. \tag{4.12}$$

Therefore,

$$\begin{aligned} R(P(C|\mathbf{Y}), \hat{P}_{\mathbf{\Lambda}}(C|\mathbf{Y})) \;=\; & -H(P(C|\mathbf{X})) \\ & -I(\mathbf{Y}, C|\mathbf{\Lambda}) - E_{P(C,\mathbf{Y})}\left[\log\left(\hat{P}(C)\right)\right]. \end{aligned} \tag{4.13}$$

Equation (4.13) proves the lemma. ■

It should be noted that mutual information is invariant to any one-to-one map, and therefore maximizing the conditional mutual information given the model is equivalent to improving our estimate of the mutual information by improving our estimate of the *a posteriori* probability density function using the model as proved by Lemma 4.2. In the previous section, we showed that the decoding is unaffected by using class-dependent features gener-

ated by volume-preserving maps. So in this section, we use the reflecting symplectic map described in Chapter 3 to generate the new set of features. Using the algorithm described in Section 3.4, the values of the symplectic map parameters can be updated in each iteration to maximize the conditional mutual information given the HMM model, $I(\mathbf{Y}, C|\mathbf{\Lambda})$, instead of the likelihood using any gradient-based optimization algorithm [127]. In this case, the HMM parameters are updated using the extended Baum-Welch algorithm [14] to maximize the conditional mutual information.

4.4.1 Results on the TIMIT database

The symplectic maximum conditional mutual information approach is used to study the optimal feature space for diagonal-covariance Gaussian mixture HMM modeling of the TIMIT database. The phoneme set is divided to three clusters: silence, vowel-like, and consonants. We associated with each cluster a symplectic map. The parameters of the symplectic maps are trained to maximize an empirical estimate of the conditional mutual information between the generated features and the HMM states.

The baseline 26-feature vector consists of 12 Mel-frequency cepstrum coefficients (MFCC), energy, and their deltas. In each iteration, the new feature vector is calculated using the current symplectic transformation parameters; then the maximum conditional mutual information estimates of the HMM model parameters are calculated using the extended Baum-Welch algorithm [14]. Then, the maximum conditional mutual information estimates of the symplectic map parameters are calculated using the conjugate-gradient algorithm. After the iterative algorithm converges to a set of locally optimal HMM and symplectic parameters, the training data are transformed by the symplectic map yielding the final symplectic maximum conditional mutual information transform (SMCMI) feature vector for each cluster of phonemes.

In our experiments, the 61 phonemes defined in the TIMIT database are mapped to 48 phoneme labels for each frame of speech. These 48 phonemes are collapsed to 39 phonemes for testing purposes as in [112]. The number of mixtures per state was varied between 3 and 13 based on the number of training observations assigned to the state. The parameters of the recognizer and the symplectic map are trained using the training portion of the TIMIT database.

To compare the performance of the proposed algorithm with other approaches, we generated acoustic features using the class-dependent SMLT approach described in the previous section. Testing this recognizer, using the test data in the TIMIT database, we get the phoneme recognition results in Table 4.3. These results are obtained by using monophone HMM models and a bigram phoneme language model and by keeping the insertion error around 10%. The table compares strong-sense class-dependent SMCMI recognition results to the ones obtained by MFCC, and strong-sense class-dependent SMLT. It shows that the class-dependent symplectic maps trained using an MCMI criterion outperform those trained using a maximum likelihood criterion by 0.5%.

Table 4.3 Phoneme recognition accuracy (%) on TIMIT for MFCC features and class-dependent features generated by SMLT or SMCMI.

Acoustic Features	**Recognition Accuracy**
MFCC	63.1%
SMLT	64.3%
SMCMI	64.8%

4.4.2 Alternative implementation using the GPD algorithm

We will use an iterative algorithm based on the generalized probabilistic descent algorithm (GPD) [15] to estimate the parameters that minimize an empirical estimate of the recognition error that is usually used in MCE-based algorithms.

Based on Bayes decision theory, the actual recognition error is a zero-one function that we cannot use as a criterion to optimize our parameters due to its discontinuity, so the GPD algorithms use a smooth misclassification measure given by

$$e_k(\mathbf{x}, \mathbf{\Lambda}) = \left[\frac{1}{K-1} \sum_{q, q \neq k} \{\log P_{\mathbf{\Lambda}}(C_q|\mathbf{x})\}^{\beta} \right]^{\frac{1}{\beta}} - \log P_{\mathbf{\Lambda}}(C_k|\mathbf{x}), \tag{4.14}$$

where $P_{\mathbf{\Lambda}}(C_k|\mathbf{x})$ is the *a posteriori* probability of the correct class, $P_{\mathbf{\Lambda}}(C_q|\mathbf{x})$ is the *a posteriori* probability of the qth class, and β is a positive constant [15].

Using a set of class-dependent symplectic maps $\{f_k\}_{k=1}^{K}$, let us denote the parameters of each map $f_k(.)$ by $\mathbf{W}_k$, where $\mathbf{W}_k = (\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_k, \mathbf{D}_k)$ as defined in Section 3.5. Using class-dependent observation vectors, the misclassification measure becomes

$$e_k(\mathbf{x}, \mathbf{\Lambda}, \mathbf{W}) = \left[\frac{1}{K-1} \sum_{q, q \neq k} \{\log P_{\mathbf{\Lambda}}(C_q|\mathbf{y}_q)\}^{\beta} \right]^{\frac{1}{\beta}} - \log P_{\mathbf{\Lambda}}(C_k|\mathbf{y}_k), \tag{4.15}$$

where $\mathbf{y}_q = f_q(\mathbf{x})$ and $\mathbf{W} = [\mathbf{W}_1 \mathbf{W}_2 \cdots \mathbf{W}_K]$.

Then a loss function that is a smooth monotonically increasing function of this misclassification measure is defined. We will use the sigmoid function that is usually used in GPD algorithms

$$\ell_k(\mathbf{x}, \mathbf{\Lambda}, \mathbf{W}) = \frac{1}{1 + \exp(-(\alpha e_k(\mathbf{x}, \mathbf{\Lambda}, \mathbf{W})) + \gamma))}, \tag{4.16}$$

for $\alpha > 0$.

Given a set of observations $\Phi = \{\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^N\}$, we can define an empirical average cost as

$$L(\mathbf{\Lambda}, \mathbf{W}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \ell_k(\mathbf{x}^i, \mathbf{W}, \mathbf{\Lambda}) 1(\mathbf{y}^i \in C_k), \tag{4.17}$$

where

$$1(z) = \begin{cases} 0 & \text{if } z \text{ is false} \\ 1 & \text{if } z \text{ is true .} \end{cases}$$

This well-defined cost function can be conveniently minimized by a gradient decent algorithm, using the following adaptation rule:

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \epsilon(t) \nabla L(\mathbf{\Lambda}, \mathbf{W})|_{\mathbf{W}=\mathbf{W}_t}, \tag{4.18}$$

where $\mathbf{W}_t$ denotes the features mapping parameter set in the tth iteration, and $\{\epsilon(t)\}$ is a set of positive numbers that satisfy

$$\sum_{t=1}^{N} \epsilon(t) \to \infty \text{ as } N \to \infty$$

and

$$\sum_{t=1}^{N} \epsilon^2(t) < \infty \text{ as } N \to \infty.$$

The GPD approach guarantees also the convergence of this optimization problem in probability, when we update the parameters adaptively for each new utterance.

We used the same baseline system described before to test the algorithm on the TIMIT database. The phoneme set is divided to three clusters: silence, vowel-like, and consonants. We associated with each cluster a symplectic map, and the parameters of the symplectic maps are trained jointly with the HMM parameters using the MCE criterion after setting $\gamma = 0$,

$\beta = 1$, and $\alpha = 1$. The symplectic parameters are trained using the GPD algorithm over batches of the training data of 1000 frames each, while updating the HMM parameters every 10 iterations. Testing this recognizer, using the test data in the TIMIT database, we get the phoneme recognition results in Table 4.4. These results are obtained by using monophone HMM models and a bigram phoneme language model, and by keeping the insertion error around 10%. The table compares the recognition results of the class-dependent symplectic maps trained using the MCE criterion (SMCE) to the ones obtained by MFCC, and the class-dependent SMLT. The results show that the class-dependent symplectic maps trained using MCE criterion outperform those trained using MLE.

Table 4.4 Phoneme recognition accuracy (%) on TIMIT for MFCC features and class-dependent features generated by SMLT or SMCE.

Acoustic Features	**Recognition Accuracy**
MFCC	63.1%
SMLT	64.3%
SMCE	64.7%

Alternatively, we can use MCMI-based or MCE-based algorithms to optimize $\mathbf{W}$ such that the empirical average cost is minimized, while using the expectation maximization algorithm to optimize the model parameters $\mathbf{\Lambda}$.

CHAPTER 5

DISCRIMINATIVE DIMENSIONALITY REDUCTION AND FEATURES SELECTION

Classification and recognition systems can be divided into two groups: generative (or informative) and discriminative [128]. In generative systems, the classification or recognition is done by examining the likelihood of each class producing the features and assigning it to the most likely class, while in discriminative systems, the focus is on modeling the class boundaries or the class membership probabilities directly. Examples of the former are Fisher discriminant analysis, hidden Markov models (HMM) trained using maximum likelihood estimation, and naive Bayes. Examples of the latter include logistic regression, neural networks, and support vector machines (SVM). Discriminative approaches are known to be robust against errors in structural assumptions [80]. This property arises from a precise match between the training objective and the criterion by which they are subsequently evaluated. On the other hand, generative models deal effectively with uncertain or incomplete examples. Several approaches have been proposed for merging the generative and the discriminative methods. Examples of these approaches are hybrid systems like artificial neural networks and HMM (ANN/HMM) systems [23], and training the generative systems based on a discriminative criterion. For example in speech recognition, HMM is a generative model that can be trained using minimum classification error (MCE) [40] or maximum mutual information (MMI) criterion [13]. In this chapter, we will suggest merging the generative and the discriminative methods by using a generative model for the recognizer, but selecting its

input features based on a discriminative criterion. We will describe also a method for jointly optimizing the features and the recognizer model based on discriminative criteria.

In Chapter 3, we introduced a model enforcement approach to the problem of acoustic feature design for speech recognition. We proved that optimizing the features to satisfy the model constraints will decrease the recognition error. But one may wonder where this original feature space comes from. An important step required before designing features that satisfy the probabilistic model of the recognizer is to provide a compact representation of the raw training data. This step can be based on knowledge-based or data-driven approaches. In experiments on the TIMIT database in Chapter 3, the original feature vector consisted of the MFCC coefficients, energy, and their deltas, which is an example of knowledge-based approaches. In experiments on the Superhuman and RT03 databases in Chapter 3, the original feature vector consisted of LDA generated features, which is an example of data-driven approaches.

In this chapter, we will introduce discriminative approaches to acoustic feature selection and design that try to maximize the conditional mutual information between the features and the speech units or to minimize an empirical estimate of the recognition error.

The first section describes an algorithm for feature selection based on mutual information maximization and its application to selecting an acoustic-features representation of phonological features. In Section 5.2, the algorithm is applied to features selection for phoneme recognition. An interpretation of LDA as a constrained maximum conditional mutual information projection problem is presented in Section 5.3. Finally, a generalization of LDA, by removing the constraints, to the maximum conditional mutual information linear projection algorithm and its application to phoneme recognition are introduced in Section 5.4.

5.1 MMI Acoustic-Features Representation of Phonological Features

This section addresses the problem of finding a subset of the acoustic feature space that best represents a set of phonological factors. A maximum mutual information approach is presented for selecting acoustic features to be combined to represent the distinctions coded by a set of correlated phonological factors. Each set of phonological factors is chosen on the basis of acoustic phonetic similarity, so the sets can be considered approximately independent. This means that the output of recognizers that recognize these sets independently using the acoustic representation achieved by an algorithm presented in this section can be combined together to increase efficiency and robustness of the overall speech recognition system. The mutual information between the phonological factor sets and their achieved acoustic representation is increased by up to 220% over the best single-type acoustic representation in the feature space of the same length.

A framework for defining the theoretically optimal method for feature subset selection was presented in [83]. It proves that for a feature to be unnecessary to model a certain property, it should have a Markov blanket within the complete feature set. However, this optimal feature selection approach is computationally intractable. So we present an algorithm that calculates a good approximation of the acoustic feature subset that has the maximum mutual information with some phonological factors.

In the rest of this section, we will discuss the phonological factors used and the values that can be assigned to them, and how mutual information can be calculated for different acoustic features and phonological sets from the training data. Then, the algorithm used for maximum mutual information selection of the acoustic features to model each factor of speech is described, and the different experiments and results achieved.

5.1.1 Phonological features selection

In this method, the feature stream used within each recognizer is selected from an acoustic feature space formed from LPC cepstrum coefficients, MFCC cepstrum coefficients based on FFT, PLP cepstrum coefficients, energy, their deltas, and their averages. The selection of a fixed-length acoustic feature representation for each phonological factors set is based on maximizing the mutual information between the acoustic feature stream and the corresponding phonological factor set. An algorithm that tries to approximate this maximization within a small number of iterations will be described.

Voicing, manner of articulation, place of articulation, and duration are the main aspects of the speech signal that are selected as factors to be modeled in our system. This selection satisfies two main requirements: that these factors are enough to discriminate among all phonemes of English, and these factors can be assumed to be independent. Each factor can be assigned one of a set of values which is shown in Table 5.1 These values are chosen based

Table 5.1 Phonological factors of speech and their values.

Phonological Factor	**Values**
Voicing	voiced, unvoiced, silence
Manner of Articulation	vowel, nasal, fricative, stop, glide, silence
Place of Articulation	17 combinations of binary features: (round, anterior, distributed,lateral, low, high, back)
Duration	short, medium long

on the set of distinctive features given by Stevens in [125] such that all distinct configurations of different phonemes can be identified.

5.1.2 Maximum mutual information estimation

The mutual information between two random vectors $\mathbf{X}$, representing the phonological set, and $\mathbf{Y}$, representing the acoustic feature, is

$$I(\mathbf{X}, \mathbf{Y}) \quad = \quad E\left[\log \frac{P(\mathbf{y}|x_i)}{P(\mathbf{y})}\right]. \tag{5.1}$$

An empirical estimate of the mutual information is calculated by modeling $P(\mathbf{y}|x_i)$ for $i = 1, 2, \cdots, J$ by a Gaussian mixture probability density function, where J is the number of values that the phonological set can take, x_i is the ith value of the phonological set. The Expectation-Maximization (EM) algorithm is used to calculate the parameters of these density functions using the training data. The number of densities within each mixture varies from 2 to 13 depending on the amount of training data assigned to the specific value of the phonological factor that the probability density function models. An estimate of $P(\mathbf{y})$ is obtained by calculating $\{P(x_i)\}_{i=1}^{J}$ from the training data. Then the mutual information between each acoustic feature available and the phonological factor under consideration is calculated. The expectation in Equation (5.1) is approximated by a summation over all possible values that appear in the training data.

To maximize the mutual information between a phonological set and the subset of acoustic features that are used to model it, we need an intractable amount of computation to test all possible subsets of the acoustic features we have. Hence, an algorithm is developed that guarantees we will achieve a subset with high mutual information but not necessarily the optimal one.

5.1.3 An algorithm for MMI feature selection

This section describes an algorithm for selecting a vector of M acoustic features that provide a relatively large amount of information about some phonological factor r. First, Equa-

tion (5.1) is evaluated for every acoustic feature individually, to find the mutual information between each individual feature and the phonological factor. Second, the M features with the highest individual mutual information scores are combined to form an M-dimensional initial feature vector $\mathbf{V}_r$ describing the phonological factor r. Then, Gaussian mixture models of the different classes of speech based on this phonological factor are built using this feature vector. Then, the average mutual information of the phonological factor set with the corresponding acoustic feature vector is calculated. Also, the mutual information between each acoustic feature used and the phonological factors set is calculated based on the marginal probability density function of this feature. The acoustic features are ordered based on this mutual information values. The worst F features are replaced by the same number of features from the ordered list of features based on their individual mutual information with the phonological factor. The new Gaussian mixture models based on the new feature vector are calculated and the process repeats. When the average mutual information decreases, the F worst features are replaced with the best $\frac{F}{2}$ features that were removed in the previous stage and $\frac{F}{2}$ acoustic features from the ordered list based on mutual information between phonological factor and the individual acoustic feature. The algorithm can be summarized as follows:

For $r = 1$ to H, where H is the number of phonological factor sets, repeat the following steps:

1. For $i = 1$ to N_r,

   Calculate the *a priori* probabilities of different values of phonological factors, $P(x_{ri})$, where N_r is the number of values that the rth phonological factor can have.

2. For $i = 1$ to N_r,

   For $a = 1$ to J,

   Using the EM algorithm, build a Gaussian mixture model of the conditional PDF of the acoustic feature given a certain value of the phonological factor, $P(y_a|x_{ri})$, where

J is the total number of acoustic features under test.

3. For $i = 1$ to N_r,

   For $a = 1$ to J,

   Calculate the mutual information between the phonological factor and the acoustic feature, $I(X_r, Y_a)$, as

$$I(X_r, Y_a) = \sum_{k=1}^{O} \sum_{i=1}^{N_r} \log \frac{P(y_a^k | x_{ri})}{P(y_a^k)},$$

   where O is the number of frames in the training data.

4. Order the acoustic features in descending order based on mutual information calculated in the previous step and save the ordered list L_r.

5. Initialize the acoustic feature vector $\mathbf{V}_r$ with the top M features in L_r.

6. While L_r is not empty, do the following steps:

   (a) For $i = 1$ to N_r,

   Using EM algorithm, build a Gaussian mixture model of $P(\mathbf{V}_r | x_{ri})$.

   (b) For $i = 1$ to N_r,

   For $a = 1$ to M,

   Calculate $I(X_r, V_{ra})$ as

$$I(X_r, V_{ra}) = \sum_{k=1}^{O} \sum_{i=1}^{N_r} \log \frac{P(v_{ra}^k | x_{ri})}{P(v_{ra}^k)}.$$

   (c) Sort the features in $\mathbf{V}_r$ in descending order.

   (d) Calculate the average mutual information

$$I(X_r, \mathbf{V}_r) = \frac{1}{M} \sum_{i=1}^{M} I(X_r, V_{ri})$$

(e) Remove the worst F features from the list of features in $\mathbf{V}_r$.

(f) Compare the value of $I(X_r, \mathbf{V}_r)$ with its value in previous iteration. If more, add the next best F features from L_r to $\mathbf{V}_r$. If less, add the next best $\frac{F}{2}$ features from L_r to $\mathbf{V}_r$ and the best $\frac{F}{2}$ features removed from $\mathbf{V}_r$ in the previous iteration.

7. End.

5.1.4 Performance evaluation

The speech is sampled at 16 kHz, and preemphasized, then a Hamming window with a width of 20 ms is applied every 10 ms. The acoustic features are calculated for 3300 utterances from the TIMIT database. These acoustic features are 12 LPC based cepstrum coefficients, 12 MFCC coefficients, 12 PLP coefficients, energy, and their averages over periods of 150 msec, and their differences over periods of 5 msec. These acoustic features sum up to 111 features. The 61 phonemes defined in the TIMIT database are mapped to values of the phonological factors, and hence the phoneme labels are mapped to phonological factors labels for each frame of speech. The algorithm described before is used to select a 39-feature vector from the 111 features available to represent each phonological factor. The number of acoustic features of each category in the final representation of each phonological factor is shown in Table 5.2.

High mutual information between an acoustic feature and the phonological factor does not necessarily imply that it will have high mutual information with the phonological factor when included in a certain acoustic feature vector. This is due to the correlation between the features in the vector. However, our experiments show that more than 70% of the features that end up in the final representation of the phonological factor were in the top 39 features of high mutual information with the phonological factor based on initial individual features probability density functions. That is why the initialization of the feature vector with these features decreases the time and amount of computation required to get good results.

Table 5.2 Number of acoustic features in each MMIA representation of the phonological factors.

Acoustic Type	Voicing	Duration	Manner	Place
Energy	1	1	1	1
△ Energy	1	1	1	1
Energy Avg.	1	0	1	0
LPCC	2	3	2	5
△ LPCC	0	6	4	5
LPCC Avg.	0	9	7	2
MFCC	3	3	2	3
△ MFCC	5	0	5	6
MFCC Avg.	12	12	5	8
PLP	2	2	3	5
△ PLP	6	2	6	2
PLP Avg.	5	0	2	1

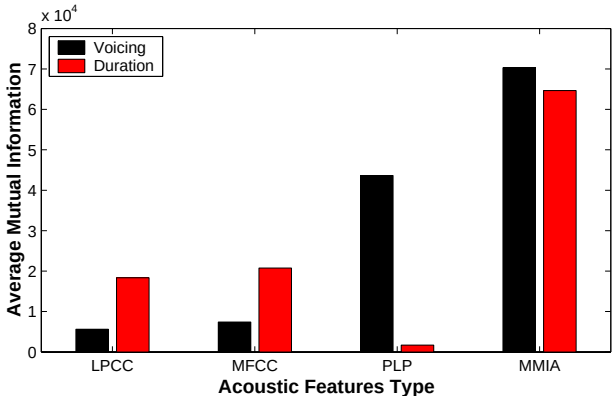

Figure 5.1 Average Mutual Information of Voicing and Duration with Their Acoustic Representation

As shown in Figure 5.1, the maximum mutual information acoustic (MMIA) representation achieved by our algorithm came out with an increase in mutual information of about 100% in the case of voicing and 220% in the case of duration. The average mutual information achieved its maximum in five iterations in the case of voicing and in eight iterations in the case of duration. Although the achieved feature vector is not necessarily the optimal over the available features, it has an average mutual information that is two to three times better than the best feature representation with the same feature vector length.

Also, shown in Figure 5.2, the maximum mutual information acoustic (MMIA) representation achieved by our algorithm came out with an increase in mutual information of about 9% in the case of place of articulation and 2.5% in the case of manner of articulation. The average mutual information achieved its maximum in eight iterations in the case of place of articulation and in ten iterations in the case of manner of articulation.

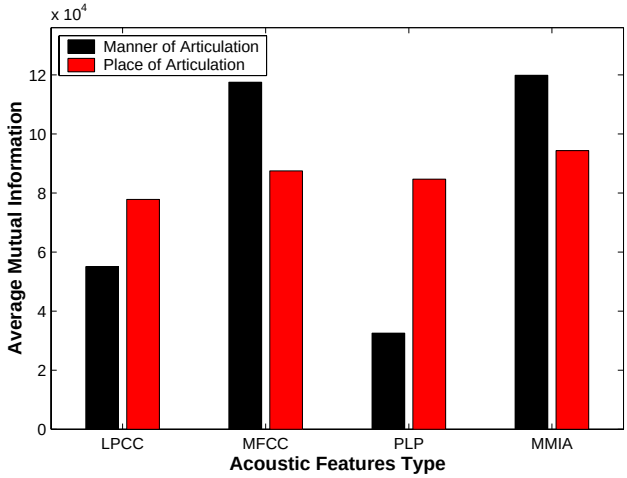

Figure 5.2 Average Mutual Information of Place and Manner of Articulation with Their Acoustic Representation

The small improvement in average mutual information in the latter case compared to improvements in Figure 5.1 is due to using acoustic features that are calculated to model the vocal tract, not the source of excitation or the phoneme duration.

5.2 MMI Feature Selection for Phoneme Recognition

This part addresses the problem of finding a subset of the acoustic feature space that best represents the phoneme set used in a speech recognition system. Our maximum mutual information approach is applied here to select the acoustic features to be combined to represent the distinctions among the phonemes.

The overall phoneme recognition accuracy is slightly increased for the same length of feature vector for clean speech and at 10 dB compared to FFT-based Mel-frequency cep-

strum coefficients (MFCC) by using acoustic features selected based on a maximum mutual information criterion.

Using 16 different feature sets, the rank of the feature sets based on mutual information can predict phoneme recognition accuracy with a correlation coefficient of 0.71 compared to a correlation coefficient of 0.28 when using a criterion based on the average pair-wise estimate of the divergence to rank the feature sets.

The feature stream used in each recognizer is selected from an acoustic feature space formed from linear prediction cepstrum coefficients (LPCC), MFCC cepstrum coefficients based on FFT, perceptual linear prediction (PLP) cepstrum coefficients, FM coefficients, energy, their deltas, and the average of the deltas. The FM coefficients are a nonlinear measure of local spectral compactness, based on the theory of band-limited phase demodulation.

To maximize the mutual information between a phoneme set and the subset of acoustic features that are used to model it, we used the algorithm described in Section 5.1 and in [17], which approximates this maximization within a small number of iterations. The algorithm is guaranteed to achieve a subset with high mutual information but not necessarily the optimal one.

Instead of using the feature set that achieves the maximum mutual information, we select the N best feature sets generated by this algorithm and select from them the one with the highest recognition accuracy.

We used this algorithm also to select the acoustic-feature representation of phonemes using a criterion based on an estimate of the average divergence between all phoneme pairs.

The divergence $L(i,j)$ between the probability density functions $P(\mathbf{y}|x_i)$ and $P(\mathbf{y}|x_j)$ is

$$L(i,j) \;=\; \int P(\mathbf{y}|x_i) \log \frac{P(\mathbf{y}|x_i)}{P(\mathbf{y}|x_j)} \mathbf{dy}. \tag{5.2}$$

For two uncorrelated n-dimensional Gaussian probability density functions, $P(\mathbf{y}|x_i)$ and $P(\mathbf{y}|x_j)$, with mean vectors $\mu_i = [\mu_{i1}\mu_{i2}\cdots\mu_{in}]$ and $\mu_j = [\mu_{j1}\mu_{j2}\cdots\mu_{jn}]$, and diagonal

covariance matrices $\mathbf{\Sigma}_i$ and $\mathbf{\Sigma}_j$, respectively, $L(i,j)$ is

$$L(i,j) = \sum_{r=1}^{n} d_r(i,j), \tag{5.3}$$

where $d_r(i,j)$ is the divergence between the marginal distributions corresponding to the rth feature, and is given by

$$d_r(i,j) = \frac{-1}{2} + \frac{\sigma_{ir}^2}{2\sigma_{jr}^2} + \frac{(\mu_{ir}-\mu_{jr})^2}{2\sigma_{jr}^2} + \log\frac{\sigma_{jr}}{\sigma_{ir}}, \tag{5.4}$$

where σ_{ir}, and σ_{jr} are the variances of the rth feature in $\mathbf{\Sigma}_i$ and $\mathbf{\Sigma}_j$, respectively. Then the average pairwise estimate of the divergence of the conditional PDFs of each phoneme c with phoneme k is [64]

$$D(c,k) = \sum_{i=1}^{m_c}\sum_{j=1}^{m_k} H_i L(i,j) + H_i \log\frac{H_i}{H_j}, \tag{5.5}$$

where H_i is the weight of the ith Gaussian PDF in the mixture of Gaussians that model phoneme c, H_j is the weight of the jth Gaussian PDF in the mixture of Gaussians that model phoneme k, m_c is the number of Gaussian PDFs in the mixture of Gaussians modeling phoneme c, m_k is the number of Gaussian PDFs in the mixture of Gaussians modeling phoneme k, and the average estimate of the divergence based criterion for the entire phoneme set, D, is

$$D = \sum_{c=1}^{J}\sum_{k=1,k\neq c}^{J} P(c)D(c,k), \tag{5.6}$$

where $P(c)$ is the *a priori* probability of phoneme c.

5.2.1 Experiments and results

The speech is sampled at 16 kHz, and preemphasized, then a Hamming window with a width of 20 ms is applied every 10 ms. The acoustic features are calculated for 4500 utterances from the TIMIT database. These acoustic features are 12 LPC based cepstrum coefficients, 12 MFCC coefficients, 12 PLP coefficients, 14 FM coefficients, energy, and the average of their deltas over periods of 150 ms., and their differences over periods of 5 ms. These acoustic features sum up to 153 features. The 61 phonemes defined in the TIMIT database are mapped to 48 phoneme labels for each frame of speech. The algorithm described before and in [17] is used to select a 39-feature vector from the 153 features available. The indexes of the acoustic features of each category in the final representation of the phonemes are shown in Table 5.3. This acoustic feature representation has an empirical mutual information with the phoneme set that is about 30% over the best single-type acoustic features. The average empirical mutual information achieved its maximum in 10 iterations.

Table 5.3 Indexes of acoustic features in the final MMIA representation of the phoneme set.

Acoustic Type	**Indexes of Coefficients**	**Total Number**
Energy	1	1
△ Energy	None	1
Energy Avg.	None	1
LPCC	1,11	12
△ LPCC	1,2,3,4,5,6	12
LPCC Avg.	1,2,3,4, 5	12
MFCC	5,6,8,9,10,11,12	12
△ MFCC	9,10,11,12	12
MFCC Avg.	4,5,6,7,8,9,10,11,12	12
PLP	None	12
△ PLP	5	12
PLP Avg.	None	12
FM	1	14
△ FM	7,8	14
FM Avg.	1	14

Using intermediate feature sets generated while trying to maximize the mutual information, we trained many three-state left-to-right HMM phoneme models. These models are built based on maximum likelihood estimation (MLE) training using the EM algorithm. HMM phoneme models based on MFCC, FM, LPCC, energy, and their deltas and average deltas were trained for comparison. Tables 5.4, and 5.5 show the phoneme recognition accuracy of the maximum mutual information acoustic (MMIA) features compared to MFCC, LPCC, and FM for clean speech and at 10 dB with and without the language model, respectively. MMIA has consistent slightly superior performance with and without using bigram language model. Experiments based on all sets of features except FM were done to test how well an empirical estimate of the mutual information can predict phoneme recognition accuracy based on certain features.

As shown in Figure 5.3, the MMIA representation algorithm can predict the acoustic representation that will give a better recognition accuracy. The correlation coefficient between the rank and the phoneme recognition accuracy is 0.71. Also low values of phoneme recognition accuracy based on an acoustic feature set result in low values of the mutual information between this feature set and phonemes.

Table 5.4 Phoneme recognition accuracy (%) on TIMIT for clean speech and at 10 dB with bigram model.

Acoustic Features	**Clean Speech**	**At 10 dB**
MMIA	62.91	49.24
MFCC	62.82	47.53
FM	61.64	46.19
LPCC	58.74	45.95

The rank of the acoustic feature set that achieves best results is second based on mutual information. However, testing the two feature sets ranked first and second based on mutual information in a noisy environment (at 10 dB with additive white Gaussian noise), their phoneme recognition accuracies are 49.24% and 48.78%, respectively, compared to 47.53%

Table 5.5 Phoneme recognition accuracy (%) on TIMIT for clean speech and at 10 dB without language model.

Acoustic Features	Clean Speech	At 10 dB
MMIA	58.59	42.65
MFCC	58.15	41.93
FM	57.98	42.37
LPCC	55.85	40.18

for MFCC. These results suggest that the correlation coefficient of the rank and phoneme recognition accuracy improves in noisy environments.

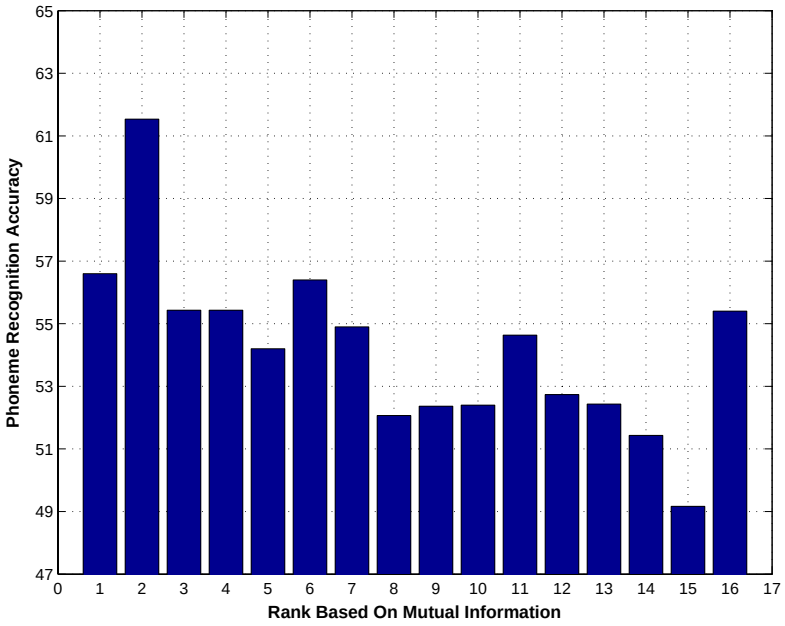

Figure 5.3 Phoneme Recognition Accuracy of Feature Sets Selected Based on Mutual Information

The same algorithm is used to select an acoustic feature representation maximizing the criterion based on average Kullback-Liebler divergence D between all phoneme pairs in the phoneme set.

As shown in Figure 5.4, the divergence-based criterion for feature selection is not as good as mutual information in predicting the acoustic representation that will give a better recognition accuracy. The correlation coefficient between the rank, in this case, and the phoneme

recognition accuracy is only 0.28. The rank of the acoustic-feature representation that gives the best phoneme recognition accuracy is 6 based on maximum average Kullback-Liebler divergence criteria. Low values of phoneme recognition accuracy based on an acoustic-feature set do not necessarily result in low values of the average Kullback-Liebler divergence among phoneme models based on this feature set.

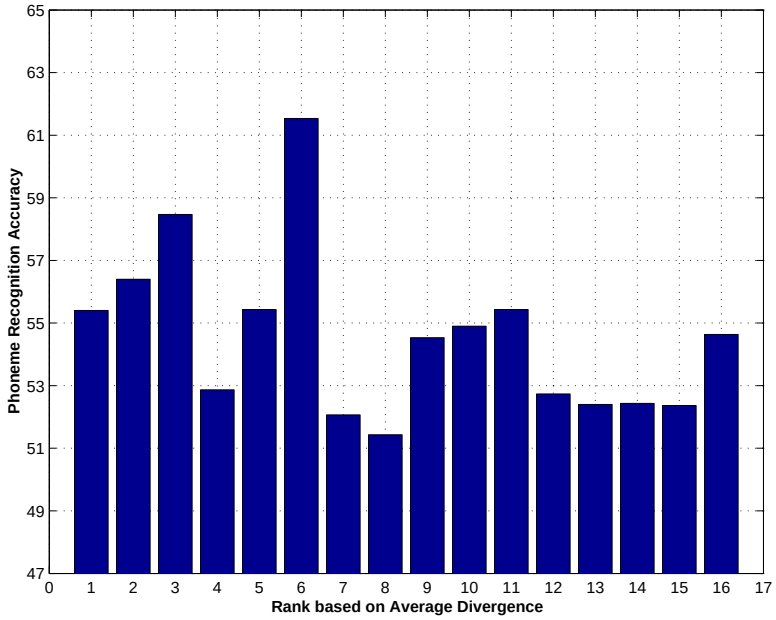

Figure 5.4 Phoneme Recognition Accuracy for Feature Sets Selected Based on Average Divergence

5.2.2 Discussion of the results

The maximum mutual information feature selection approach introduced in this section increases the average mutual information between phonemes and their acoustic-features representation by 30% more than the best single category (LPCC, MFCC, FM, or PLP) representation of the same length. These results were achieved in 10 iterations of the algorithm, so the time and computational requirement are negligible compared to trying to achieve the optimal maximum mutual information feature vector even over a moderate set of acoustic

features. This provides a promising approach for speech recognition systems based on a combination of different acoustic features. This approach can be easily combined with adaptive techniques to get better recognition accuracy in noisy environments. Due to approximations in the assumed probabilistic model of phonemes, an increase in the empirical estimate of the mutual information between phonemes and their acoustic feature representation estimated using these models does not guarantee an increase in the recognition accuracy. However, the results we achieved prove that it is a good approximation of the phoneme recognition accuracy that could be achieved using certain acoustic features. This allows testing different acoustic-feature combinations using this approach before even designing the recognizer, and then selecting few possible combinations based on our approach. A small number of recognition experiments is enough to select the best acoustic representation that should be used in modeling phonemes in the recognizer. An important advantage of this approach is that it can be easily generalized to use the same probabilistic model used in any recognition system. It can be applied also to any speech unit, not only to phonemes.

5.3 Discriminative Generalizations of LDA

One of the main objectives of speech signal analysis in ASR systems is to produce a parameterization of the speech signal that reduces the amount of data that is presented to the speech recognizer, and captures salient characteristics suited for discriminating among different speech units. Most ASR systems use cepstral features augmented with dynamic information from the adjacent speech frames. The algorithms for cepstral features estimation use concepts based on human speech perception like Mel-frequency scaling and critical band filters to simulate the front-end of the human auditory system. Even with additional techniques for speaker normalization and combating environmental noise, incorporating properties of human speech production and auditory perception is not necessarily the optimal approach to feature extraction for speech recognition, as they are not optimized to discriminate among

speech units. This motivated the application of data-driven dimensionality techniques to feature extraction for speech recognition. Most dimensionality reduction techniques applied to speech recognition are variants or extensions of linear discriminant analysis (LDA) [16].

5.3.1 Limitations of LDA and HDA

The results reported on the application of LDA to speech recognition show consistent gain for small vocabulary tasks and mixed results for large vocabulary applications [68]. This can be attributed mainly to making assumptions about the problem that are unrealistic like equal class-conditional covariance matrices, and using an optimality criterion that is not necessarily consistent with the objective of minimizing the recognition error. It was shown that linear discriminant analysis is related to the maximum likelihood estimation of parameters for a Gaussian model, with *a priori* assumptions on the structure of the model [79]. This result is further generalized by assuming that class distributions are a mixture of Gaussians [80]. In [68], LDA is generalized to the case of classes of different covariance matrices and this generalization is referred to as heteroscedastic discriminant analysis (HDA). An alternative interpretation of HDA as a constrained maximum likelihood projection for a Gaussian model is introduced in [56].

The objective function in all these methods is not directly related to minimizing the recognition error, and therefore does not necessarily minimize the discrimination loss due to dimensionality reduction. LDA transformation, for example, tends to preserve distances of already well-separated classes [129]. Maximizing the mutual information between the features and the class is more intuitively related to minimizing the recognition error, and therefore we argue that it is a better objective for discriminant analysis than maximizing the likelihood under some model assumptions or constraints.

In this section, we show that calculating the LDA transformation matrix is a maximum conditional mutual information estimation (MCMIE) problem with constraints on both the

class-conditional and the unconditional PDFs. By relaxing these constraints, we present in Section 5.4 a generalization of LDA to MCMI projection (MCMIP), and describe an algorithm that calculates the MCMIP transform given the recognizer model. This generalization has three advantages: it maximizes the *a posteriori* probability of the model corresponding to the training data given the data which is closely related to minimizing the training data recognition error, it is calculated in the lower-dimensional space, and it takes into consideration the assumptions of the recognizer model.

5.3.2 Maximum mutual information interpretation of LDA

There are several possible class separability measures. One of the most general measures of the ability of the features to discriminate among classes is its mutual information with the classes. Mutual information is an invariant measure under any one-to-one transformation. Therefore, for a full-rank linear transform of the $n \times 1$ feature vector $\mathbf{x}$,

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \theta \\ \psi \end{bmatrix} \mathbf{x}, \tag{5.7}$$

where $\mathbf{y}$ is a $p \times 1$ vector, $\mathbf{z}$ is a $(n-p) \times 1$ vector, θ is a $p \times n$ matrix, and ψ is a $(n-p) \times n$ matrix,

$$I(\mathbf{Y}, C) \leq I(\mathbf{X}, C), \tag{5.8}$$

with equality if and only if $I(\mathbf{Z}, C) = 0$. This happens if and only if the feature vector $\mathbf{Z}$ is statistically independent of the class identity C [90]. Therefore, we should expect that getting rid of these features will have negligible effect on the recognizer performance or even improve it, if it has a negligible mutual information with the class identities. The mutual

information between the feature vector $\mathbf{Y}$ and the set of classes C is

$$I(\mathbf{Y}, C) = E_{P(\mathbf{Y},C)}\left[\log \frac{P(\mathbf{y}|c)}{P(\mathbf{y})}\right], \tag{5.9}$$

where $\{P(\mathbf{y}|c_j)\}_{j=1}^{J}$ and $P(\mathbf{y})$ are the class-conditional and the unconditional PDFs, respectively. Since we do not have the true PDFs, we calculate an estimate of the mutual information, which is the conditional mutual information given a maximum likelihood estimate of the parameters $\mathbf{\Lambda}$ of both $\{P(\mathbf{y}|c_j)\}_{j=1}^{J}$ and $P(\mathbf{y})$

$$\hat{I}(\mathbf{Y}, C|\mathbf{\Lambda}) = \sum_{i=1}^{N} \log \frac{P(\mathbf{y}^i|c^i, \mathbf{\Lambda})}{P(\mathbf{y}^i|\mathbf{\Lambda})}, \tag{5.10}$$

where N is the number of training frames, and c^i is the class corresponding to the ith frame.

Our goal here is to show that LDA is equivalent to the problem of finding the linear transformation matrix θ that maximizes the conditional mutual information between the lower-dimensional feature vector $\mathbf{Y}$ and the class identity C with *a priori* assumptions on the structure of the model. Let each class-conditional PDF in the lower-dimensional space be modeled by a Gaussian PDF with all of them sharing the same covariance matrix

$$P(\mathbf{y}|c_j) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{W}_\mathbf{y}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}-\mu_j)^T \mathbf{W}_\mathbf{y}^{-1}(\mathbf{y}-\mu_j)\right), \quad \text{for } j = 1, \cdots J, \tag{5.11}$$

where $\mathbf{W}_\mathbf{y}$ is the maximum-likelihood estimate (MLE) of the class-conditional covariance matrix, and μ_j is the MLE of the mean. Let also the unconditional PDF in the lower-dimensional space be modeled by a Gaussian PDF

$$P(\mathbf{y}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{\Sigma}_\mathbf{y}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}-\mu)^T \mathbf{\Sigma}_\mathbf{y}^{-1}(\mathbf{y}-\mu)\right), \tag{5.12}$$

where $\mathbf{\Sigma_y}$ is the maximum-likelihood estimate of the covariance matrix, μ is the MLE of the global mean.

Then maximizing the conditional mutual information given the maximum-likelihood estimate of these models with respect to θ is equivalent to maximizing

$$V = \log|\mathbf{\Sigma_y}| - \log|\mathbf{W_y}|. \tag{5.13}$$

Using the relations

$$\mathbf{\Sigma_y} = \theta^T(\mathbf{W}+\mathbf{B})\theta, \tag{5.14}$$

$$\mathbf{W_y} = \theta^T\mathbf{W}\theta, \tag{5.15}$$

and that the logarithm is a monotonic function, the objective function to be maximized becomes

$$O = \frac{\left|\theta^T(\mathbf{W}+\mathbf{B})\theta\right|}{|\theta^T\mathbf{W}\theta|}. \tag{5.16}$$

The $p \times n$ transformation θ^* that maximizes the objective function in Equation (5.16) is the matrix consisting of the p eigenvectors of the Fisher covariance matrix $\mathbf{W}^{-1}\mathbf{B}$ corresponding to the largest p eigenvalues, and therefore is the solution of the LDA maximization also as described in Chapter 2.

It should be noted that the assumption that $P(\mathbf{y})$ is Gaussian is inconsistent with the assumption that $\{P(\mathbf{y}|c_j)\}_{j=1}^J$ are Gaussian, as in general if $\{P(\mathbf{y}|c_j)\}_{j=1}^J$ are Gaussian PDFs, then $P(\mathbf{y})$ is a Gaussian mixture PDF. This explicit modeling of $P(\mathbf{y})$ that is inconsistent with the models for $\{P(\mathbf{y}|c_j)\}_{j=1}^J$ is a serious limitation of LDA. It is the main reason that the LDA solution in many cases does not correspond to minimizing the recognition error.

Heteroscedastic discriminant analysis (HDA), as described in Chapter 2, is an extension

to LDA that removes the equal covariance constraint [68]. HDA was first formulated as a maximum likelihood estimation problem for normal populations with common covariance matrix in the rejected subspace. An alternative interpretation of HDA as a constrained maximum likelihood projection for a full-covariance Gaussian model is introduced in [56]. HDA can be related to the maximization of the conditional mutual information in the lower dimensional space by removing the equal class-conditional covariance from the previous derivation for LDA. The assumption that $P(\mathbf{y})$ is Gaussian is still inconsistent with the assumption that $\{P(\mathbf{y}|c_j)\}_{j=1}^{J}$ are Gaussian. Using the convexity of the relative entropy [90], it can be shown that this assumption underestimates the conditional mutual information as opposed to calculating $P(\mathbf{y})$ from the class-conditional PDFs $\{P(\mathbf{y}|c_j)\}_{j=1}^{J}$, i.e.,

$$\hat{I}_{DA}(\mathbf{Y}, C|\mathbf{\Lambda}) \;\leq\; \hat{I}(\mathbf{Y}, C|\mathbf{\Lambda}), \tag{5.17}$$

where $\hat{I}_{DA}(\mathbf{Y}, C|\mathbf{\Lambda})$ is the conditional mutual information estimated with an explicit Gaussian model of $P(\mathbf{y})$, and $\hat{I}(\mathbf{Y}, C|\mathbf{\Lambda})$ is the conditional mutual information estimated by calculating $P(\mathbf{y})$ from the class-conditional PDFs $\{P(\mathbf{y}|c_j)\}_{j=1}^{J}$.

5.4 Maximum Conditional Mutual Information Projection

In the following, we will relax the constraints of discriminant analysis to develop the maximum conditional mutual information projection (MCMIP) approach. This generalization has three advantages: it maximizes the *a posteriori* probability of the model corresponding to the training data given the data which is closely related to minimizing the training data recognition error, it is calculated in the lower-dimensional space, and it takes into consideration the assumptions of the recognizer model.

5.4.1 MCMIP formulation

Given a set of class-conditional probabilistic models used by the classifier or the recognizer, the goal of MCMIP is to find a p-dimensional subspace of an n-dimensional feature space that retains the discrimination information contained in the original high-dimensional space by maximizing an estimate of the conditional mutual information between the features and the class identity. In other words, MCMIP searches for the $p \times n$ linear transformation or projection θ^* of the features that maximize the conditional mutual information $\hat{I}(\mathbf{Y}, C|\mathbf{\Lambda})$, i.e.,

$$\theta^* = \arg\max_{\theta} \hat{I}(\mathbf{Y}, C|\mathbf{\Lambda}), \tag{5.18}$$

where $\mathbf{y} = \theta\mathbf{x}$. From the previous discussion of discriminant analysis, the feature vector $\mathbf{Y}$ achieved by MCMIP has a higher conditional mutual information with the class identities given the classifier's set of class-conditional probabilistic models than the one obtained by discriminant analysis approaches. From Equation (5.10), it can be easily shown that maximizing $\hat{I}(\mathbf{Y}, C|\mathbf{\Lambda})$ is equivalent to maximizing the *a posteriori* probability of the model corresponding to the training data given the data which is closely related to minimizing the training data recognition error.

5.4.2 Implementation of MCMIP for speech recognition

Applying the MCMIP approach for dimensionality reduction to an HMM-based speech recognizer requires the estimation of the conditional mutual information given the HMM parameters. The parameters of the HMM recognizer can be calculated using maximum likelihood estimation or discriminant approaches like maximum mutual information. We choose to use the expectation maximization (EM) algorithm to get maximum likelihood estimates of the HMM parameters [93]. Using these estimates of the parameters, the empirical estimate of

the mutual information to be maximized is

$$\hat{I}(\mathbf{Y}, C|\mathbf{\Lambda}) = \sum_{i=1}^{N} \left(\log P_{\mathbf{\Lambda}}(\mathbf{y}^i|c^i) - \log \left(\sum_{j=1}^{J} P_{\mathbf{\Lambda}}(\mathbf{y}^i|c_j) P_{\mathbf{\Lambda}}(c_j) \right) \right), \tag{5.19}$$

where c^i is the maximum likelihood state assignment for the ith frame from the training data, N is the number of frames in the training data, and

$$\begin{aligned} P_{\mathbf{\Lambda}}(\mathbf{y}^i|c_j) &= \sum_{k=1}^{K} H_{jk} \frac{1}{(2\pi)^{\frac{n}{2}} |\mathbf{\Sigma}_{jk}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (\mathbf{y} - \mu_{jk})^T \mathbf{\Sigma}_{jk}^{-1} (\mathbf{y} - \mu_{jk}) \right), \\ \sum_{k=1}^{K} H_{jk} &= 1 \end{aligned} \tag{5.20}$$

for all $j = 1, 2, \cdots, J$, where H_{jk} is the weight of the kth Gaussian PDF in the Gaussian mixture of state j, K is the number of Gaussian PDFs in the Gaussian mixture, μ_{jk} is the mean of the kth Gaussian PDF in the mixture, and $\mathbf{\Sigma}_{jk}$ is the covariance matrix of the kth Gaussian PDF in the mixture.

To use a gradient-based algorithm to maximize our empirical estimate of the conditional mutual information $\hat{I}(\mathbf{Y}, C|\mathbf{\Lambda})$ with respect to the linear transform θ, we calculate the derivative of the objective function with respect to θ

$$\begin{aligned} \frac{d\hat{I}(\mathbf{Y}, C|\mathbf{\Lambda})}{\mathbf{d}\theta} &= \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{P(\mathbf{y}^i|c^i k)}{P_{\mathbf{\Lambda}}(\mathbf{y}^i|c^i)} \mathbf{\Sigma}_{c^i k}^{-1} (\mu_{c^i k} - \mathbf{y}^i) \mathbf{x}^{i^T} \\ &\quad - \sum_{i=1}^{N} \sum_{j=1}^{J} \sum_{k=1}^{K} \frac{P_{\mathbf{\Lambda}}(\mathbf{y}^i|c_{jk}) P_{\mathbf{\Lambda}}(c_{jk})}{P_{\mathbf{\Lambda}}(\mathbf{y}^i)} \mathbf{\Sigma}_{jk}^{-1} (\mu_{jk} - \mathbf{y}^i) \mathbf{x}^{i^T}. \end{aligned} \tag{5.21}$$

The steps of the iterative algorithm to update the transformation matrix θ and the HMM parameters are

1. Initialize the transformation matrix θ.

2. Calculate the feature vectors $\mathbf{y}$ using the relation $\mathbf{y} = \theta\mathbf{x}$, where $\mathbf{x}$ is the input acoustic feature vector.

3. Using the EM algorithm, estimate the HMM parameters and segment the training data.

4. Using the current HMM parameters and training data segmentation, estimate θ that maximizes the conditional mutual information $\hat{I}(\mathbf{Y}, C|\mathbf{\Lambda})$ using the conjugate-gradient algorithm.

5. Iterate (starting from 2) until convergence.

5.4.3 Experiments and results

The MCMIP algorithm is used to study the optimal feature subspace for diagonal-covariance Gaussian mixture HMM modeling of the TIMIT database.

The baseline 26-feature vector consists of 12 MFCC coefficients, energy, and their deltas. The input to the MCMIP algorithm consists of five of these feature vectors centered at the target frame. This 130-feature vector is then transformed using the MCMIP algorithm to a 26-feature vector. In each iteration, the new feature vector is calculated using the current MCMIP transformation parameters, then the maximum likelihood estimates of the HMM model parameters are calculated. Then, the MCMIP transformation matrix is calculated using the conjugate-gradient algorithm. After the iterative algorithm converges to a set of locally optimal HMM and MCMIP parameters, the training data are transformed by the MCMIP matrix yielding the final MCMIP feature vector.

In our experiments, the 61 phonemes defined in the TIMIT database are mapped to 48 phoneme labels for each frame of speech as described in [112]. These 48 phonemes are collapsed to 39 phonemes for testing purposes as in [112]. A three-state left-to-right model for each triphone is trained. The number of mixtures per state was varied between 3 and

13, depending on the number of training observations assigned to the state. The parameters of the recognizer and the MCMIP transform are trained using the training portion of the TIMIT database. The parameters of the triphone models are then tied together using the same approach as in [115].

To compare the performance of the proposed algorithm with other approaches, we generated acoustic features using LDA and HDA. We used the same 130-feature vector input to MCMIP with both LDA and HDA and kept the dimensions of the output of LDA and HDA the same as the MCMIP output.

Testing this recognizer, using the test data in the TIMIT database, we get the phoneme recognition results in Table 5.6. These results are obtained by using a bigram phoneme language model and by keeping the insertion error around 10% as in [112]. The table compares MCMIP recognition results to the ones obtained by the baseline MFCC, LDA, and HDA.

Table 5.6 Phoneme recognition accuracy (%) on TIMIT for MFCC features and features generated by LDA, HDA or MCMIP.

Acoustic Features	**Recognition Accuracy**
MFCC	73.7%
LDA	73.8%
HDA	74.1%
MCMIP	74.7%

5.4.4 Discussion

In this work, we described a framework for discriminant analysis for speech recognition. This framework is an extension of current approaches by relaxing the constraints imposed on the model in LDA and HDA approaches. Our approach maximizes the conditional mutual information between the feature vector and the HMM states, which is closely related to recognition error, as opposed to maximizing the likelihood in LDA and HDA approaches,

which is not directly related to recognition error. We introduced also an iterative algorithm to calculate the MCMIP matrix for an HMM-based recognizer. Phoneme recognition experiments using features generated by this algorithm show significant improvement compared to previous dimensionality reduction transforms like LDA and HDA.

CHAPTER 6

SUMMARY AND DIRECTIONS

In this chapter, we will summarize the most important points, in the author's view, that are presented in the dissertation, and give directions for future related work. In Section 6.1, we will discuss these issues for the model enforcement approach, and we will discuss them for the discriminative dimensionality reduction approach in Section 6.2.

6.1 The Model Enforcement Approach

The model enforcement problem formulation provides a unified feature transformation framework that can be applied to any statistical classification or recognition problem. It provides a unified framework for many previous linear techniques in several research areas. Examples of these areas include statistics, pattern recognition, signal processing, and data mining. It extends these techniques to not-necessarily-linear approaches. We described in Chapter 3 some applications based on a class of nonlinear volume-preserving transforms—namely symplectic maps. However, the choice of the class of maps that we restrict our solutions to is in general problem-dependent. Hence, the problem of determining this class for a given application is an interesting area of future research. It should be noted that the main advantages of most existing linear techniques for feature transformation over nonlinear techniques are simplicity and computational efficiency. Therefore, simplicity and computational efficiency should be among the main factors that affect the choice of the class of maps. Other possible factors can be obtained from knowledge about the features and/or the model used in the classification

or recognition system. For the ASR problem, our choice of symplectic maps was motivated by the computational efficiency during both training and decoding, and the knowledge that the problem to be solved is caused by sources that are represented nonlinearly in the original feature space.

The approach presented here for strong-sense class-dependent features provides solutions to many problems related to using class-dependent features in statistical classification and recognition systems. It gives class-dependent features that have comparable likelihoods and avoids the need of noise-only models that are usually used with weak-sense class-dependent features. The choice of the class of maps that are used in generating strong-sense class-dependent features remains as an important topic for future research for ASR and other applications. Our choice of using symplectic maps was motivated by the computational efficiency and the simplicity of the decoding process guaranteed by the volume-preserving property of the symplectic maps. As the computational capabilities of computers improve, the importance of computational efficiency decreases and many other classes of maps can be tested.

6.2 Discriminative Feature Selection and Dimensionality Reduction Approaches

As explained in Chapter 5, maximizing the conditional mutual information between the classes and the features satisfies many requirements of an optimal objective criterion for dimensionality reduction and a possible alternative to the optimal objective criterion for feature selection. By giving an interpretation of LDA as a special case of the maximum conditional mutual information projection approach, we can consider many possible generalizations of LDA by relaxing constraints on the probabilistic model or the projecting function. Contrary to previous generalizations of LDA, these generalizations maximize a discriminative objective criterion in the lower-dimensional feature space. We introduced in Chapter 5 a linear generalization. Many other possible generalizations can be considered in future re-

search. A related problem that, in the author's view, should be considered in future research is the consistency of the solutions based on an empirical estimate of the conditional mutual information.

REFERENCES

[1] J. R. Deller, J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals.* Upper Saddle River, NJ: IEEE Press, 2000.

[2] B. Gold and N. Morgan, *Speech And Audio Signal Processing.* New York, NY: Wiley, 1999.

[3] G. Saon, M. Padmanabhan, and R. Gopinath, "Eliminating inter-speaker variability prior to discriminant transforms," in *ASRU Workshop Proceedings*, Trento, Italy, December 2001, pp. 73–76.

[4] L. Lee and R. Rose, "A frequency warping approach to speaker normalization," *IEEE Trans. On Speech And Audio Processing*, vol. 6, no. 1, pp. 353–356, January 1998.

[5] M. J. F. Gales, "Maximum likelihood linear transformation for HMM-based speech recognition," Cambridge University, Engineering Department, Technical Report CUED/F-INFENG, 1997.

[6] R. Lippmann "Speech recognition by machines and humans," *Speech Communication*, vol. 22, no. 1, pp. 1–15, July 1997.

[7] P. M. Baggenstos, "Class-specific features in classification," *IEEE Trans. On Signal Processing*, vol. 47, pp. 3428–3432, December 1999.

[8] S. Kay, "Sufficiency, classification, and the class-specific feature theorem," *IEEE Trans. On Information Theory*, vol. 46, no. 4, pp. 1654–1658, July 2000.

[9] K. Kirchoff, G. A. Fink, and G. Sagerer, "Conversational speech recognition using acoustic and articulatory input," in *IEEE Proceedings of ICASSP*, Istanbul, 2000, pp. 891–894.

[10] J. Glass, J. Chang, and M. McCandless, "A probabilistic framework for feature-based speech recognition," in *Proc. of Int. Conf. of Spoken Language Processing*, Philadelphia, PA, 1996, pp. 2277–2280.

[11] M. J. F. Gales, "Maximum likelihood multiple subspace projections for hidden Markov models" *IEEE Trans. On Speech And Audio Processing*, vol. 10, no. 2, pp. 37–47, February 2002.

[12] A. Bailey, "Class-dependent features and multicategory classification," Ph.D. dissertation, ECE Department, University of South Hampton, 2001.

[13] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *IEEE Proceedings of ICASSP*, Tokyo, Japan, 1986, pp. 49–52.

[14] Y. Normandin, "Hidden Markov models, maximum mutual information estimation, and the speech recognition problem," Ph.D. dissertation, Dept. of Elect. Eng., McGill University, Montreal, 1991.

[15] S. Katagiri, B.-H. Juang, and C.-H. Lee, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic decent method," *IEEE Processings*, vol. 86, no. 11, pp. 2345–3375, November 1998.

[16] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification.* New York, NY: Wiley, 2000.

[17] M. K. Omar and M. Hasegawa-Johnson, "Maximum mutual information based acoustic-features representation of phonological features for speech recognition," in *IEEE Proceedings of ICASSP*, Orlando, Florida, 2002, pp. 81–84.

[18] M. K. Omar, K. Chen, M. Hasegawa-Johnson, and Y. Brandman, "An evaluation of using mutual information for selection of acoustic-features representation of phonemes for speech recognition," in *Proc. of Int. Conf. of Spoken Language Processing*, Denver, CO, September 2002, pp. 2129–2132 .

[19] M. K. Omar and M. Hasegawa-Johnson, "Maximum conditional mutual information projection for speech recognition," in *Proc. Eurospeech*, Geneva, Switzerland, 2003, pp. 505–508.

[20] K. Torkkola, "On feature extraction by mutual information maximization," in *IEEE Proceedings of ICASSP*, Orlando, Florida, 2002, pp. 821–824.

[21] C. M. Bishop, *Neural Networks for Pattern Recognition.* New York: Oxford University press Inc., 2000.

[22] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[23] H. A. Bourlard, and N. Morgan *Connectionist Speech Recognition A Hybrid Approach.* Norwell, MA: Kluwer Academic Publishers, 1994.

[24] Frederick Jelinek, *Statistical Methods For Speech Recognition.* Cambridge, MA: MIT Press, 2001.

[25] S. B. Davis and P. Mermelstein "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, vol. 28, pp. 357–366, Aug. 1980.

[26] H. Hermansky "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. America Proc.*, vol. 87, pp. 1738–1752, 1990.

[27] H. Leung, B. Chigier, and J. Glass "A comparative study of signal representation and classification techniques for speech recognition," in *IEEE Proceedings of ICASSP*, Minneapolis, MN, 1993, pp. 680–683.

[28] M. J. Hunt "Spectral signal processing for ASR," in *ASRU Workshop Proc.*, Keystone, CO, 1999, pp. 357–366.

[29] C. R. Jackowoski, H. D. H. Vo, and R. Lippmann "A comparison of signal processing front ends for automatic word recognition," *IEEE Trans. On Speech And Audio Processing*, vol. 3, no. 4,pp. 286–293, 1995.

[30] D. R. Cox, *The Analysis of Binary Data.* London, UK: Methuen, 1970.

[31] J. W. Tukey, *Exploratory Data Analysis.* Reading, MA: Addison-Wesley, 1977.

[32] J. W. Tukey, "On the comparative anatomy of transformations," *Ann. Math. Statist.*, vol. 28, pp. 602–632, 1957.

[33] G. E. P. Box, and D. R. Cox, "An analysis of transformations," *Journal of Royal Statist. Soc.*, vol. 26, pp. 211–252, 1964.

[34] D. F. Andrews, R. Gnanadesikan, and J. L. Warner, "Transformations of multivariate data," *Biometrics*, vol. 27, pp. 825–840, 1971.

[35] A. Ljolje, "The importance of cepstral parameter correlations in speech recognition," *Computer, Speech, and Language*, vol. 8, pp. 223–232, 1994.

[36] R. Koshiba, M. Tachimori, and H. Kanazawa, "A flexible method of creating HMM using block-diagonalization of covariance matrices," in *Int. Conf. on Spoken Language Processing*, 1998, pp. 371–374.

[37] M. J. F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Trans. On Speech And Audio Processing*, vol. 7, no. 3, pp. 272–281, May 1999.

[38] J. A. Bilmes, "Factored sparse inverse covariance matrices," in *IEEE Proceedings of ICASSP*, Istanbul, Turkey, 2000, pp. 1009–1012.

[39] L. K. Saul, and M. G. Rahim "Maximum likelihood and minimum classification error factor analysis for automatic speech recognition," *IEEE Trans. On Speech And Audio Processing*,vol. 8, no. 2, pp. 115–125, March 2000.

[40] B.-H. Juang, and S. Katagari, "Discriminative learning for minimum error classification," *IEEE Trans. on Signal Processing*, vol. 40, no. 12, pp. 3043–3053, 1992.

[41] M. E. Tipping and C. Bishop, "Mixtures of principal component analysis," in *Proc. of IEEE 5th Int. Conf. Artificial Neural Networks*, 1997, pp. 13–18.

[42] S. Roweis, "EM algorithms for PCA and SPCA," in *Advances in Neural Information Processing Systems*, Vol. 10, Cambridge, MA: MIT press, 1998, pp. 626–632.

[43] S. Kajerekar, N. Malayath, and H. Hermansky, "Analysis of sources of variability in speech," *Proc. of EUROSPEECH*, pp. 343–346, 1999.

[44] R. Hariharan, I. Kiss, and O. Viikki, "Noise robust speech parameterization using multiresolution feature extraction," *IEEE Trans. On Speech And Audio Processing*, vol. 9, no. 8, pp. 856–865, November 2001.

[45] H. Hermansky and N. Malayath, "Spectral basis functions from discriminant analysis," *Proc. of Int. Conf. of Spoken Language Processing*, 1998, pp. 1379–1382.

[46] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis.* New York, NY: Wiley, 2001.

[47] T.-W. Lee, *Independent Component Analysis.* Norwell, MA: Kluwer Academic Publishers, 1998.

[48] T.-W. Lee. M. Girolami, A. J. Bell, and T. J. Sejnowski, "A unifying information-theoretic framework for independent component analysis," *Computers & Mathematics with Applications*, vol. 31, no. 11, pp. 1–21, March 2000.

[49] S. Choi, H. Hong, H. Glotin, F. Berthommier, "Multichannel signal separation for cocktail party speech recognition: A dynamic recurrent network," *Neurocomputing*, vol. 49, nos. 1-4, pp. 299–314, December 2002.

[50] T.-W. Lee, A. J. Bell, and R. Orglmeister, "Blind source separation of real world signals," in *IEEE International Conference on Neural Networks*, Houston, TX, 1997, pp. 2129–2135.

[51] G.-J. Jang, T.-W. Lee, Y.-H. Oh, "Learning statistically efficient features for speaker recognition," *Neurocomputing*, vol. 49, nos. 1-4, pp. 329–348, December 2002.

[52] J.-H. Lee, H.-Y. Jung, and T.-W. Lee, "Speech feature extraction using independent component analysis," in *IEEE Proceedings of ICASSP*, vol. 3, Istanbul, Turkey, 2000, pp. 1631–1634.

[53] I. Potamitis, N. Fakotakis, and G. Kokkinakis, "Independent component analysis applied to feature extraction for robust automatic speech recognition," *Electronic Letters*, IEE, vol. 36, no. 23, pp. 1977–1978, Nov. 2000.

[54] I. Potamitis, N. Fakotakis, and G. Kokkinakis, "Spectral and cepstral projection bases constructed by independent component analysis," in *Proc. of Int. Conf. of Spoken Language Processing*, Beijing, China, 2000, pp. 63–66.

[55] R. A. Gopinath, "Maximum likelihood modeling with Gaussian distributions for classification," in *IEEE Proceedings of ICASSP*, Seattle, Washington, 1998, pp. 661–664.

[56] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces," in *IEEE Proceedings of ICASSP*, Istanbul, Turkey, 2000, pp. 1129–1132.

[57] J. Huang, B. Kingsbury, L. Mangu, M. Padmanabhan, G. Soan, and G. Zweig, "Recent improvements in speech recognition performance on large vocabulary conversational speech (Voicemail and Switchboard)," in *Proc. of Int. Conf. of Spoken Language Processing*, Beijing, China, 2000, vol. 4, pp. 338–341.

[58] A. K. Halberstadt, "Heterogeneous acoustic measurements and multiple classifiers for speech recognition," Ph.D. dissertation, MIT, ECE Department, 1998.

[59] W. Goldenthal, "Statistical trajectory models for phonetic recognition," MIT Laboratory for Computer Science, Technical Report, MIT/LCS/TR-642, 1994.

[60] H. Leung, I. Hetherington, and V. Zue, "Speech recognition using stochastic segment neural networks," in *IEEE Proceedings of ICASSP*, San Francisco, CA, 1992, pp. 613–616.

[61] M. Ostendorf and S. Roucous, "A stochastic segment model for phoneme-based continuous speech recognition," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, vol. 37, no. 12, pp. 1857–1869, December 1989.

[62] V. Zue, J. Glass, D. Goodine, H. Leung, M. Philips, J. Polifroni, and S. Seneff, "Recent progress on the SUMMIT system," in *Proceedings of Speech and Natural Language Workshop*, Hidden Valley, PA, 1990, pp. 380–384.

[63] R. Chengalvarayan and L. Deng, "Use of generalized dynamic feature parameters for speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 5, no. 3, pp. 232–242, May 1997.

[64] M. K. Omar, "Phonetic segmentation of Arabic speech for verification using HMM," M.SC. thesis, Cairo University, Egypt, January 1999.

[65] P. C. Woodland and D. Povey, "Large scale discriminative training for speech recognition," in *ASRU Workshop Proceedings*, Delavan, Wisconsin, December 2000, pp. 7–16.

[66] L. Rabiner, J. G. Wilpon, and F. K. Soong, "High performance connected digit recognition using hidden Markov models," *IEEE Trans. ASSP*, vol. 37, no. 8, pp. 1214–1225, 1989.

[67] P. M. Baggenstoss, "A modified Baum-Welch algorithm for hidden Markov models with multiple observation spaces," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 4, pp. 411–416, May 2001.

[68] N. Kumar, "Investigation of silicon-auditory models and generalization of linear discriminant analysis for improved speech recognition," Ph.D. dissertation, John Hopkins Univ., Baltimore, MD, 1997.

[69] H. Hermansky "A statistical approach to metrics for word and syllable recognition," *J. Acoust. Soc. America Proc.*, vol. 66, no. S1, p. S35(A), 1979.

[70] P. F. Brown, "The acoustic-modeling problem in automatic speech recognition," IBM Thomas J. Watson Research Center, Technical Report RC 12750, 1987.

[71] H. Yu and A. Waibel "Streamlining the front end of a speech recognizer," in *Proc. of Int. Conf. of Spoken Language Processing*, Beijing, China, 2000, pp. 353–356.

[72] S. A. Zahorian, D. Qian, and J. Jagharghi, "Acoustic-Phonetic transformations for improved speech recognition," in *IEEE Proceedings of ICASSP*, vol. 1, 1991, pp. 561–564.

[73] X. Aubert, R. Haeb-Umbach, and H. Ney, "Continuous mixture densities and linear discriminant analysis for improved context-dependent acoustic models," in *IEEE Proceedings of ICASSP*, vol. 2, 1993, pp. 648–651.

[74] R. Roth, J. K. Baker, J. M. Baker, L. Gillick, M. J. Hunt, Y. Ito, S. Loewe, J. Orloff, B. Peskin, and F. Scattona, "Large vocabulary continuous speech recognition of Wall Street Journal data," in *IEEE Proceedings of ICASSP*, vol. 1, 1991, pp. 640–643.

[75] O. Siohan, "Acoustic-phonetic transformations for improved speech recognition," in *IEEE Proceedings of ICASSP*, vol. 1, 1995, pp. 125–128.

[76] C. M. Ayer, "A discriminatively derived transform capable for improved speech recognition accuracy," Ph.D. dissertation, ECE Department, University of London, 1992.

[77] G. Yu, W. Russel, R. Schwartz, and J. Makhoul, "Discriminant analysis and supervised vector quantization for continuous speech recognition ," in *IEEE Proceedings of ICASSP*, 1990, pp. 658–688.

[78] L. Wood, D. Pearce, and F. Novello, "Improved vocabulary-independent sub-word HMM modeling," in *IEEE Proceedings of ICASSP*, vol. 1, 1991, pp. 181–184.

[79] N. Campbell, "Canonical variate analysis - a general formulation," *Australian Journal of Statistics*, vol. 26, pp. 86–96, 1984.

[80] T. Hastie, and R. Tibshirani, "Discriminant analysis by Gaussian mixtures," AT&T Bell Laboratories, Technical Report, 1994.

[81] D. P. W. Ellis and J. A. Bilmes, "Using mutual information to design feature combinations," in *Int. Conf. on Spoken Language Processing*, vol. 3, 2000, pp. 79–82.

[82] M. Hasegawa-Johnson, "Time-frequency distribution of partial phonetic information measured using mutual information," in *Int. Conf. on Spoken Language Processing*, Beijing, 2000, pp. 133–136.

[83] D. Koller and M. Sahami, "Toward optimal feature selection," in *Proceedings of the 13th International Conference on Machine Learning (ML)*, Bari, Italy, July 1996, pp. 284–292.

[84] D. Pearce, "Enabling new speech driven services for mobile devices: an overview of the ETSI standards activities for distributed speech recognition front-ends," in *Applied Voice Input/Output Society Conference (AVIOS2000)*, San Jose, CA, May 2000, pp. 237–240.

[85] B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of Markov chains," *IEEE Trans. on Information Theory*, vol. IT 32, no. 2, pp. 729–734, March 1986.

[86] D. Roth, "Learning to resolve natural language ambiguities: A unified approach," in *Proceedings of 15th Conference of the American Association for Artificial Intelligence,* 1998, pp. 806–813.

[87] M. S. Bartlett, *Face Image Analysis by Unsupervised Learning.* Boston: Kluwer Academic Publishers, 2001.

[88] M. K. Omar and M. Hasegawa-Johnson "Approximately independent factors of speech using symplectic maps," *IEEE Trans. on Speech and Audio Processing*, to be published.

[89] M. K. Omar and M. Hasegawa-Johnson "Model enforcement: A unified information-theoretic feature transformation for classification and recognition," *IEEE Trans. on Signal Processing*, to be published.

[90] T. M. Cover, and J. A. Thomas, *Elements of Information Theory.* New York, NY: Wiley, 1997.

[91] H. V. Poor, *An Introduction to Signal Detection and Estimation.* New York: Springer-Verlag, 1994.

[92] V. N. Vapnik, *Statistical Learning Theory.* New York, NY: Wiley, 1998.

[93] T. K. Moon, "The expectation maximization algorithm," *IEEE Signal Processing Magazine*, pp. 47–60, November 1996.

[94] T. F. Quateri, *Discrete-Time Speech Signal Processing Principles And Practice.* Upper Saddle River, NJ: Prentice Hall, 2002.

[95] S. Axelrod, R. A. Gopinath, and P. Olsen, "Modeling with a subspace constraint on inverse covariance matrices," in *Proc. of Int. Conf. of Spoken Language Processing*, Denver, Colorado, 2002, pp. 2177–2180.

[96] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* New York: McGraw-Hill, 1991.

[97] R. Abraham, J. E. Masden, *Foundation of Mechanics.* Redwood City, CA: The Benjamin/Cummings Publishing Company, 1978.

[98] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, Boston: Kluwer Academics, 1998.

[99] L. Parra, G. Deco, and S. Miesbach, "Statistical independence and novelty detection with information preserving nonlinear maps," *Neural Computation*, vol. 8, pp. 260–269, 1996.

[100] R. J. P. De Figueiredo, "Implications and applications of Kolmogorov's superposition theorem," *IEEE Trans. on Automatic Control*, pp. 1227–1230, 1980.

[101] A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems.* Washington D. C.: Winston and Sons, 1977.

[102] V. N. Vapnik and A. Y. Chervonenkis, "Necessary and sufficient conditions for consistency of the method of empirical risk minimizatition," *Pattern Recognition and Image Analysis*, vol. 1, no. 3, pp. 284–305, 1991.

[103] T. Evgeniou, M. Pontil, and T. Poggio, "A uified framework for regularization networks and support vector machines," in *Advances in Large Margin Classifiers*, Cambridge, MA: MIT Press, 1999.

[104] V. N. Vapnik, *The Nature of Statistical Learning Theory* New York: Springer-Verlag, 2000.

[105] A. Smola, "Learning with kernels," Ph.D. dissertation, Technische Universitat Berlin, 1998.

[106] D. P. Bertsekas, *Nonlinear Programming.* Belmont, MA: Athena Scientific, 1999.

[107] L. Rabiner, and B.-H. Junag, *Fundamentals of Speech Recognition.* Upper Saddle River, NJ: Prentice-Hall, 1993.

[108] F. Girosi, "An equivalence between sparse approximation and support vector machines," *Neural Computation*, vol. 10, no. 6, pp. 1455–1480, 1998.

[109] J. P. Leblanc and P. L. De Leon "Speech separation using kurtosis maximization," in *IEEE Proceedings of ICASSP*, Seattle, Washington, 1998, pp. 1029–1032.

[110] A. K. Halberstadt and J. R. Glass, "Heterogeneous measurements and multiple classifiers for speech recognition," in *Proc. of Int. Conf. of Spoken Language Processing*, Sydney, Australia, 1998, pp. 1379–1382.

[111] S. Young, "The general use of tying in phoneme-based HMM speech recognition," in *IEEE Proceedings of ICASSP*, San Francisco, CA, 1992, pp. 569–572.

[112] K.-F. Lee and H.-W. Hon, "Speaker independent phone recognition using hidden Markov models," *IEEE Trans. on ASSP*, vol. 37, no. 11, pp. 1641–1648, November 1989.

[113] A. Robinson, "An application of recurrent nets to phone probability estimation," *IEEE Trans. on Neural Networks*, vol. 5, no. 2, pp. 298–305, March 1994.

[114] S. Young, "Large vocabulary continuous speech recognition," *IEEE Signal Processing Magazine*, vol. 13, no. 5, pp. 45–57, 1996.

[115] S. Young and P. Woodland, "State clustering in hidden Markov model continuous speech recognition," *Computer, Speech, and Language*, vol. 8, no. 4, pp. 369–383, October 1994.

[116] M. K. Omar and M. Hasegawa-Johnson, "Non-Linear independent component analysis for speech recognition," in *The 9th International Conference on Information Systems, Analysis, and Synthesis*, Orlando, Florida, July 31-August 2, 2003, pp. 204–209.

[117] L. C. Parra, *Symplectic Nonlinear Component Analysis,* in *Advances in Neural Information Processing Systems,* Vol. 8, Cambridge, MA: MIT Press, 1996, pp. 437–443.

[118] X. D. Huang, Y. Ariki, and M. A. Jack *Hidden Markov Models For Speech Recognition.* Edinburgh, UK: Edinburgh University Press, 1990.

[119] H. A. David, *Order Statistics,* New York: Wiley, 1981.

[120] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communications*, vol. 28, pp. 84–94, Jan. 1980.

[121] B. Kingsbury, L. Mangu, G. Saon, G. Zweig, S. Axelrod, V. Goel, K. Visweswariah, and M. Picheny, "Toward domain-independent conversational speech recognition," in *Proc. of Eurospeech*, 2003, pp. 1881–1884.

[122] A. Janin et al., "The ICSI meeting corpus," in *Proc. ICASSP*, 2003, pp. 364–367.

[123] M. Padmanabhan, G. Saon, J. Huang, B. Kingsbury, and L. Mangu, "Automatic speech recognition performance on a voicemail transcription task," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 7, pp. 433–442, October 2002.

[124] G. Saon, G. Zweig, B. Kingsbury, L. Mangu, and U. Chaudhari, "An architecture for rapid decoding of large vocabulary conversational speech," in *Proc. of Eurospeech*, Geneva, Switzerland, 2003, pp. 1977–1980.

[125] K. N. Stevens, *Acoustic Phonetics.* Cambridge, MA: MIT press, 1998.

[126] M. K. Omar and M. Hasegawa-Johnson, "Strong-sense class-dependent features for statistical recognition," in *Proc. of IEEE Statistical Signal Processing Workshop*, St. Louis, 2003, pp. 473–476.

[127] M. K. Omar and M. Hasegawa-Johnson, "Non-linear maximum likelihood feature transformation for speech recognition," in *Proc. of Eurospeech*, Geneva, Switzerland, 2003, pp. 2497–2500.

[128] Y. D. Rubinstein, and T. Hastie, "Discriminative vs. informative learning," in *Proc. of Knowledge Discovery and Data Mining*, 1997, pp. 49–53.

[129] M. Loog and R. Haeb-Umbach, "Multi-class linear dimension reduction by generalized Fisher criteria," in *Proc. of Int. Conf. of Spoken Language Processing*, Beijing, China, 2000, vol. 2, pp. 1069–1072.

VITA

Mohamed Kamal Mahmoud Omar received his bachelor's degree in communication and electronics engineering with distinction and degree of honor in 1995 from Cairo University, Egypt. He received his master's degree in communication and electronics engineering in 1999 from Cairo university. He is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign (UIUC).

From 1995 to 1997, he was a fellow in the national research center (NRC), Cairo. From 1997 to 1998, he was a research engineer in the research and development international (RDI) center, Egypt. From 1998 to 1999, he was a development engineer in the Mentor Graphics subsidiary in Egypt. In September 1999, he joined the computer science faculty at Ain-Shams university, Cairo, as a research and teaching assistant. Since August 2000, he has been a research assistant in the Beckman Institute at UIUC.

He was awarded the Cairo University medal in 1995 as a distinguished graduate, and in 1999 for his master's thesis. He is the recipient of the 2002 H. R. Perry Fellowship from the Department of Electrical and Computer Engineering at UIUC. His research interests include statistical learning, pattern recognition, and signal processing.