SPARSE BAYESIAN METHODS FOR CONTINUOUS SPEECH RECOGNITION

By

Jonathan E. Hamaker

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computer Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

September 2002

SPARSE BAYESIAN METHODS FOR CONTINUOUS SPEECH RECOGNITION

By

Jonathan E. Hamaker

Approved:

Joseph Picone
Professor of Electrical and Computer Engineering
(Director of Dissertation)

Lois C. Boggess
Professor of Computer Science
(Minor Advisor)

James E. Fowler
Assistant Professor of Electrical and Computer Engineering
(Committee Member)

Eric Hansen
Assistant Professor of Computer Science
(Committee Member)

Jane Harvill
Assistant Professor of Statistics
(Committee Member)

Nicholas Younan
Graduate Coordinator of Computer Engineering in the Department of Electrical and Computer Engineering

James C. Harden
Department Head of the Department of Electrical and Computer Engineering

A. Wayne Bennett
Dean of the College of Engineering

William A. Person
Director of the Graduate School

Name: Jonathan E. Hamaker

Date of Submission: September 13, 2002

Institution: Mississippi State University

Major Field: Computer Engineering

Major Professor: Dr. Joseph Picone

Title of Study: SPARSE BAYESIAN METHODS FOR CONTINUOUS SPEECH RECOGNITION

Pages in Study: 80

Candidate for Degree of Doctor of Philosophy

The prominent modeling technique for speech recognition today is the hidden Markov model with Gaussian emission densities. However, they suffer from an inability to learn discriminative information. Artificial neural networks have been proposed as a replacement for the Gaussian emission probabilities under the belief that the ANN models provide better discrimination capabilities. However, the use of ANNs often results in over-parameterized models which are prone to overfitting. Techniques such as cross-validation have been suggested as remedies to the overfitting problem but employing these is wasteful of both resources and computation. Further, cross-validation does not address the issue of model structure and over-parameterization.

Recent work on machine learning has moved toward automatic methods for controlling generalization and parameterization. A model that has gained much popularity recently is the support vector machine (SVM). SVMs use the principle of structural risk minimization to simultaneously control generalization and performance on the training set. A recent dissertation from this university has employed the SVM in a hybrid

framework for speech recognition. While the HMM/SVM hybrid produced a decrease in the error rate, the implementation had some significant shortfalls which we hope to address in this work. First, the SVMs are not probabilistic in nature and, thus, are not able to adequately express the posterior uncertainty in predictions. This is particularly important in speech where there is significant overlap in the feature space. The SVMs also make unnecessarily liberal use of parameters to define the decision region.

In this dissertation, we study a Bayesian model which takes the same form as the SVM model. This model, termed the relevance vector machine (RVM), provides a fully probabilistic alternative to the SVMs. The RVMs have been found to provide generalization performance on par with SVMs while typically using nearly an order of magnitude fewer parameters. Sparseness of the model is automatic using MacKay's automatic relevance determination methods. In this work we propose to develop the first speech recognition system using RVMs. Similar to hybrid HMM/ANN systems, the RVM model will replace the Gaussian density in the HMM models. To accomplish this, we must develop closed-loop training routines which insure convergence and optimality. Computational issues make this an impossibility currently and must be addressed before a scalable system is feasible.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Inspiration for computer-based speech recognition can be traced back as far as the advent of the Turing test [REF]. The Turing test required that a machine be able to communicate with a human in such a way that the human could not detect that communication was with a computer and not a human. Beginning with the hope for these intelligent computers, scientists and science fiction writers alike have envisioned the ubiquitous "computer" capable of recognizing, understanding and responding to any verbal command given at anytime and anywhere. However, the current state of the technology has left this vision still quite short of realization.

In recent years, automatic speech recognition (ASR) systems have seen increased market penetration in the form of small dialog and dictation applications [REF - market analysis]. However, the pervasiveness has been limited to niche markets or technically savvy users [REF]. A primary reason is that, while speech recognition has been used for small military, intelligence and call center applications for years, its performance on conversational speech is still lacking. Recent NIST evaluations [REF] show that state-of-the-art research systems require gigabytes of memory, running at 100-300 times real-time and still are only capable of a 20-30% error rate [REF]. This dissertation asserts that a reexamination of the technology which underlies ASR systems is necessary. In this view, the goal is to advance state-of-the-art in recognition performance while maintaining reasonable resource consumption.

1.1. Speech Recognition

Spoken communication is the most natural form of information exchange employed by humans. The communication process requires a speaker to encode information into a set of signals (speech production) and a listener to receive those signals (speech perception), recognize (or decode) the components of the signal (often words, as in speech recognition) and infer the implied meaning of the components and take action (speech understanding) [1,2]. The process of human speech recognition often uses a combination of sensory sources including facial gestures, body language, auditory input as well as feedback from the speech understanding facilities to produce an accurate transcription of the speaker's message. In this work, we consider the problem of converting a single sensory stream, the acoustic signal (i.e. the speaker's voice), into a sequence of words. This problem is akin to communicating over the telephone where the other sensory side-information is not available. Henceforth, we will consider this as the *speech recognition problem*.

Underlying the speech recognition problem is a complex pattern recognition system. For computer speech recognition, a system must translate a speech signal into a sequence of words using some model of human speech along with a large amount of contextual information. It is these models of human speech that are particularly interesting from a machine learning standpoint. Typically, we divide these models into three categories: acoustic models, lexical models, and language models. Acoustic models measure the relationship between the speech signal and the perceived linguistic subunits (phonemes, syllables, etc.). Lexical models describe how the linguistic subunits can be combined to create words — this is often termed as pronunciation modeling. Language models describe how words can be arranged in sequence to create meaningful representations of thought (i.e. sentences or phrases). It is the acoustic models that we are most concerned with in this dissertation.

When building acoustic models, there are two primary model requirements that dictate the type of model which is most appropriate:

1. **Acoustic variation**: Automatic recognition systems must be immune to the acoustic variation that is due to operation in different settings and with different speakers. Primary causes of this variation are differences in speech style, speaker characteristics and channel characteristics.

   - The style of speech is closely related to the target of the speech. Read speech is often slowly spoken and carefully articulated. Further, read speech usually follows tightly constrained grammatical rules. On the other hand, modeling of conversational speech suffers from the effects of poor articulation and ungrammatical phrasing.
   - The variation due to speaker characteristics is a confounding factor in building speech recognition systems. Thus, we typically classify systems as either

speaker-dependent (they are only appropriate for the modeling of a single, known speaker) or speaker-independent (they are appropriate for modeling any speaker). There is typically a very large degradation in performance when modeling with a speaker-independent system compared to a speaker-dependent system [REF]. Likewise, there is typically a very large increase in model complexity for speaker-independent systems [REF]. Adaptive systems have been proposed to counter this problem by adapting the speaker-independent models to a speaker-dependent model as more data from the target speaker is obtained [REF].

- The channel characteristics can vary due to either the input device or the ambient environment. Typical input devices include stationary microphones, close-talk headset microphones and telephones (cellular and land-based). Each has peculiar acoustic characteristics that must be normalized as part of the modeling process. Further, each contributes noise to the system which must be effectively eliminated to assure robust modeling. The ambient environment contributes noise to the system as well — for example, automobile noise in a car cell phone application. Much of the normalization and noise robustness is the domain of the feature extraction unit. However, the acoustic model must also be robust to minor variations (and some major variations) in the channel characteristics. Recognition of speech in noisy environments has obvious applications (e.g. battlefield scenarios [REF], cell phone scenarios [REF] and automobile scenarios [REF]) which are under intense research today.

2. **Temporal variation**: It is the change in the articulators over time that produces the speech patterns that we wish to model [REF]. Thus, acoustic models must be capable of modeling temporal (time-varying) sequences. Typically, however, we constrain this problem so that the model is responsible for making a decision over of a fixed interval of the continuous speech signal. This is well-motivated by the empirical observation that the human articulators change at a rate much slower than the signal sampling rate [REF]. This method is known as frame-based processing and is pervasive in modern speech recognition techniques.

Ideally, we would like to build a system based on expert knowledge of the human communication process to explain all of the variation in speech phenomena. Unfortunately, this is not a feasible approach. One need only observe the wealth of new linguistic theory published each year to see that the variation in the speech communication process is still not completely understood by scientists. Thus, we avoid the need to build an expert system by using self-organizing statistical approaches that can smooth over the unexplained variation in the data.

1.2. Statistical Speech Recognition

Computer speech recognition is typically posed as a statistical pattern recognition problem based on Bayesian methods [REF]. Given a set of acoustic observations, $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$, and a set of models describing acoustic and linguistic patterns, we must find the most probable word sequence [REF]:

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}} \; P(\boldsymbol{O}|\boldsymbol{W})P(\boldsymbol{W}) \tag{1}$$

where $P(\boldsymbol{O}|\boldsymbol{W})$ is the probability that the acoustic observations would be seen when a particular word sequence was spoken and $P(\boldsymbol{W})$ is the *a priori* probability of the word string $\boldsymbol{W}$ being spoken. $P(\boldsymbol{W})$ provides for the language model component described earlier. This statistical process is described in Figure 1.

The focus of this paper is the computation of $P(\boldsymbol{O}|\boldsymbol{W})$ which corresponds to the acoustic modeling component. In most state-of-the-art recognition systems, a hidden Markov model (HMM) is used as the acoustic model [REF]. The popularity of the HMM representation [REF] is based on its ability to simultaneously model the temporal progression of speech (speech is usually seen as a “left-to-right” process) and the acoustic variability of the speech observations. The temporal variation is modeled via an underlying Markov process while the acoustic variability is modeled by an emission distribution at each state in the Markov chain. The most commonly used emission distribution is the Gaussian mixture model (GMM).

While combinations of HMMs and Gaussian mixture models have been extremely successful, there are two fundamental limitations of these approaches:

- the parametric form of the underlying distribution is assumed to be Gaussian,
- the maximum likelihood (ML) approach, which is typically used to estimate model parameters, does not explicitly improve the discriminative ability of the model.

The ML approach maximizes the probability of the correct model while implicitly ignoring the probability of the incorrect model. Ideally, a training approach should force the model toward in-class training examples while simultaneously driving the model away from out-of-class training examples.

1.3. Discriminative Systems

Methods such as maximum mutual information [21,22] have been developed to incorporate discriminative training directly into the standard HMM/GMM framework.

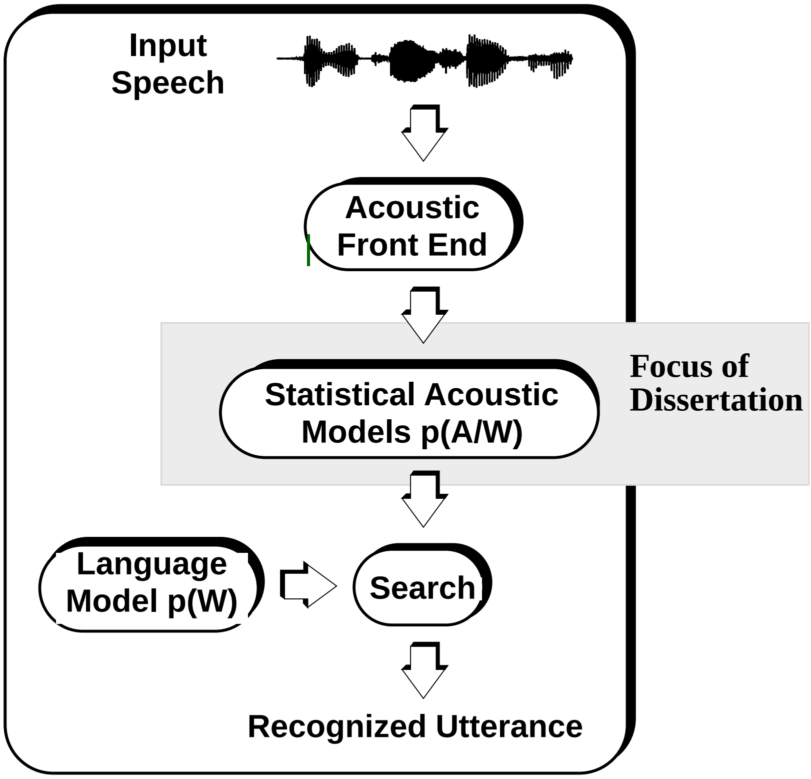


Figure 1. Block diagram analysis of a typical speech recognition system which forms an overall pattern recognition problem. Trained acoustic and language models provide constraints to a search engine whose task is to recognize the utterance that was spoken by the speaker given the constraints and the input.

MMI training simultaneously maximizes the probability of the observation sequence given the reference transcription while minimizing the probability of the observation sequence given any other transcription. Initially MMI training was impractical but recent innovations have reduced the computational load by incorporating approximations in the optimization. Most state-of-the-art HMM/GMM research systems use MMI training [REF].

The weaknesses of the HMM/GMM system have also led researchers to explore other models [23-27], such as hybrid connectionist systems [26-30], which merge the power of artificial neural networks (ANNs) and HMMs. HMM/ANN hybrids have shown promise in terms of performance but have not yet found widespread use due to some serious problems. First, ANNs are prone to overfitting the training data if allowed to blindly converge. To avoid overfitting, a cross-validation set is often used to define a stopping point for training. This is wasteful of data and resources — a serious consideration in speech where the amount of labeled training data is limited. ANNs also typically converge much slower than HMMs. Most importantly, the HMM/ANN hybrid systems have not shown the substantial improvements in recognition accuracy over HMM/GMM systems that would be necessary to force a paradigm shift.

The approaches developed in this work draw significantly from the HMM/ANN hybrid systems. However, we seek methods which are automatically immune to overfitting without the artificial imposition of a cross-validation set as well as methods which can automatically learn the appropriate model structure as part of the overall optimization process. One such model that has come to the forefront of pattern recognition research is the support vector machine (SVM) [37,38,39]. The support vector paradigm is based upon structural risk minimization (SRM) in which the learning process is posed as one of optimizing some *risk function*, $R(\alpha)$, where α are the free system parameters. In the support vector paradigm, the risk function, the empirical error and the parameterization of the model are closely related. In other words, the structure of the model, itself, can be optimized as part of the overall learning process.

In its simplest form, the support vector machine is a maximum margin classifier. That is to say that it tries to find a hyperplane that separates two classes of data such that the resulting distance between the two class boundaries and the hyperplane is maximal. In doing so, Vapnik and others [REF] have shown that this technique can optimally balance empirical risk (or training set error) and generalization error. The most important feature of SVMs is their ability to work in a high-dimensional space by incorporating kernels into the linear SVM framework. The kernels allow one to cast the data into a very high dimensional space and find the optimal hyperplane in that high-dimensional space through a functional evaluation providing for a nonlinear classifier in the feature space. Sparseness in the SVM framework is achieved because only the vectors on the boundary of the classes or in the overlap regions are needed to define the margin. Thus, many of the vectors become unnecessary in the definition of the model and can be pruned.

When there is overlap in the training data, the problem becomes more difficult. The trainer now has to minimize empirical risk, but is rarely able to drive the empirical risk to zero. Instead, one must set a trade-off parameter which balances empirical risk and the number of parameters in the optimization. In the limit, one could set all vectors in the overlap region as support vectors to assure perfect classification, but this comes at a large cost in terms of model complexity. It is debatable that such a binary classifier is even appropriate for a problem (such as speech) with severe overlap in the feature space. This critique is addressed in this dissertation.

Ganapathiraju and colleagues [27,REF SVM work] melded concepts from the SVM modeling approach and the hybrid HMM/ANN systems to define an HMM/SVM hybrid framework. This modeling approach gave improved performance for continuous speech recognition tasks. However, the results also exposed the shortcomings of the SVM framework: the use of a binary decision rule, overly liberal use of parameters, and the need for held-out data sets.

Largest among these drawbacks is that the SVM uses a binary decision rule rather than a soft decision rule. It is usually desirable to not only classify each example as belonging to one class or another but to also know the degree of confidence in that assignment (or the probability of that assignment being correct). This is particularly true for speech recognition problems where there is significant overlap in the feature space. We often have a search space full of competing hypotheses which must be rank-ordered so that we may choose the best one. Binary classifiers provide no direct means to make this choice.

SVMs produce a distance which may be seen as a rough measure of confidence. However, for unseen data that does not match well to the training set, the distance can be a misleading indicator. Ganapathiraju and others have used sigmoids [REF,REF] to map the distance to a posterior probability. However, experimentation on speech corpora has shown that the quality of the posterior is pretty poor. In fact, a simple binary output of 0 for out-of-class or 1 for in-class has nearly equivalent performance.

A second drawback of SVMs is the relatively large size of the models. While more sparse than a fully expanded model, SVMs tend to scale linearly as the size of the training set increases. This is mostly due to the incorporation of vectors in overlap regions as support vectors. The effect can be partially mitigated using heuristic pruning schemes [REF], but the sparseness of SVM models still leaves room for improvement.

Finally, the SVM requires the estimation of trade-off parameters which balance generalization and empirical risk. These are usually estimated using held-out data set which is wasteful of both data and resources. We prefer a method where the accuracy and sparsity of the model are simultaneously optimized. This and the other two drawbacks above are addressed directly by the work presented in this dissertation.

1.4. The Relevance Vector Machine

The relevance vector machine (RVM) introduced by Michael Tipping [REF] is a Bayesian modeling approach that addresses the problems with the SVM, while retaining many of the desirable properties of SVMs. As with SVMs, the basic form of the relevance vector machine is that of a linear combination of basis functions. Also, the RVMs retain the kernel function approach that is used with SVMs. Thus, the advantage of working in a high-dimensional space is maintained when we move to RVMs. A serious complaint with using SVMs is that they are not in the form of a probabilistic model as we are accustomed to using in speech recognition systems. The RVM addresses that complaint by building a fully specified probabilistic model.

The first step in building an RVM model is the definition of a posterior probability model through the use of the logistic link function. This simply maps values on an infinite range to the range [0,1]. This step is common to many types of machine learning including neural networks. What is not so common is the second step or what one might consider the Bayesian step in building our probabilistic model. That is the specification of a prior distribution across the weights that we wish to learn. In this case, we are using an automatic relevance determination prior originally suggested by MacKay [REF]. This prior imposes a zero-mean gaussian distribution across each weight. Each weight has associated with it a hyperparameter that controls the tightness of the prior distribution on that weight. The goal of the RVM method becomes learning the optimal value for this hyperparameter. In practice, a great majority of the hyperparameters converge to infinity, thus creating a zero-valued weight and sparseness is achieved. In fact, RVM models are often one or two orders of magnitude more sparse than SVM models trained on the same data [REF].

There is a caveat to the training procedure for RVMs, however. Central to the method is the computation of an inverse Hessian matrix. This computation requires the inversion of an MxM Hessian matrix where M is initially set to the size of training set. For large training sets, this computation is impractical. This has limited the initial work with RVMs to only include small-scale problems which are of little interest to speech researchers. In this dissertation, we develop and describe techniques to extend the RVM methods to much larger data sets.

1.5. Dissertation Contributions

RVMs have been proven to perform on par with SVM models while maintaining superior sparsity [REF]. However, they are yet to be applied to an extremely complex task. This dissertation describes the first application of RVMs to the complex task of speech recognition. A comparison to SVM-based speech recognition is made and the RVM techniques are found to exceed the performance of the hybrid HMM/SVM system [REF]. The work presented in this dissertation uses Bayesian modeling approaches along with ARD priors to address each of the primary objections to the hybrid HMM/SVM models while preserving the advantages of discriminative modeling.

A significant contribution of this dissertation is the development of training methods for large training sets. Automatic data selection methods have been developed that allow one to determine which training vectors are most likely to contribute to the final model. This leads to one of two reduced set methods developed in this dissertation. The second reduced set method is developed as an incremental training technique where the model is incrementally built without ever having to consider the full training set. These techniques together have increased the training capacity of the RVM approach by two orders of magnitude.

1.6. Dissertation Organization

The remainder of the dissertation begins with a review of current speech recognition acoustic modeling technology. Particular attention is given to parameterization and training of HMM models in Chapter 2. The hybrid neural network systems that serve as inspiration for much of the new work are also discussed. Chapter 2 concludes with a detailed discussion of the weaknesses of both of these modeling techniques. The application of SVMs to data modeling for speech recognition is given in Chapter 3. A thorough review of Ganapathiraju's [27] work with hybrid HMM/SVM models is given. Chapter 3 concludes with a discussion of the inadequacies of the HMM/SVM system. This discussion provides motivation for the work presented in this dissertation.

Beginning with Chapter 4, the theory of sparse Bayesian modeling is developed. The evidence paradigm developed by MacKay [47] and explained in Chapter 4 leads directly to the creation of RVMs by Tipping [44]. Chapter 4 (along with appendices) gives a thorough development of the training paradigm for the RVM along with the underlying assumptions. Chapter 5 describes the practical application of RVMs to speech recognition which was developed as part of this dissertation. This includes both an incremental training paradigm that allows one to apply RVMs to large data sets as well as a data selection method that can be used to train RVMs on large data sets.

Chapter 6 and 7 describe the data and experiments, respectively, used to validate the techniques developed in this dissertation. Both static and dynamic classification tasks are examined. The results are compared to the hybrid HMM/SVM system as well as a baseline HMM system. Chapter 8 concludes the dissertation and includes a discussion of promising avenues for future research. Appendices have been provided, where appropriate, to provide full development of key equations in the dissertation.

CHAPTER 2

STATISTICAL APPROACH TO SPEECH RECOGNITION

In this chapter, we describe the predominant approach to speech recognition. It is a statistical approach and is framed in a maximum likelihood paradigm using hidden Markov models (HMMs) with Gaussian mixture model (GMM) emission distributions to learn the long-range and local phenomena associated with speech patterns. While tremendously successful, a criticism of these systems is that they are not able to adequately model the discriminative information present in the speech signal. Hybrid systems are described which combine the discriminative-modeling power of artificial neural networks and the temporal modeling power of the HMM. The training techniques for these hybrid systems will serve as inspiration for the techniques developed in this dissertation.

2.1. The Speech Recognition Problem

At the heart of computer speech recognition is a pattern recognition problem. It can be stated thusly: given a set of acoustic observations, $\boldsymbol{O} = \boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T$, and a set of models describing acoustic and linguistic patterns, we must determine which patterns were observed and, in doing so, determine which word sequence, $\boldsymbol{W} = w_1, w_2, \ldots, w_M$ was spoken. Four questions quickly arise from this problem statement:

1. How do we obtain the acoustic observations?

2. What model do we use for the acoustic and linguistic patterns?

3. How do we train these models?

4. How do we find the best word sequence when given a new set of observations?

The first of these questions embodies the problem of finding a suitable transformation of the sampled speech signal into a compact feature space which has properties amenable to pattern recognition techniques. The component of a speech system that implements the transformation is the acoustic front end. Volumes have been written on front end processing (for example see [4,5]), however, a fairly generic frame-based, cepstral front end depicted in Figure 2 is at the core of most acoustic front ends for speech

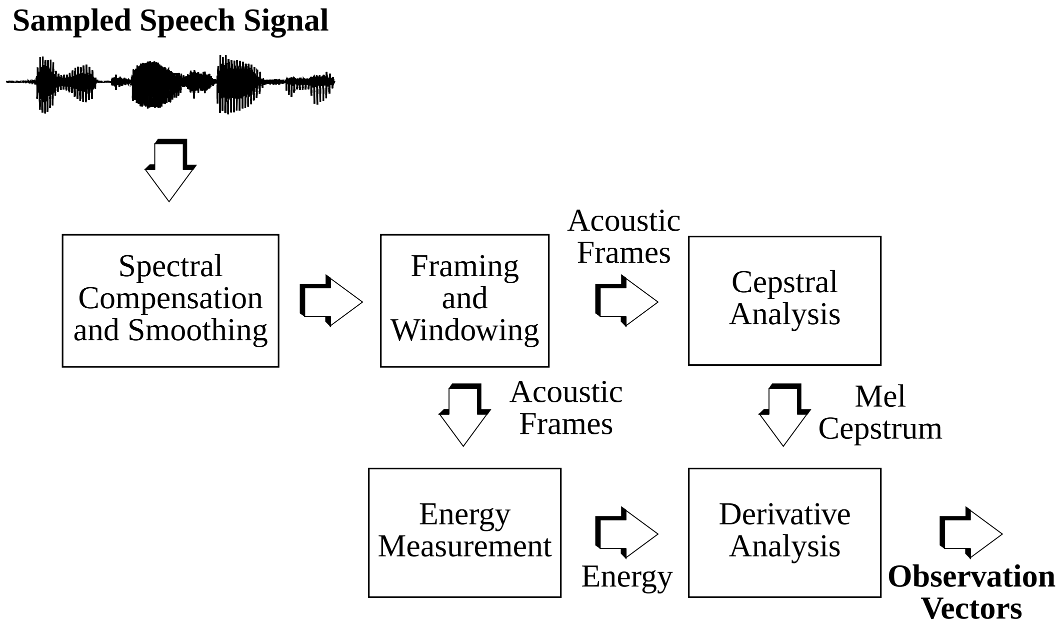


Figure 2. Typical Mel-Cepstral acoustic front end.

recognition and is used in this work [6]. While this front end is not the only possibility (see, for example [7]), it has been widely used in speech recognition applications.

At the core of the cepstral front end is a frame-based analysis which gives a short-time analysis of the sampled speech signal [4]. Under the assumption that the speech signal is stationary over short periods, a frame duration on the order of 10 milliseconds is commonly used. The frame-based approach allows us to analyze the signal in terms of its short-term frequency content. Mel-scale cepstral analysis (MFC) [6] is performed to provide a compact representation of the vocal tract impulse response. The measured cepstral response is correlated with the shape of the vocal tract and position of the articulators at the time at which the frame of speech was uttered. While the frame-based analysis assumes stationarity, it is an unrealistic assumption. Articulators do not instantaneously switch position at frame boundaries, nor are they completely motionless during the frame's duration [8]. To account for some of the transitory behavior, first and second derivative features are typically appended to the feature vector.

With the acoustic observations in place, we can address the second question from above: what model of the acoustic and linguistic patterns do we use? Speech can be loosely seen as a concatenation of units embedded in a hierarchy as shown in Figure 3. For example, we might say that speech is a concatenation of sentences which are, in turn, a concatenation of words which are a concatenation of syllables which, finally, are a concatenation of phones. The phone is often considered to be the smallest, non-divisible unit of sound. In describing the concatenative model, however, we made a false assumption. In conversational speech it is rarely possible to perceptually isolate a single

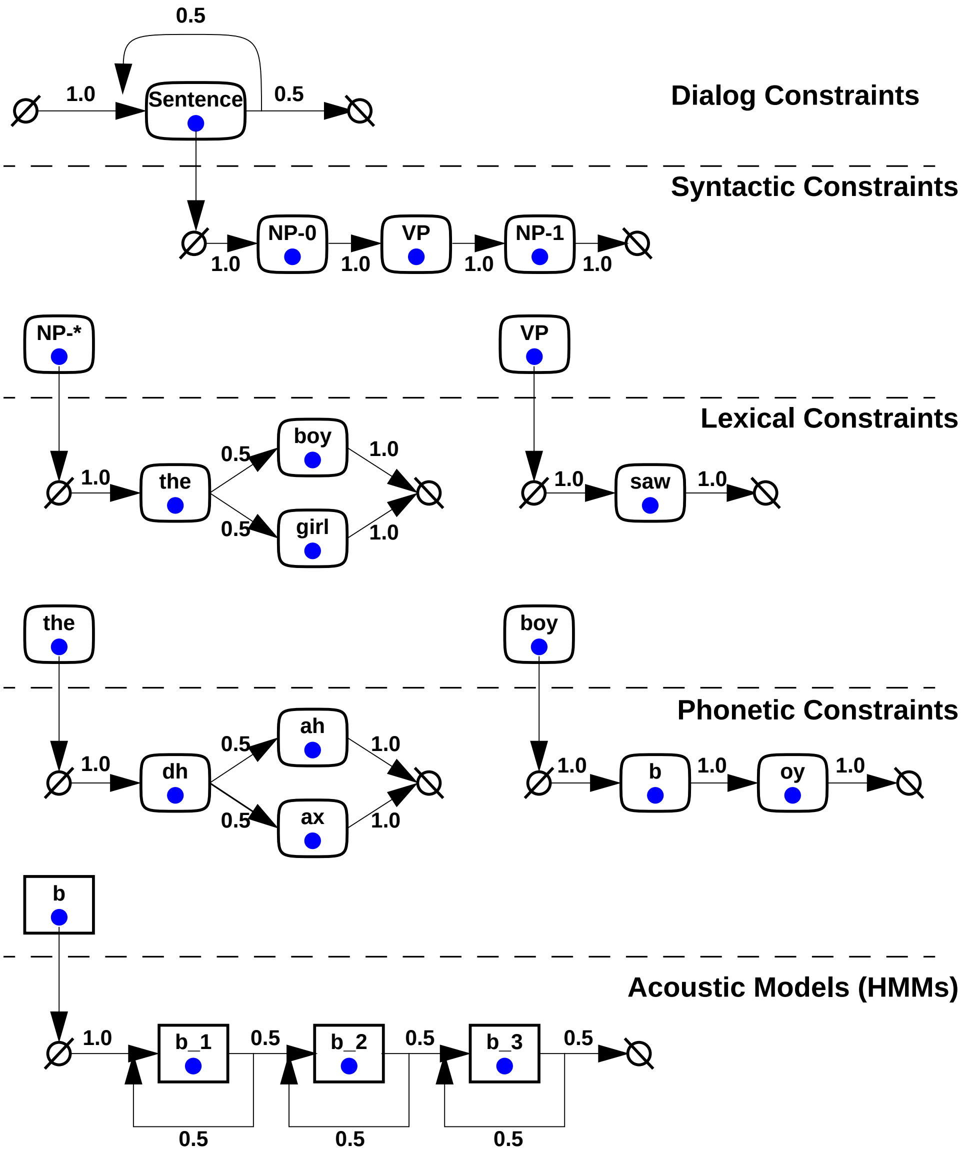


Figure 3. Speech is roughly modeled as a hierarchical constraint system. At each level of the hierarchy, a different knowledge source is applied. The job of a speech recognition system is to combine these knowledge sources in an optimal manner. Often the lowest level in the hierarchy is modeled by hidden Markov models and is responsible for the acoustic match (i.e. modeling the observations sequences generated by the acoustic front end).

phone. Rather, our perception of a phone is formed from the surrounding phonemic context [9]. For example the ‘a’ sound in the words “am” and “apple” differ — the proximity of the nasal sound, ‘m’ causes the ‘a’ in “am” to be nasalized. This type of effect is particularly prevalent in conversational speech where the speakers are seldom cautious in their articulation [10].

To model these coarticulation effects, we use a context-dependent model in which the model for a base sound is dependent upon the surrounding context. In our example above, the ‘a’ in “am” and the ‘a’ in “apple” would be modeled separately. In most speech applications, a single left context phone and a single right context phone modify the phone in question. This unit is known as a triphone and tends to lead to large increases in performance [11]. Larger contexts have also been applied with some smaller increases in performance [12]. Coarticulation at word boundaries is also a major problem in conversational speech. These effects are modeled by cross-word, context-dependent models.

Speech recognition requires choosing amongst many different possible transcriptions. This requires that we have some principled manner for directly comparing candidate transcriptions so that the “best” one may be chosen. Probabilistic modeling is a natural and very common comparison paradigm and provides our answer to the fourth question above as well: how do we find the best word sequence given a new set of observations. We can reformulate the speech recognition problem as a probabilistic one where we want to find the word sequence, $\hat{\boldsymbol{W}}$, that is most probable given the acoustic observations, $\boldsymbol{O}$:

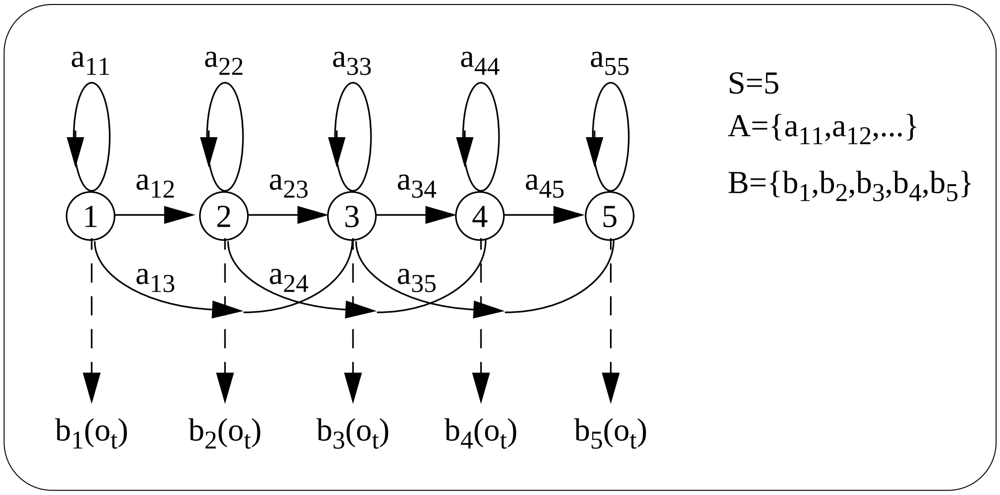


Figure 4. A simple HMM featuring a five state topology with skip transitions. Each state has a stochastic emission distribution.

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\text{argmax}} \; P(\boldsymbol{W}|\boldsymbol{O}) . \tag{2}$$

This *a posteriori* formulation gives us no way to apply information about the *a priori* probability of a word string. Thus, we use Bayes' rule to rewrite (2) as

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\text{argmax}} \; \frac{P(\boldsymbol{O}|\boldsymbol{W})P(\boldsymbol{W})}{P(\boldsymbol{O})} \tag{3}$$

where $P(\boldsymbol{O}|\boldsymbol{W})$ is the probability that the acoustic observations would be seen when a particular word sequence was spoken, $P(\boldsymbol{W})$ is the *a priori* probability of the word string $\boldsymbol{W}$ being spoken, and $P(\boldsymbol{O})$ is the *a priori* probability of the acoustic observation sequence occurring. $P(\boldsymbol{O})$ can be safely eliminated from (3) because the observation sequence, $\boldsymbol{O}$, is constant during the maximization. This yields

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\text{argmax}} \; P(\boldsymbol{O}|\boldsymbol{W})P(\boldsymbol{W}) . \tag{4}$$

The terms in (4) are usually modeled separately. $P(\boldsymbol{W})$ is determined by a statistical *language model* which might take the form of a stochastic grammar or an N-gram language model [13,14]. $P(\boldsymbol{O}|\boldsymbol{W})$ is given by an *acoustic model*. This acoustic modeling component of the recognition system is explored in this dissertation. In most state-of-the-art recognition systems, the hidden Markov model (HMM) is used as the statistical acoustic model [15,16,17,18]. The HMM (an example of which is shown in Figure 4) is a doubly-stochastic state machine that can be fully described by the triple $\{S, \boldsymbol{A}, \boldsymbol{B}\}$. Here, S is the number of states in the machine, $\boldsymbol{A} = \{a_{ij}\}$ is the state-transition probability set, and $\boldsymbol{B} = \{b_j(\boldsymbol{o}_t)\}$ is the emission probability distribution.

The popularity of HMMs as a model of speech phenomena is owed to the HMMs ability to simultaneously model the temporal progression of speech (speech is usually seen as a "left-to-right" process) and the acoustic variability of the speech observations. The temporal variation is modeled via an underlying Markov process while the emission distribution models the acoustic variability. This acoustic variability may come as a result of differing speakers, channel conditions, stress levels, dialect, accent, etc. in the speech training corpus. The most commonly used emission distribution is the Gaussian mixture model (GMM) described by

$$b_j(\boldsymbol{o}_t) = \sum_{i=1}^{K} C_{ij} N(\boldsymbol{o}_t|\mu_{ij}, \Sigma_{ij}), \quad \sum C_{ij} = 1 \text{, where} \qquad (5)$$

$$N(\boldsymbol{o}_t|\mu_{ij}, \Sigma_{ij}) = \frac{1}{\sqrt{(2\pi)^n|\Sigma_{ij}|}} \exp\left(-\frac{1}{2}(\boldsymbol{o}_t - \mu_{ij})^{\mathrm{T}} \Sigma_{ij}^{-1} (\boldsymbol{o}_t - \mu_{ij})\right). \qquad (6)$$

In (5) and (6), the C_i are the mixture weights and define the contribution of each distribution to the total emission score and n is the dimension of the acoustic observation vector. μ_{ij} and Σ_{ij} provide the parameterization of each Gaussian mixture and represent the mean and covariance of the distribution, respectively.

Finally, when building the acoustic models with HMMs, one must decide exactly which acoustic unit (e.g. word, syllable or phone) to use. Most state-of-the-art systems, are based on the cross-word context-dependent phones described earlier. In these systems, each context-dependent phone (usually a triphone) is modeled by an HMM. Figure 3 shows how the HMM fits into the hierarchical model described previously.

2.2. Parameter Estimation

Embodied in (4) is the design for an optimal recognition system. If the prior probability, $P(\boldsymbol{W})$, and the class-conditional probability, $P(\boldsymbol{O}|\boldsymbol{W})$, are known then we simply need to choose the word sequence that maximizes (4). However, it is rare that these distributions are known for a practical problem. Instead, we typically have some small training data set that is drawn from the true distributions as well as some expert knowledge to help guide our decisions. The problem then becomes to use this data and knowledge to estimate the true distributions and then to use those estimates as though they were the true distribution. This problem is non-trivial and is the core of most pattern recognition textbooks [REF Duda,Hart].

Accurately estimating the distributions, itself, is a very difficult task due to the large space of potential solutions and limited data. Increasing the dimensionality of the observation vector only compounds the problem. Making some reasonable assumptions about the form of the distribution and parameterizing those distributions can greatly simplify the problem. For instance, if we can assume the form of $b_j(\boldsymbol{o_t})$ in Figure 4 to be Gaussian then the problem reduces from estimating some unknown distribution to one of estimating the mean and variance of the distribution — a tractable problem even with limited training data. The crux of this dissertation is the comparison what assumptions should be made about the form of the distribution as well as how to estimate the parameters of the assumed distributions.

The answer to the third question above (how do we train these models?), comes from taking an account of the tunable parameters in the probabilistic system described above. These are the language model probabilities, pronunciation model probabilities and the parameters associated with the acoustic model — for the HMM/GMM system, these are the state transitions ($\{a_{ij}\}$), mixture weights (C_{ij}), means (μ_{ij}) and covariances (Σ_{ij}). We ignore the language model and pronunciation model parameters in this dissertation and point the reader to [13, REF Jurafsky]. Here, we concentrate on the acoustic model parameters.

For HMM/GMM systems, the parameters of the system are typically seen as fixed but unknown values that must be estimated. Defining their estimate as the value that maximizes the probability of having observed the training sample leads to the maximum likelihood (ML) approach. In contrast, the SVMs described in Chapter 3 use a risk minimization approach which avoids the estimation of the probability distribution altogether. The RVMs described in Chapters 4 and 5 use a Bayesian approach where, parameters of the system are seen as random variables with known prior distributions.

In the ML approach, the training algorithm must choose, from a family of possible solutions, the parameterization that maximizes the probability of the data given the chosen parameterization. This is illustrated in Figure 5. ML approaches are attractive because they often have very good convergence properties as the size of the training set increases. In fact, the ML method described below for HMM/GMM systems guarantees convergence

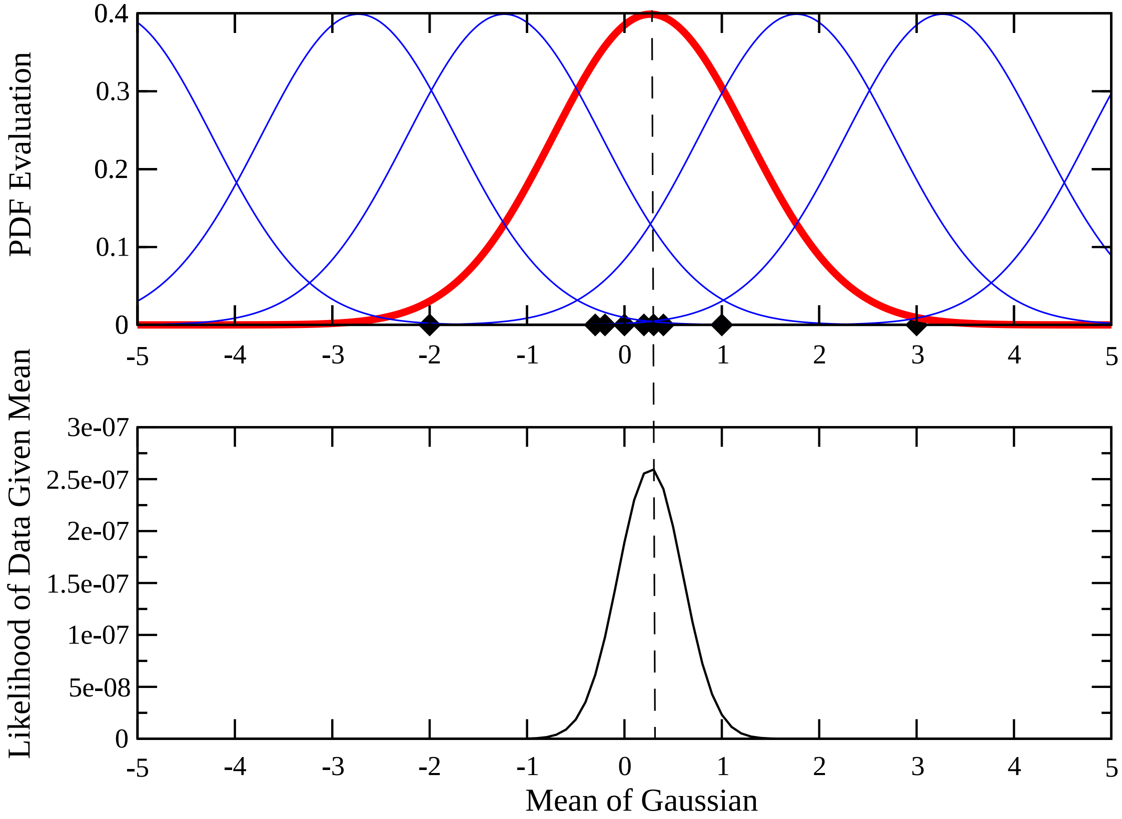


Figure 5. In the top plot, a family of possible solutions are available to model the given data (filled diamonds on the x-axis). Each of the members of the solution family shown are Gaussian densities with a fixed variance. They are parameterized by their mean. The ML approach chooses the parameterization that maximizes the likelihood of the data given the parameterization. The bottom plot shows the likelihood of the data given a chosen mean. The maximum of this likelihood is at the mean of the data (as we would expect). This maximum likelihood parameter specifies the best model which is shown as a bold red curve above.

to a local maximum. Further, ML methods tend to have compact and efficient implementations compared to other methods.

2.3. ML Approaches for HMM/GMM Systems

As with most machine learning tasks, ML training of HMM acoustic models begins with a labeled training data set. This training set consists of speech data and corresponding word transcriptions (sometimes phonetic transcriptions are available as well). However, in speech, there is a complicating factor: the time alignment of the labels to the speech is usually unknown. For instance, we may be given a five-second segment of speech and told that the transcription is “the boy ate candy” (see Figure 6), but we do not know in which time interval each word occurred. Therefore, we can not immediately determine which acoustic observations should be used to train the individual emission

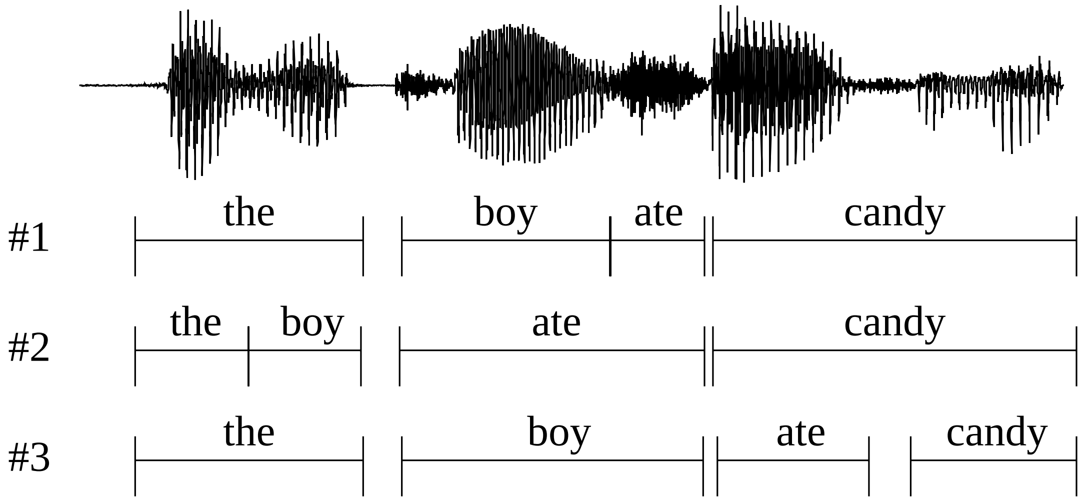


Figure 6. The three alignments above are each candidate "correct" alignments of the words to the data. Typically this alignment is not known *a priori* so it must be inferred as part of the optimization process. For a 5 second utterance with 10 millisecond frames and a four word transcription, there are nearly 10^{17} possible segmentations.

probabilities. This is known as the *segmentation problem*. There are two different techniques that solve the segmentation problem. They differ in whether they use a hard binary segmentation decision or a soft segmentation decision approach.

A simple two-step approach can be taken to overcome the segmentation problem. First, we hypothesize the sequence of HMM states which were most likely to have generated the sequence of acoustic observations given the current parameter set. This is known as a *state-frame alignment*. An example of the state-frame alignment is given in Figure 7. Second, we update the parameter set according to that state-labeled alignment. This is known as Viterbi training [15] because the first step is a Viterbi alignment of the data to the current model. With this procedure, updating of the HMM/GMM parameters is a straightforward computation of the means and covariances for each GMM given the observations [2].

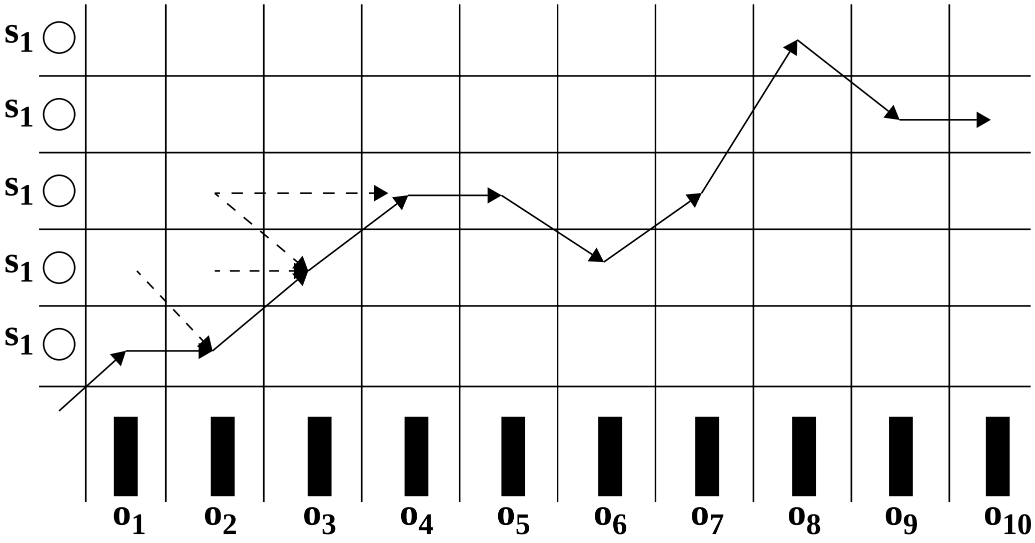


Figure 7. State-frame alignment produced by the Viterbi training process. Note that only one path survives the Viterbi process. All other paths are pruned in the optimization (denoted by dashed lines).

In the Viterbi training paradigm, a binary decision is made as to whether a state occurred. In other words, the *a posteriori* probability that a particular state generated a particular observation is either 0 or 1. While simple to implement, it is questionable whether the current model is sufficiently accurate to warrant a hard binary decision. Further, and more importantly, Viterbi training does not guarantee iterative convergence. In 1972, Baum and colleagues [19] addressed these problems by defining a soft-decision training paradigm where all possible state-frame sequences are considered in the optimization. As will be shown shortly, this algorithm (alternately known as either the Baum-Welch algorithm or the Forward-Backward algorithm) is a special case of the expectation-maximization (EM) algorithm [20] which has guaranteed convergence properties.

In practice, the Baum-Welch training algorithm is implemented in a forward-backward framework [2,16,17]. We define the forward probability, $\alpha_j(t)$, as the probability of having observed the partial observation sequence, $\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t$ and state j at time t and with model parameterization λ':

$$\alpha_j(t) = P(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, q_t = j | \lambda') . \qquad (7)$$

The forward probability at time t can be defined recursively as a function of the forward probabilities at time $t-1$. The backward probability, $\beta_j(t)$, is likewise defined as the probability of observing the partial observation sequence, $\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T$, and state j at time t:

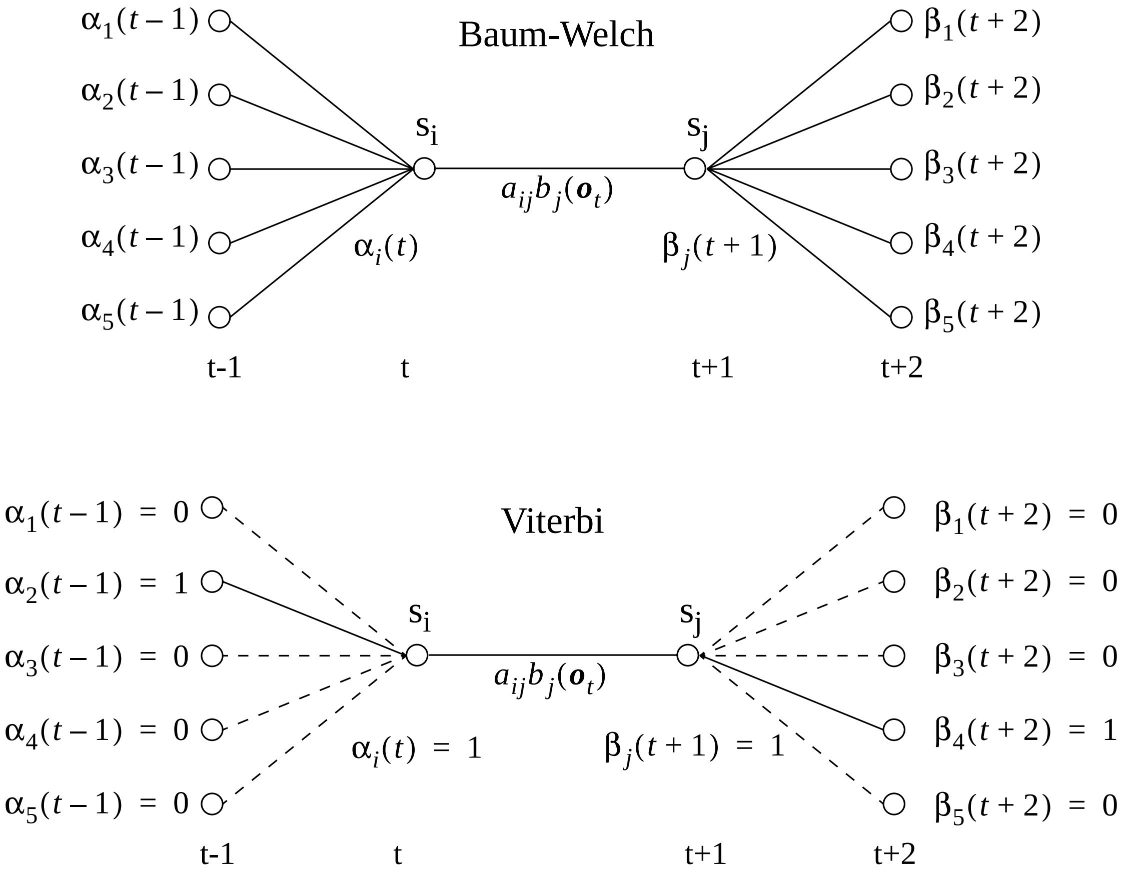


Figure 8. Demonstration of Baum-Welch training and Viterbi training for a single state transition in a five-state, fully-connected model. Baum-Welch uses a soft decision mechanism where all paths at time t-1 are used to estimate the quantities at time t. Viterbi training eliminates all but one path from contention (the pruned paths are indicated by dashed lines above).

$$\beta_j(t) = P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T, q_t = j|\lambda') \tag{8}$$

and can be defined recursively in terms of the backward probabilities at time $t+1$. These two probabilities along with the state transition matrices, $\{a_{ij}\}$, and emission distributions, $b_j(\boldsymbol{o}_t)$ provide all of the information needed to update the parameters [17]. Figure 8 compares Viterbi and Baum-Welch training paradigms. Note that Viterbi training is a special case of Baum-Welch training where the $\alpha_j(t)$ and $\beta_j(t)$ are all unity for the states in the best alignment and zero for all other state alignments.

Though unknown at the time, the Baum-Welch algorithm is actually a specialization of the general expectation-maximization (EM) algorithm formalized by Dempster, et al. [20] in 1977. The goal in development of the EM algorithm is to find a condition under which the observed data, $\boldsymbol{O}$, is more likely under a new setting, λ, of the adjustable parameters than it is under the current setting, λ'. The EM theorem is given below.

EM Theorem: If

$$\sum_{\boldsymbol{q}} P(\boldsymbol{q}|\boldsymbol{O}, \lambda')\log P(\boldsymbol{q}, \boldsymbol{O}|\lambda) > \sum_{\boldsymbol{q}} P(\boldsymbol{q}|\boldsymbol{O}, \lambda')\log P(\boldsymbol{q}, \boldsymbol{O}|\lambda') \qquad (9)$$

then

$$P(\boldsymbol{O}|\lambda) > P(\boldsymbol{O}|\lambda'). \qquad (10)$$

In short, the EM theorem guarantees that, if (9) holds, then the data is more probable under the new parameter setting than other the old.

The EM algorithm, thus, requires that we take the expectation $P(\boldsymbol{q}, \boldsymbol{O}|\lambda)$ with respect to $P(\boldsymbol{q}|\boldsymbol{O}, \lambda')$ and then find the parameter settings $\hat{\lambda}$ which maximize the expectation. $\hat{\lambda}$ then becomes the new setting of the parameters. This implies an iterative process where, if a $\hat{\lambda}$ can be found so that (9) is satisfied, $P(\boldsymbol{O}|\lambda)$ monotonically increases. If a $\hat{\lambda}$ that satisfies (9) can not be found then the current parameter settings represent a local maximum. This monotonic convergence is a highly desirable property and is the key feature that has led to the widespread adoption of the EM algorithm. From (9) one can also see that the expectation is taken over another auxiliary variable, $\boldsymbol{q}$. In the case of HMMs, $\boldsymbol{q}$ is the hidden state transition sequence defined by the soft transition decision in the Baum-Welch algorithm.

The Baum-Welch algorithm satisfies the EM principles by defining an auxiliary function,

$$Q(\lambda', \lambda) = \sum_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda') \log P(\boldsymbol{O}, \boldsymbol{q}|\lambda) . \tag{11}$$

where λ are the new estimates of the system parameters, λ' are the current system parameters, and $\boldsymbol{q}$ is a given state sequence (i.e. a given state-frame alignment). This is equivalent to the left-hand side of (9) where the constant, with respect to $\boldsymbol{q}$, $P(\boldsymbol{O})$ has been multiplied on both sides of (9). Maximizing $Q(\lambda, \lambda')$ with respect to λ insures that

$$Q(\lambda, \lambda') \geq Q(\lambda', \lambda') \tag{12}$$

which, by the EM theorem, implies that $P(\boldsymbol{O}|\lambda) \geq P(\boldsymbol{O}|\lambda')$. Thus, maximizing the auxiliary function monotonically increases the likelihood of the data given the model [2,19,20] until a critical point is reached. Note that the sum over all $\boldsymbol{q}$ in (11) implies a soft decision as to which is the “correct” state-frame alignment. Contrast this to the Viterbi training algorithm where a single alignment was assumed to be the true alignment.

The product of $\alpha_j(t)$ and $\beta_j(t)$ gives the probability of any alignment containing state j at time t

$$P(\boldsymbol{O}, q_t = j|\lambda') = \alpha_j(t)\beta_j(t)\,. \tag{13}$$

Likewise, the total probability of observing the sequence, $\boldsymbol{O}$, is just the marginalization across all states at any time

$$P(\boldsymbol{O}|\lambda') = \sum_{j=1}^{S} \alpha_j(t)\beta_j(t)\,. \tag{14}$$

Finally, we can define the probability of any alignment making a transition from state i to state j while observing $\boldsymbol{o}_{t-1}$ in state i and $\boldsymbol{o}_t$ in state j as

$$P(\boldsymbol{O}, q_{t-1} = i, q_t = j|\lambda') = \alpha_i(t-1)a_{ij}b_j(\boldsymbol{o}_t)\beta_j(t)\,. \tag{15}$$

The above three probabilistic equations amount to the expectation step of the EM algorithm and provide all of the key components for updating the HMM/GMM parameters. With (13), (14) and (15) in place, we can substitute them into the auxiliary function, (11), and maximize with respect to each model parameter. This process defines the maximization step of the EM algorithm which yields the parameter update equations. These update equations are fully derived in [2,13].

2.4. HMM Limitations

HMM/GMM systems have been very successful in recent years. However, one must examine the underlying assumptions of the model to understand how they potentially limit the accuracy of the recognizer. These assumptions are listed and described below.

1. **Conditional independence:** The conditional independence assumption in HMM/GMM systems states that all probabilities in the system are conditionally dependent on the current state and are conditionally independent of all other variables in the system. This assumption is clearly false. First, the probability of an acoustic observation given a particular state is highly correlated with both past and future observations. One could take advantage of this correlation by including surrounding acoustic context in the observation stream — using data transformation techniques such as LDA [REFs] is one approach that is popular. As mentioned earlier, this correlation is partially mitigated by the inclusion of differential parameters in the feature vector [5]. However, the differential features work to circumvent the conditional independence assumption because the differential features in one feature frame are actually defined by features from surrounding frames of data. Ideally, one would also want to condition the distribution itself on the acoustic context, but that is impractical in conventional systems.

2. **Local stationarity:** Using an HMM to model a phone requires one to accept that speech can be appropriately modeled by a Markov chain. This implies that the speech signal is locally stationary — in other words, the system producing the signal is constant over the span associated with each state of the HMM. This assumption is what allows us to use frame-based processing with the HMM. In reality, the articulators are very slow moving over short intervals [REF] but they

are not stationary. Further, noise sources and channel degradation are rarely stationary.

3. **Topology and density assumptions:** In practice the HMM topology (number of states and connections between those states) as well as the form of the observation emission density (number of mixtures and the parametric form of those mixtures) is decided *a priori*. While it should be possible to simultaneously train both the system structure as well as the parameters of the system, only ad hoc procedures have been proposed at this point [REF]. We would prefer modeling paradigms that allow one to optimize the model structure while simultaneously optimizing the accuracy of the model. Likewise, the choice of GMM emission distributions makes assumptions about the parametric form of the underlying distribution which may lead to a poor match with the true underlying distribution. A non-parametric form for the emission distribution which is allowed to adjust to the observed data is preferred.

In addition to these underlying assumptions, the use of maximum likelihood approaches proves to be a limitation of the HMM/GMM system. Maximum likelihood approaches do not improve the discriminative abilities of the model. In other words, the ML approach maximizes the probability of each model separately, using only the data corresponding to that model. The probability of competing, incorrect, models are implicitly ignored. Ideally, the training approach should force the model toward in-class training examples while simultaneously driving the model away from out-of-class training examples. This limitation, in particular, has led to an important area of current research regarding discriminative modeling.

Two approaches have been pursued in building acoustic models with better discriminative capabilities. First are techniques which attempt to construct and train HMM/GMM systems to learn discriminative information. These include maximum mutual information [21,22] and minimum classification error [23] methods that will be discussed briefly below. Second are techniques which attempt to replace the HMM/GMM, altogether, with a new model that is capable of better discrimination performance. The artificial neural network (ANN) hybrid systems [24,25,26,27] detailed below were the pioneers in this research. This dissertation also compares two newer discriminative models, support vector machines and relevance vector machines, that build on the ANN hybrid system approach.

2.5. Discriminative Training of HMM/GMM Systems

In ML training the objective is to optimize the likelihood of the data with respect to the given training word sequence. Specifically, the ML approach ignores all other training word sequences that could have produced the same observed data. As discussed earlier, there is significant overlap in the feature space so the probability of multiple word sequences producing the same observation sequence is non-zero. In 1986, Bahl, et al [21] were among the first to describe discriminative training methods for HMM/GMM speech systems when they introduced maximum mutual information (MMI) training as a replacement for ML training.

The key idea behind the MMI approach is to take into account the probability of both the correct and competing word strings while optimizing the likelihood of the data. In MMI training a new objective function defined as follows. For R training observations and corresponding transcriptions, $\{w_r\}$

$$L_{MMI}(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda(\boldsymbol{O_r}|M_{w_r})P(w_r)}{\Sigma_{\hat{w}} p_\lambda(\boldsymbol{O_r}|M_{\hat{w}})P(\hat{w})} \tag{16}$$

where the subscript λ indicates that all probabilities are taken with respect to the current model set and M_w denotes the model sequence corresponding to the word sequence w. The numerator term is the familiar maximum likelihood objective function estimated with the training word sequence. The denominator, on the other hand, is the probability of the observation sequence accumulated over all other possible word sequences. The optimization of (16) necessitates the maximization of the numerator term (the probability of the correct model sequence generating the data) while simultaneously minimizing the denominator term (the probability of any other model sequence generating the data). In this respect, the MMI criteria is a discriminative training approach.

While the MMI approach was promising, achieving significant recognition gains on small vocabulary tasks [REF], it has taken over a decade for this technique to make its way into use for large vocabulary speech recognition systems. This is primarily due to the rather large computational resource usage in estimating the MMI parameters. The denominator term in (16) is computed over every possible word sequence in the recognition grammar. For all but small fixed grammars this problem is intractable.

Lattice-based methods [REF Valtchev thesis and conference papers] have been introduced that reduce the full recognition grammar to a more manageable subset of the grammar where only the most confusable alternate word sequences are maintained in the lattice. This has largely solved the efficiency problems though a large amount of computer disk space is necessary to hold lattices for large training sets.

MMI training is not the only discriminative technique that has been applied to HMM/GMM models. For example, Ephraim et al. [REF Ephraim, Dembo and Rabiner, 1989] proposed the minimum discrimination information (MDI) technique which attempts to overcome the problem of model mismatch. However, there is no simple implementation of MDI as there is with MMI. Minimum classification error (MCE) which attempts to build and minimize an objective function directly related to the ASR error rate has also been proposed [REF: Juang, Chou, CH Lee Trans SAP 1997] as a replacement for ML techniques. However, MMI has found traction in the research community for two primary reasons. First, there exists a reasonably efficient technique that can be computed completely offline so there is no recognition efficiency overhead due to the model training approach. Second, the MMI technique has produced impressive performance gains — as much as 4% error rate reduction over an ML baseline on Switchboard [REF - HTK 2002 eval presentation].

2.6. Discriminative Modeling Using ANN Hybrids

The weaknesses of the HMM/GMM system have also led researchers to seek new acoustic models to replace the HMM/GMM which mitigate some or all of its limitations [24,25,26,27]. Hybrid connectionist systems which merge the power of artificial neural networks (ANNs) and HMMs have received a particularly large amount of attention from the research community in the past decade as an alternative to HMM/GMM systems [24,25,26,28,29,30]. The primary advantages of using the hybrid HMM/ANN systems in speech are:

1. ANNs are trained discriminatively to learn how to not only accept the correct class assignments but to reject the incorrect class assignments.

2. ANN classifiers are able to learn complex probability functions in high-dimensional feature spaces. GMM systems are usually restricted to smaller dimensional vectors (on the order of 30-50) due to amount of training data that would be necessary in estimating the parameters of the GMM distribution. HMM/ANN system designers have put this to good use by using a longer feature vector consisting of a concatenation of the acoustic observations used in the HMM/GMM system; i.e. $\boldsymbol{o}_t^{ANN} = [\boldsymbol{o}_{t-k}, \ldots, \boldsymbol{o}_{t-1}, \boldsymbol{o}_t, \boldsymbol{o}_{t+1}, \ldots, \boldsymbol{o}_{t+k}]$ [24,26]. Note that this also circumvents the independence assumption since consecutive observations for the ANN system are highly correlated.

While some systems have used ANNs to model both the temporal and acoustic properties of speech [31,32], most of the ANN speech systems have used the ANN as a replacement for the GMM probability distribution and have maintained the HMM as a model of the temporal properties as shown in Figure 9. The outputs of a 1-of-N classifier trained under the mean-squared error criteria are known to approximate the posterior class

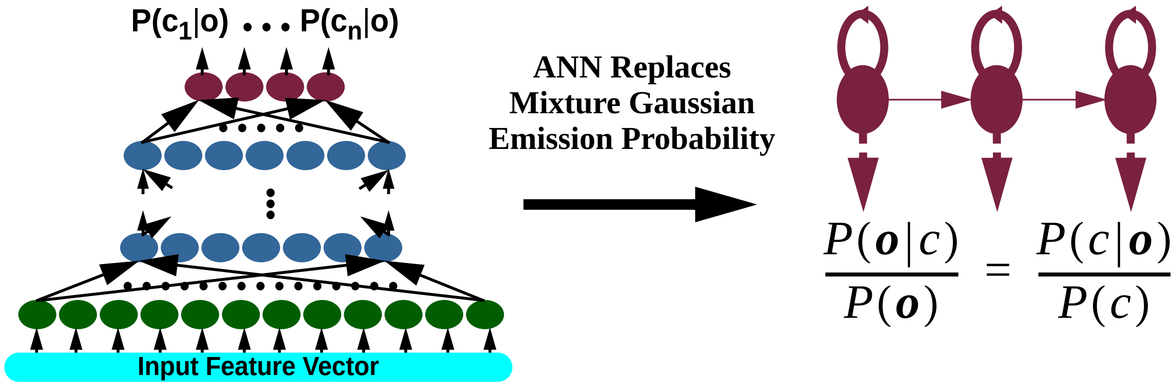


Figure 9. Example of an HMM/ANN hybrid configuration. The ANN acts as a state classifier. Each state of the HMM is assigned to one of the ANN outputs. The ANN posterior classifications are normalized to produce the likelihood emission.

probability, $P(c|\boldsymbol{o})$, where the approximation accuracy is asymptotic in the size of the training set [33]. Recall from the discussion of acoustic modeling earlier that the our goal is to model $\hat{\boldsymbol{W}}$ which maximizes (3). In HMM/GMM systems, we directly build a model of $P(\boldsymbol{O}|\boldsymbol{W})$, but with the ANN systems, we effectively have the posterior phone class probability, $P(C|\boldsymbol{O})$. Thus, the posterior class probabilities need to be converted to likelihoods using Bayes' rule

$$\frac{P(c|\boldsymbol{o})}{P(c)} = \frac{P(\boldsymbol{o}|c)}{P(\boldsymbol{o})}. \qquad (17)$$

In practice, the *a priori* class probabilities are estimated from the training data [24,29].

Using (17), the ANN can be used as a direct substitute for the GMM in the HMM framework. Thus, it makes sense that they could/should be trained in the same manner. Initially the hybrid systems were trained using a Viterbi (hard decision) training paradigm as described for HMM/GMM systems above [24,29]. The HMM/ANN system with the current ANN probability estimators was used to create a single alignment of the acoustic observations to the HMM states. The ANN posterior estimators were then trained on each observation that aligned to the HMM state using a typical ANN training algorithm such as back propagation. Parallel training methods were pursued due to the resource-intensive nature of ANN training [34]. Because ANNs are prone to overfitting, a held-out cross-validation set is necessary to test for convergence of the models to a local maxima.

It is well known that, with infinite training data and sufficient model complexity, a neural network trained on binary (0/1) targets will learn the posterior probability distribution perfectly [33]. However, it is less clear how the same ANN will perform when the training data is limited and the model topology is not matched to the true posterior distribution. Yan, et al. [35] claim that, when given unseen data, an ANN trained under such circumstances will produce unreasonable output. An appropriate response would be to make a probability estimate which displays a lack of posterior knowledge about the correct classification (a uniform probability for all classes, for instance). Instead, the ANNs often make extremely confident predictions despite the lack of any prior training which supports the prediction. To address this issue, researchers have recently begun to explore the use of the Baum-Welch framework as a method for training HMM/ANN hybrids [35,36]. The goal of this method of training the HMM/ANN system is to train the ANN to learn the posterior emission probability distribution from the targets that are readily available from the Baum-Welch procedure:

$$\gamma_j(t) = P(q_t = j|\boldsymbol{O}, \lambda) = \frac{\alpha_j(t)\beta_j(t)}{\sum_{k \in \boldsymbol{S}} \alpha_k(T)}. \tag{18}$$

The ANN is then directly trained on these $\gamma_j(t)$ values.

The HMM/ANN hybrids have shown promise in terms of performance but have not yet found widespread use due to some serious problems. ANNs are prone to overfitting the training data if allowed. To avoid overfitting, a cross-validation set must be used to define a stopping point for the training set. This is wasteful of data and resources — a serious consideration in speech where the amount of labeled training data is very limited. ANNs also typically converge much slower than HMMs. Most importantly, the HMM/ANN hybrid systems have not shown substantial improvements in recognition accuracy over HMM/GMM systems.

2.7. Summary

This chapter has reviewed the most common acoustic modeling framework for speech recognition systems — HMMs with GMM emission probability distributions. Lack of discriminative capabilities in ML-trained HMM/GMM systems is a large limitation that has led to a wealth of research. This chapter has discussed the use of discriminative training techniques including MMI training. The use of ANNs as replacements for the GMM distributions has also been discussed.

Of particular importance in this chapter are the training techniques used in the HMM/GMM systems and the hybrid HMM/ANN systems. The relevance vector machines explored in this dissertation will act in a fashion similar to the ANNs as posterior estimators. Thus, the approaches developed in this dissertation will draw significantly from the HMM/ANN work. However, we will seek methods which are automatically immune to overfitting without the artificial imposition of a cross-validation set as well as methods which can automatically learn the appropriate model structure. The next two chapters define such methods, the support vector machine and relevance vector machine, which both describe principled methods for avoiding overfitting — structural risk minimization for the support vector machine and Bayesian automatic relevance determination for the relevance vector machine.

CHAPTER 3

SUPPORT VECTOR MACHINES FOR SPEECH RECOGNITION

Given a training corpus, $\boldsymbol{O} = \{(\boldsymbol{o}_1, t_1), (\boldsymbol{o}_2, t_2), \ldots\}$ where $\boldsymbol{o}_i$ is the i'th input observation and t_i is the corresponding target (e.g. class assignment or class probability), the goal of a learning machine is to learn the mapping $t = f(\boldsymbol{o})$ under some appropriate optimization scheme. One flexible and popular class of functions are those which are linear combinations of basis functions on the input observations

$$y(\boldsymbol{o};\boldsymbol{w}) = w_o + \sum_{i=1}^{M} w_i\phi_i(\boldsymbol{o}) = \boldsymbol{w}^T\phi(\boldsymbol{o}). \tag{19}$$

A special form of (19) is one in which there is a basis function prescribed for each training vector. These models are generally referred to as *vector machines*. The following chapters discuss and compares two such models: the Support Vector Machine (SVM) [37,38,39,40,41,42,43] and Relevance Vector Machine (RVM) [44,45,46].

3.1. Support Vector Machines

Learning is a process by which a learning machine is optimized under a given set of constraints. We can pose this process as one of optimizing some *risk function*, $R(\alpha)$, where the optimal machine is the one whose free parameters, α, are set such that the risk is minimized. This minimization is written as

$$\hat{\alpha} = \underset{\alpha}{\text{argmin}} \; R(\alpha) = \underset{\alpha}{\text{argmin}} \int Q(\boldsymbol{o}, y, \alpha) dP(\boldsymbol{o}, t) \tag{20}$$

where $Q(\boldsymbol{o}, t, \alpha)$ is a loss function which penalizes the mismatch between both the form and the parameterization of the learning machine and the true function, f; and $P(\boldsymbol{o}, t)$ is the joint distribution of the observations and targets. Finding a minimum for (20) is usually impossible because $P(\boldsymbol{o}, t)$ can not be found *a priori*. Thus, we look for a simplification of (20) that is tractable.

A popular variation of the *actual risk*, $R(\alpha)$, which can be easily evaluated is the measured mean risk, or *empirical risk*, defined as,

$$R_{emp}(\alpha) = \frac{1}{l} \sum_{i = 1 \ldots l} Q(\boldsymbol{o}_i, t_i, \alpha). \tag{21}$$

where l is the number (assumed finite) of training observations. R_{emp} is therefore the loss computed from a fixed training set under the maximum entropy assumption of uniformity for $P(\boldsymbol{o}, t)$. Finding the α which minimizes (21) gives the *empirical risk minimization (ERM)* solution and is one of the most commonly used optimization procedures in machine learning (e.g. mean-square error optimization). However, the issue of the generalization of the learning machine is not specifically addressed when we use ERM. In fact, ERM requires that the training set be representative of the true data distribution to be effective. There could be several settings for the free parameters which give us the same empirical risk. To determine which settings are optimal, we have to know which one would achieve the least actual risk.

Vapnik [37] provides an elegant solution to this problem. Through his analysis of bounds on the actual risk he proved that bounds exist for the actual risk such that,

$$R(\alpha) \leq R_{emp}(\alpha) + f(h) \tag{22}$$

where h is the Vapnik-Chervonenkis (VC) dimension and is a measure of the capacity of a learning machine to learn any training set [37,39] and $f(h)$ is the VC confidence. If $f(h)$ is small (and we have done our job well of fitting the model to the training set), the machine generalizes well because the actual risk is guaranteed to be close to the empirical risk. For binary classifiers where the loss functions are indicator functions, $f(h)$ is defined by

$$\frac{\varepsilon(l)}{2}\left(1 + \sqrt{1 + \frac{4R_{emp}(\alpha_l)}{\varepsilon(l)}}\right) \tag{23}$$

where α_l is the parameter set that defines the learning machine for a particular training set and $\varepsilon(l)$ is the measure of the difference between the actual risk and the empirical risk [49] which we can use to compare system configurations which achieve equivalent empirical risks.

We can write $\varepsilon(l)$ in terms of the VC dimension, h, and the size of the training set, l, as,

$$\varepsilon(l) = 4\frac{h(\log(2l/h + 1)) - \log\eta/4}{l}. \tag{24}$$

From (24), we can see that when l/h is large, ε and $f(h)$ are both small which implies a convergence of the actual risk and the empirical risk [39]. This result matches our intuition that a less complex machine (i.e. one where the capacity is much smaller than the number of training samples) will generalize better than an overly complex machine given that they achieve the same empirical risk. With this result, we can guarantee both a small empirical risk (training error) and good generalization — an ideal situation for a learning machine. The converse property of (24) is also true — when l/h is small, both ε and $f(h)$ are large and good generalization can not be guaranteed.

The principle of *structural risk minimization* (SRM) [37,49] is formulated to find the minimum point on the curve describing the bound on the expected risk. It provides a principled method to trade-off the accuracy of the trained machine and the complexity of the machine. For a fixed training set size, the VC dimension, h, becomes the controlling parameter in l/h. The joint optimization of R_{EMP} and $f(h)$ is not tractable in practical problems. Thus, the principle of SRM is implemented in one of two distinct ways:

1. Fix the VC confidence to an appropriately low value and optimize the empirical risk.

2. Fix the empirical risk to an appropriately low value and optimize the VC confidence.

The support vector methodology [38,39,41,42,43] implements SRM using the latter approach where the empirical risk is fixed at a minimum (typically zero for separable data sets) and the SVM learning process optimizes for a minimum confidence interval. The SRM principle thus orders the solutions which are optimal in the ERM sense. In the next section, the support vector classifiers will be ordered according to the margin between the class boundaries and the separating hyperplane.

Support Vector Classifiers - Margin Maximization

Figure 1 shows a 2-class classification example where the training samples are linearly separable. H_1 and H_2 define two hyperplanes on which the closest in-class and out-of-class examples lie. The distance separating these hyperplanes is defined as the margin between the two classes. SVMs use the SRM principle to impose an order on the optimization process by ranking candidate separating hyperplanes based on the margin. For separable data, the optimal hyperplane is the one that maximizes the margin. The existence of a unique hyperplane that maximizes the margin of separation between the classes is guaranteed [37]. The learning procedure is, thus, tasked with finding the location of the optimal hyperplane.

Following [39], let $\boldsymbol{w}$ be a vector that is normal to the separating hyperplane and let $\{\boldsymbol{o}_i, t_i\}, i = 1, \ldots l$ be the training set of length l where $t_i = \pm 1$ indicates class membership (note that this is a binary classification problem with two class indicators, $+1$ and -1). Since $\boldsymbol{w}$ is a normal (not necessarily a unit normal though) to the separating hyperplane, any point, $\boldsymbol{o}$, lying on the separating hyperplane satisfies

$$\boldsymbol{w} \cdot \boldsymbol{o} + b = 0 \tag{25}$$

where $|b| / \|\boldsymbol{w}\|$ is the perpendicular distance of the hyperplane from the origin. We can require that all of the training samples follow the relations

$$\boldsymbol{o}_i \cdot \boldsymbol{w} + b \geq +1 \qquad \text{for } t_i = +1 \tag{26}$$

$$\boldsymbol{o}_i \cdot \boldsymbol{w} + b \leq -1 \qquad \text{for } t_i = -1 . \tag{27}$$

These can be combined into a single set of inequalities,

$$t_i(\boldsymbol{o}_i \cdot \boldsymbol{w} + b) - 1 \geq 0 \qquad \forall i . \tag{28}$$

Vectors for which the equality condition in (28) holds are known as *support vectors*.

We can require that all points satisfying the equality condition in (26) lie on the hyperplane $H_1\colon \boldsymbol{o}_i \cdot \boldsymbol{w} + b = 1$ with normal vector $\boldsymbol{w}$ and distance from the origin of $|1 - b| / \|\boldsymbol{w}\|$. Similarly, all points satisfying the equality condition in (27) lie on $H_2 : \boldsymbol{o}_i \cdot \boldsymbol{w} + b = -1$ and distance from the origin of $|-1 - b| / \|\boldsymbol{w}\|$. Relating the distance from the origin of each hyperplane, one can see that the distance between the two hyperplanes (which we defined as the margin earlier) is equal to $2 / \|\boldsymbol{w}\|$. Since we are currently only concerned with completely separable data, the margin can be maximized by minimizing $||\boldsymbol{w}||^2$ subject to the constraints of (28). Note that only the support vectors contribute to the SVM solution because it is only those that define the margin. This will become an important property which leads to sparseness in the solution space.

Techniques exist to optimize convex functions with constraints using the theory of Lagrange multipliers [50]. Using these techniques we can pose the functional

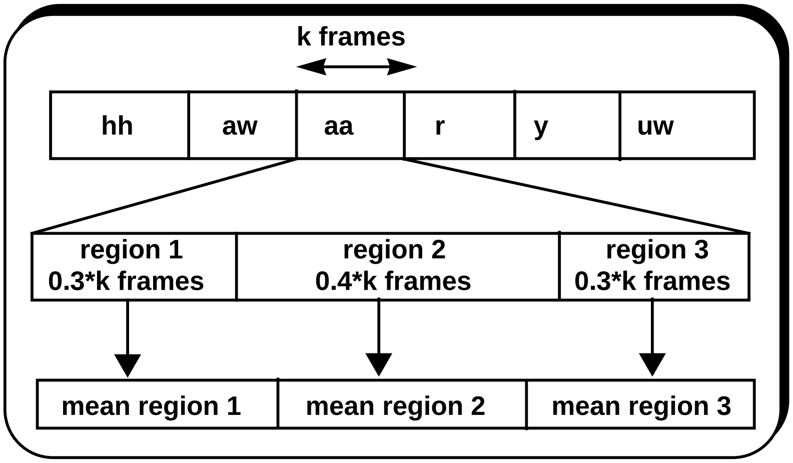


Figure 1. Difference between empirical risk minimization and structural risk minimization for a simple example involving a hyperplane classifier. Each hyperplane ($C0$, $C1$ and $C2$)

$$L_P \equiv \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \alpha_i t_i(\boldsymbol{o}_i \cdot \boldsymbol{w} + b) + \sum_{i=1}^{N} \alpha_i \tag{29}$$

which is called the *primal* formulation of the convex optimization problem. Setting the gradient of L_P with respect to $\boldsymbol{w}$ and b to zero gives

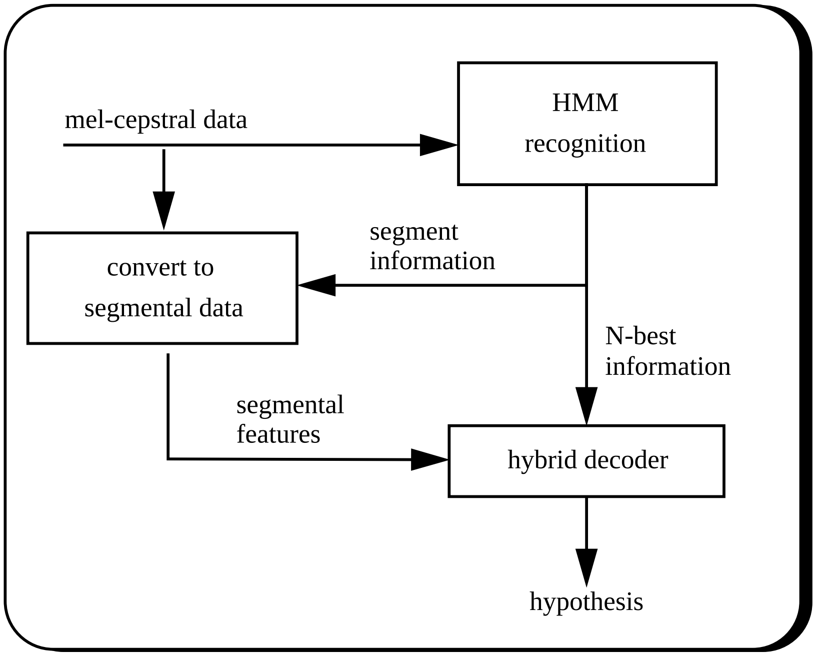
$$\boldsymbol{w} = \sum_j \alpha_j t_j \boldsymbol{o}_j, \text{ and} \tag{30}$$

$$\sum_i \alpha_i t_i = 0. \tag{31}$$

Equations (25) and (30) imply that the decision function can be defined as,

$$f(\boldsymbol{o}) = \sum_{i=1}^{N} \alpha_i t_i(\boldsymbol{o}_i \cdot \boldsymbol{o}) + b \tag{32}$$

where the sign of $f(\boldsymbol{o})$ can be used to classify examples as either in-class or out-of-class. This equation defines the SVM classifier. Notice the correspondence between (32) and



achieves perfect classification and, hence, zero empirical risk. How-

(19): b corresponds to w_0, α_i to w_i, N to M, and $t_i(\boldsymbol{o}_i \cdot \boldsymbol{o}) = \phi_i(\boldsymbol{o})$. The classifier is completely defined in terms of the training examples and the weights. However only those training examples that lie on the hyperplanes, i.e. the support vectors, define the classifier. In practice, the proportion of the training set that becomes support vectors is small making the classifier sparse. Interestingly, the data set itself defines how complex the classifier needs to be thereby defining the lower limit for the VC confidence, $f(h)$ [39].

Kernel Methods for Nonlinear, Non-separable Decision Problems

The preceding analysis has been only for those problems where the data is linearly separable (i.e. a straight line can be drawn that completely separates the two classes of data). Unfortunately, most real-world data does not conform to this prescription. The data ever, $C0$ is the optimal hyperplane because it maximizes the margin

CHAPTER 1

— the distance between the hyperplanes $H1$ and $H2$. Maximizing the margin indirectly results in better generalization.

Figure 2. Composition of the segment level feature vector assuming a 3-4-3

proportion for the three sections.

Figure 3.	Flow graph for hybrid HMM/SVM system [27].

No.	Information Source		HMM		Hybrid	
	Transcription	Segmentation	AD	SWB	AD	SWB
1	N-best	Hypothesis	11.9	41.6	11.0	40.6
2	N-best	N-best	12.0	42.3	11.8	42.1
3	N-best + Ref.	Reference	—	—	3.3	5.8
4	N-best + Ref.	N-best + Ref.	11.9	38.6	9.1	38.1

Table 1. Summary of recognition experiments for hybrid HMM/SVM system [27]. The experiments are differentiated by the corpus (Alphadigits or Switchboard), segmentation type (single segmentation or n-best segmentation) and n-best rescoring type (n-best or oracle n-best + ref). All results are word error rates.

may be nonlinearly separable, or completely inseparable. In either case, we must find a method which optimally bounds the risk while minimizing error on the training set. These problems are attacked with two clever additions to the linear SVM methodology.

In many modeling paradigms, the problem of optimization for non-separable data is solved through the use of soft decision classifiers that place a probability on correctly classifying each training example. However, the SVM is not posed as a probabilistic problem, so we instead introduce the concept of *slack variables* [38]. The hyperplane constraint equations, (26) and (27), become

$$\boldsymbol{o}_i \cdot \boldsymbol{w} + b \geq +1 - \xi_i \qquad \text{for } t_i = +1, \tag{33}$$

$$\boldsymbol{o}_i \cdot \boldsymbol{w} + b \leq -1 + \xi_i \qquad \text{for } t_i = -1, \text{ and} \tag{34}$$

$$\xi_i \geq 0 \qquad \forall i, \tag{35}$$

where ξ‘s are the slack variables (one per input observation) that account for training errors since, for an error to occur, ξ_i must exceed unity. Thus, $\sum \xi_i$ gives an upper bound on the number of training errors [38]. A natural way to control the number of training errors is to assign an extra cost for making an error. This is done through the use of a trade-off parameter, C, which is the penalty incurred by the optimizer for accepting a training error. A large value of C will tend to reduce the number of training errors - often at the cost of a more complex model. C is a user-defined parameter that requires a cross-validation procedure to estimate.

Providing for a nonlinear decision region is accomplished using the *kernel* modeling method [51]. Notice that, in the optimization problem formulated in (29), the only place in which the data appears is in the form of dot products, $\boldsymbol{o}_i \cdot \boldsymbol{o}_j$. If we define a transformation of the data to a higher dimensional space by the function $\phi(\boldsymbol{o})$ then we can still construct optimal margin classifiers if we can evaluate the dot product $\phi(\boldsymbol{o}_i) \cdot \phi(\boldsymbol{o}_j)$. It would be highly advantageous if we could define a *kernel* function, K such that

$$K(\boldsymbol{o}_i, \boldsymbol{o}_j) = \phi(\boldsymbol{o}_i) \cdot \phi(\boldsymbol{o}_j). \tag{36}$$

With this function, the dot product in the high-dimensional space could be computed without having to know the explicit form of $\phi(\boldsymbol{o})$. The decision function, (32), then becomes

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i t_i K(\boldsymbol{x}, \boldsymbol{x_i}) + b. \tag{37}$$

Using this kernel method, the SVM is able to transform the training data to a high-dimensional space and construct a linear binary classifier in that space which maximizes a nonlinear margin in the original space. However, only functions which represent a dot product in some space are eligible as kernel functions. Mercer's condition [37] describes the requirements for a function to be a dot product kernel. If a kernel is used which does not satisfy the Mercer conditions, the quadratic optimization is no longer applicable and may lead to a problem whose solution does not converge. Some commonly used kernels include the polynomial and RBF kernels

$$K_{poly}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y} + 1)^{d} \tag{38}$$

$$K_{RBF}(\boldsymbol{x}, \boldsymbol{y}) = \exp\{-\Upsilon|\boldsymbol{x} - \boldsymbol{y}|^{2}\}. \tag{39}$$

Kernel-based vector machines have had great success on static classification tasks, (those in which no information can be gleaned from the ordering of the exemplars in the input set) for many years (for example [52,53,54,55]) . However, it is only recently that these techniques have been employed on dynamic classification tasks (those in which the ordering of exemplars is in some way informative) [27,56,57,58]. In this dissertation, we are particularly interested in the application of such machines to the speech recognition problem discussed in Chapter 2. In the remainder of this chapter, we detail the first attempt to apply SVMs to the large vocabulary speech recognition problem using a hybrid HMM/SVM system [27,59,60,61,62].

3.2. Support Vector Methods

Initial attempts to add discriminative information to HMM/GMM speech recognition systems used maximum mutual information (MMI) approaches [21,22] and minimum classification error methods [23]. MMI, in particular, has recently been shown to be quite effective on conversational speech [22]. Later, connectionist systems [e.g. 24,25,26,28,29,30] were employed that used an inherently discriminative ANN acoustic model. While the connectionist systems have been able to match state-of-the-art performance, they did not achieve the great performance gains that were expected on large vocabulary tasks.

New approaches to discriminative modeling for speech recognition have centered around the powerful SVM paradigm described above. The interest in these models for speech is due to two important characteristics of the SVM model. First, SVMs are formulated as optimal generalization machines — overfitting of the data is explicitly avoided in the modeling. Contrast this to neural network approaches where overfitting is typically controlled using a cross-validation process that is wasteful of resources and whose performance is not quantifiable (though see the next chapter for examples of relevance determination methods by MacKay [47,48] which avoid this problem). This property of SVMs has translated to classification performance that has consistently exceeded neural networks and GMMs [25,53,64]. Second, the SVM (through the use of

Mercer kernels) has the ability to build a binary classifier in a high-dimensional space. Unlike other classifiers, neither the dimensionality nor the sparsity of the data in the transform space is a limiting factor for SVMs.

Initial applications of SVMs to speech came in the form of speaker verification systems [65]. Their success was limited, though, due primarily to lack of efficient training methods. Phone classification was the next problem to be tackled using SVMs [59,55]. These systems performed on par with state-of-the-art and their performance was far superior to neural network systems [25] on the same task. With the phone classification problem, the SVM systems were forced to address the first problem with applying SVMs to speech - nonuniform segment lengths. Their solution to this problem was to artificially impose a fixed vector length using a segmental modeling approach that will be described in detail below.

Steps toward using SVMs for word-level continuous speech recognition came in the form of isolated word recognition systems. Bazzi and Katabi [57] built a digit recognition system that employed the same techniques as the phone classification systems. Each digit was modeled with a single one-vs-all classifier. A decimation approach was used to solve the nonuniform segment problem which can be described by the following algorithm:

1. Compute a distance measure, $d_i = f(\boldsymbol{o}_i, \boldsymbol{o}_{i-1})$, for $0 \le i < N$.

2. Find i for which d_i is a minimum. Remove $\boldsymbol{o}_i$ and decrease N by 1.

3. Repeat 1 and 2 until N is the desired size.

Following the decimation stage, a PCA transform was computed to bring the decimated feature vector to its final size. Using a small training set, the SVM system was able to achieve a 5.1% error rate compared to 9.3% error for a GMM classifier. However, state-of-the-art on such tasks is a near-zero error rate.

To move from these simple applications of SVMs as static classifiers to an SVM solution for continuous speech requires addressing two primary issues. First, the dynamic nature of speech must be modeled. SVMs are inherently static classifiers while speech is a dynamically evolving process. The systems described above tried to avoid the problem of dynamics altogether by artificially imposing a fixed vector length. Hybrid connectionist systems address the dynamics of speech by embedding neural networks into an HMM structure [24,29]. The second problem to address is the need to insert SVMs into a probabilistic framework that is used to combine disparate knowledge sources. SVMs are, by definition, binary classifiers capable of giving an in-class/out-of-class judgement. This judgement is rendered by finding the distance from the hyperplane boundary. In general, only the sign of this distance provides useful information, but to apply SVMs in a probabilistic framework one has to map this distance measure to a probability measure (of course one could try to learn the probability function directly using SVM regression but then the power of the discriminative classification is lost).

3.3. Hybrid HMM/SVM System

Research into addressing these remaining issues has proceeded in two directions. First are the systems which use a Fisher kernel capable of handling variable length features [66,67,68] to solve the segmentation problem. While promising, this technique is still in the early stages and has only been applied to relatively simple tasks to date. A more mature method has been defined by Ganapathiraju [27] and colleagues [59,60,61,62,63] which follows a hybrid approach combining techniques from the connectionist systems [24,25,26,29] and segmental modeling systems [69,70]. It is the first to comprehensively address the problems associated with applying SVMs to continuous speech recognition (Chakrabartty, et al. [58] also proposed a hybrid system as well as a circuit design to implement the system in hardware. However, they have only demonstrated their system on a relatively trivial task so it is unclear if their approach holds promise).

Posterior Estimation

The first challenge faced in building the HMM/SVM system is the construction of a probabilistic model from the SVM discriminant function. The approach taken in [27,63] which is drawn from the work of Kwok [71] and Platt [72] is to build a functional mapping from the SVM distance function to a number on the range of [0,1] representing a probability function. If we let $f(\boldsymbol{o})$ be the SVM distance function and t be the class label where $t = \pm 1$, then we can write the posterior probability $P(t = 1|f)$ as

$$P(t = 1|f) = \frac{P(f|t = 1)P_1}{P(f|t = 1)P_1 + P(f|t = -1)P_{-1}}. \tag{40}$$

It remains, then, to define the form of the likelihood functions, $P(f|t = 1)$ and $P(f|t = -1)$, and the priors on the in-class and out-of-class data, P_1 and P_{-1}.

Taking the maximum entropy approach, the likelihood functions can be defined by Gaussian distributions as

$$P(f|t = 1) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(f-u_1)^2}{2\sigma_1^2}} \text{ and} \tag{41}$$

$$P(f|t = -1) = \frac{1}{\sqrt{2\pi\sigma_{-1}^2}} e^{-\frac{(f-u_{-1})^2}{2\sigma_{-1}^2}} . \tag{42}$$

Normalizing (40) by its numerator and combining exponential terms yields

$$p(t = 1|f) = \frac{1}{1 + \frac{P(f|t = -1)}{P(f|t = 1)} \frac{P_{-1}}{P_1}} = \frac{1}{1 + \frac{P_{-1}}{P_1} \frac{\frac{1}{\sqrt{2\pi\sigma_{-1}^2}} e^{-\frac{(f-u_{-1})^2}{2\sigma_{-1}^2}}}{\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(f-u_1)^2}{2\sigma_1^2}}}}, \tag{43}$$

which, after simplification gives the form

$$p(t = 1|f) = \frac{1}{1 + \frac{P_{-1}}{P_1} \frac{\sigma_1}{\sigma_{-1}} e^{-\frac{1}{2}\left[\left(\frac{f-u_1}{\sigma_1}\right)^2 - \left(\frac{f-u_{-1}}{\sigma_{-1}}\right)^2\right]}} . \tag{44}$$

Finally, if we assume that the variances of the discriminant function for in-class and out-of-class data is equal then we can expand the squared terms in the exponent to define the posterior probability in the form of a sigmoid function

$$p(t = 1|f) = \frac{1}{1 + \frac{P_{-1}}{P_1} e^{-\frac{1}{2\sigma^2}((u_1^2 - u_{-1}^2) + 2f(u_{-1} - u_1))}} = \frac{1}{1 + e^{(Af + B)}} . \tag{45}$$

Here, the parameters A and B are estimated using any suitable nonlinear optimization scheme to optimally map the discriminant function to the probability space. Note that the ratio of the priors has been incorporated into the exponential.

Recall that in the probabilistic formulation of speech presented in Chapter 2 the acoustic model was used to determine the likelihood function; i.e. the probability of the observed data given the assumed model, $P(\boldsymbol{O}|\boldsymbol{M})$. However, from (45), we have derived the posterior estimate of the probability of the model given the data, $P(\boldsymbol{M}|\boldsymbol{O})$. To generate the likelihood function, Bayes' rule needs to be applied. The failure to consider this is a potential weakness in the hybrid HMM/SVM system as it indicates a prior belief that each class is equally likely. Connectionist systems such as those in [24,29] estimate the class priors as part of the training routine. These systems have consistently shown significant degradations in performance when equal priors are applied.

Segmental modeling

A natural way to apply the new SVM acoustic model in an HMM/SVM hybrid system is to perform the classification directly at the frame level — replacing the Gaussian likelihood score with the SVM posterior described above. In fact, this is exactly the approach used by many hybrid connectionist systems. There are, however, two issues to consider in this regard.

1. **Feasibility for large corpora**: Large vocabulary training sets often contain on the order of 10-100 million frames of speech data. Even with the extremely efficient SVM optimizers available today, it is impractical to train the SVM on this quantity

of data. Connectionist systems face a similar problem in the iterative methods used for training [29]. However, parallel processing techniques [34] have been developed that allow them to use large data sets efficiently.

2. **Modeling long-term temporal structure**: Using frame-level data provides a very localized view of the speech signal. It removes the potential for modeling long-range dependencies in data such as cross-frame spectral correlations and for modeling long range “features” of the data such as phone duration [73,74,75]. A few approaches have been tried to alleviate this problem. HMM systems often include derivative terms in the feature stream to account for changes in the feature across frame boundaries [4]. Connectionist systems often concatenate a window of frames around the frame of interest to create a large feature vector [26]. The neural network is then allowed to learn the long-range correlations in the data. HMM/GMM systems could not use such an approach because the number of parameters grows linearly with the size of the feature vector. However, many systems are now using feature reduction techniques such as LDA and PCA to provide the HMM/GMM systems with a reduced-sized feature vector that still captures the most important long-range correlations [76].

To address both of these issues, the HMM/SVM system uses a segment-based approach akin to those in [69,70]. By modeling at a phone-segment level (i.e. each observation represents a sequence of frames that constitute a single spoken phone), the HMM/SVM system is able to greatly reduce the number of training vectors (by as much as

2-3 orders of magnitude) and is able to simultaneously model both the spectral and temporal structure of speech. With this approach, however, there remains the question of where to get the phone segments in the first place. The HMM/SVM system uses an HMM/GMM system to produce the segmentation information and then post-processes the data under the assumption that the segmentation is correct. Recent linguistic analysis seems to indicate that this is not a good assumption [77] for conversational speech.

Phone segments can have widely varying lengths (e.g. vowels tend to be longer and consonants tend to be shorter). However, with the conventional SVM model (in contrast to those which use Fisher kernels [66,67,68]) we require a fixed observation vector length. One way to mitigate this problem which follows the motivation of 3-state HMM phone models is to divide each segment into a fixed number of distinct subsections [78,79,80]. The frames in each subsection are then averaged and the averages are concatenated to yield a single fixed-length vector. This process is illustrated in Figure 2. While the percentage of the segment that is allocated to each subsection can be manipulated, the performance of the HMM/SVM system is insensitive to changes in the proportions [27,63].

System architecture

The hybrid HMM/SVM system is built using the rescoring paradigm shown in Figure 3. The HMM/GMM system generates a pruned hypothesis space as well as a segmentation (or set of segmentations). The SVM is used to rescore the hypothesis space given the segmentation(s). In [27,63] N-best lists are used to represent the pruned hypothesis space. These give a set of N unique hypotheses which are most highly predicted by the HMM/GMM system.

For experimental purposes, the segment information was generated in two ways. First, a single segmentation (1-best segmentation) was used to rescore all of the N-best hypotheses. This segmentation was derived from a forced-alignment of a word sequence to the speech data using the HMM/GMM system. For baseline testing, the word sequence is the 1-best hypothesis (hypothesis segmentation). This gives the best-guess segmentation of the HMM/GMM decoder. Note, however, that it may not be possible to align some of the N-best hypotheses to the 1-best segmentation, thus the 1-best segmentation acts to artificially constrain the search space for the SVM. For analysis, an oracle experiment can also be run which uses the reference transcription to find a single segmentation. An alternative segmentation method generates a separate segmentation for each entry in the N-best list (N-best segmentation) and rescores each one in turn. While more computationally expensive, this method provides a better comparison with an HMM/GMM system where the decoder is allowed to choose any segmentation for the hypotheses.

Experimental analysis

The HMM/SVM system was run on two different telephone-bandwidth tasks: the OGI Alphadigits [81] and the SWITCHBOARD (SWB) corpus [82]. The Alphadigits task is a small vocabulary (~40 words), open grammar (any word sequence is possible) task while the SWB task is a large vocabulary (modern lexicons contain as many as

100,000 words) open grammar task. The results of these experiments are shown in Table 1 [27].

The most interesting thing to note about these results is the surprisingly large gains made by the oracle system (experiment 4) for the Alphadigit system. A nearly 30% reduction in WER is achieved by the HMM/SVM system over the HMM/GMM system. This shows the potential power of the SVM classifier when it is presented with adequately rich information from the HMM system. Of course, reducing the n-best list error rate to 0% is usually not possible so we need to look for other ways to give the classifier a wider variety of hypotheses to choose from — e.g. integrating the SVM directly into the search. Another key point to note is the performance of the oracle system (experiment 3) using the reference segmentation. With this system, a 80% reduction in WER was achieved. While there is no fair comparison to an oracle HMM system given, this performance seems to establish that a good segmentation is the most important issue in applying SVMs in the hybrid framework. Making better use of the HMM framework for temporal modeling and to drive the SVM models is necessary to approach these levels of performance.

A follow-up experiment run as part of this dissertation also showed that the sigmoid posterior estimate applied by the hybrid HMM/SVM system does not significantly contribute to the performance of the hybrid system. In the experiment, the posterior estimate was replaced with a simple thresholding rule that mapped the SVM distance to the range of [0,1]. If the distance was greater than 0 (indicating a sample classified on the in-class side of the decision surface) then a probability of 1.0 was emitted. Otherwise a probability of 0.0 was emitted. In other words, the threshold probability mapping assumes perfect confidence in the classification provided by the SVM. With this modification, the total word error rate on the Alphadigits data was reduced by only 1.8% relative to the HMM/SVM system. If the sigmoid were an accurate model of the posterior, we would expect a more pronounced difference.

3.4. Summary

In this chapter, we have seen how the SVMs use a structural risk minimization argument to define an *optimal* decision surface which automatically rejects overfitting. In this way, the SVM combines the problems of prediction and decision-making. The theory of Mercer kernels are incorporated into the SVM framework to provide for extremely flexible and highly nonlinear decision surfaces. Further, the chapter has discussed the use of SVMs as classifiers for speech data. The first credible attempt at this is in the form of a hybrid HMM/SVM system. This system uses segmental modeling and posterior estimation techniques to address the issues related to interfacing SVMs to the HMM framework.

In the next chapter we will discuss the relevance vector machine (RVM) which is the object of this dissertation. RVMs use a mathematical structure that is similar to the SVM, but the RVM follows a more conventional motivation. RVMs seek to determine the posterior likelihood of a class assignment given the data, thus allowing for an external decision process. In this way, the RVM can take into account asymmetric misclassification costs, and varying class prior probabilities. Overfitting is avoided through the application of MacKay’s ARD principle [47,48]. While the generalization capability of the RVM is comparable to that of the SVM, the RVM offers a few very important advantages which will be explored in this dissertation.

REFERENCES

[1] J.R. Deller, J.G. Proakis and J.H.L. Hansen, *Discrete-Time Processing of Speech Signals*, Macmillan Publishing, New York, USA, 1993.

[2] L.R. Rabiner and B.H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1993.

[3] R. Cole, J. Mariani, H. Uszkoreit, G. B. Varile, A. Zaenen, A. Zampolli, and V. Zue, eds., *Survey of the State of the Art in Human Language Technology,* Chapter 9, Cambridge University Press, Cambridge, Massachusetts, USA, March 1998.

[4] J. Picone, "Signal Modeling Techniques in Speech Recognition," *IEEE Proceedings*, vol. 81, no. 9, pp. 1215-1247, September 1993.

[5] M. J. Hunt, "Spectral Signal Processing for ASR", *Proceedings of the 1999 Workshop on Automatic Speech Recognition and Understanding,* Keystone, Colorado, USA, December 1999.

[6] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-28, no. 4, pp. 357-366, August 1980.

[7] H. Hermansky, "Perceptual Linear Prediction (PLP) Analysis for Speech," *Journal of the Acoustical Society of America*, vol. 87, pp. 1738-1752, 1990.

[8] J. Bilmes, *Natural Statistical Models for Automatic Speech Recognition,* Ph.D. Dissertation, University of California, Berkeley, Berkeley, California, USA, 1999.

[9] L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Further results on the recognition of a continuously read natural corpus," *Proceedins of the 1980 IEEE Conference on Acoustics, Speech, and Signal Processing,* pp. 872-875, Denver, Colorado, USA, 1980.

[10] S. Greenberg, D. Ellis, and J. Hollenback, "Insights into spoken language gleaned from phonetic transcription of the Switchboard corpus," *Proceedins of the 1996 Intrenational Conference for Spoken Language Processing*, pp. 24-27, Philadelphia, Pennsylvania, USA, 1996.

[11] K. F. Lee, “Context-dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, pp. 599-609, April 1990.

[12] T. Hain, P. C. Woodland, G. Evermann, D. Povey, "CU-HTK March 2000 Hub 5E transcription system," *Proceedings of the NIST Speech Transcription Workshop*, College Park, Maryland, USA, March 2000.

[13] F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, Cambridge, Massachusetts, USA, 1997.

[14] F. Jelinek, “Up From Trigrams! The Struggle for Improved Language Models,” *Proceedings of the 2nd European Conference on Speech Communication and Technology (Eurospeech)*, pp. 1037-1040, Genova, Italy, September 1991.

[15] J. Picone, "Continuous Speech Recognition Using Hidden Markov Models", *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 7, no. 3, pp. 26-41, July 1990.

[16] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-285, February 1989.

[17] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models," *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 3, no. 1, pp. 4-16, January 1986.

[18] N. Deshmukh, A. Ganapathiraju and J. Picone, "Hierarchical Search for Large Vocabulary Conversational Speech Recognition," *IEEE Signal Processing Magazine*, vol. 16, no. 5, pp. 84-107, September 1999.

[19] L. E. Baum, T. Petrie, G. Soules, N. Weiss., “A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains,” *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164-171, 1970.

[20] A. P. Dempster, N. M. Laird and D. B. Rubin, “Maximum Likelihood Estimation from Incomplete Data,” *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1-38, 1977.

[21] 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," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 49-52, Tokyo, Japan, October 1986.

[22] P. Woodland and D. Povey, “Very Large Scale MMIE Training for Conversational Telephone Speech Recognition,” *Proceedings of the 2000 Speech Transcription Workshop,* University of Maryland, MD, USA, May 2000.

[23] E. McDermott, *Discriminative Training for Speech Recognition,* Ph.D. Dissertation, Waseda University, Japan, 1997.

[24] S. Renals, *Speech and Neural Network Dynamics,* Ph. D. dissertation, University of Edinburgh, UK, 1990.

[25] A. J. Robinson, *Dynamic Error Propagation Networks,* Ph.D. dissertation, Cambridge University, UK, February 1989.

[26] J. Tebelskis, S*peech Recognition using Neural Networks,* Ph. D. dissertation, Carnegie Mellon University, Pittsburg, USA, 1995.

[27] A. Ganapathiraju, *Support Vector Machines for Speech Recognition,* Ph.D. Dissertation, Mississippi State University, Mississippi State, Mississippi, USA, 2002.

[28] M.D. Richard and R.P. Lippmann, “Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities”, *Neural Computation*, vol. 3, no. 4, pp. 461-483, 1991.

[29] H.A. Bourlard and N. Morgan, *Connectionist Speech Recognition — A Hybrid Approach,* Kluwer Academic Publishers, Boston, USA, 1994.

[30] G.D. Cook and A.J. Robinson, “The 1997 ABBOT System for the Transcription of Broadcast News,” *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop,* Lansdowne, VA, USA, 1998.

[31] A. Robinson, “An application of recurrent nets to phone probability estimation,” *IEEE Transactions on Neural Networks,* vol. 5, no. 2, pp. 298-305, 1994.

[32] A. Robinson, et. al., “The Use of Recurrent Neural Networks in Continuous Speech Recognition,” in *Automatic Speech and Speaker Recognition -- Advanced Topics*, chapter 19, Kluwer Academic Publishers, 1996.

[33] J.S. Bridle, Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationship to Statistical Pattern Recognition, *Neuro-Computing: algorithms, architectures and applications*, Springer-Verlag, 1989.

[34] N. Morgan, J. Beck, P. Kohn, and J. Bilmes, "Neurocomputing on the RAP," in K. W. Przytula and V. K. Prasanna, eds., *Digital Parallel Implementations of Neural Networks*. Prentice Hall, 1992.

[35] Y. Yan, M. Fanty and R. Cole, "Speech Recognition Using Neural Networks with Forward-Backward Probability Generated Targets," *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*, Munich, Germany, April 1997.

[36] H. Bourlard, Y. Konig and N. Morgan, "REMAP: Recursive Estimation and Maximization of A Posteriori Probabilities in Connectionist Speech Recognition", *Proceedings of EUROSPEECH '95*, Madrid, Spain, September 1995.

[37] V.N. Vapnik, *Statistical Learning Theory,* John Wiley, New York, NY, USA, 1998.

[38] C. Cortes, V. Vapnik. Support Vector Networks, *Machine Learning*, vol. 20, pp. 293-297, 1995.

[39] C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, http://svm.research.bell-labs.com/SVMdoc.html, AT&T Bell Labs, November 1999.

[40] E. Osuna, R. Freund, and F. Girosi, “Support Vector Machines: Training and Applications,” *MIT AI Memo 1602*, March 1997.

[41] B. Schölkopf, *Support Vector Learning*, Ph.D. dissertation, R. Oldenbourg Verlag Publications, Munich, Germany, 1997.

[42] B. Schölkopf, C. Burges and A. Smola, *Advances in Kernel Methods: Support Vector Machines*, MIT Press, Cambridge, MA, USA, December 1998.

[43] M.A. Hearst, et. al., “Trends and Controversies - Support Vector Machines”, *IEEE Intelligent Systems*, vol. 13, pp. 18-28, 1998.

[44] M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *Journal of Machine Learning*, vol. 1, pp. 211-244, June 2001.

[45] M. Tipping, "The Relevance Vector Machine," in *Advances in Neural Information Processing Systems 12*, S. Solla, T. Leen and K.-R. Muller, eds., pp. 652-658, MIT Press, 2000.

[46] C. Bishop and M. Tipping, "Variational Relevance Vector Machines," *Proceedings of the 16th Conference in Uncertainty in Artificial Intellignece,* pp. 46-53, Morgan Kaufmann Publishers, 2000.

[47] D. J. C. MacKay, *Bayesian Methods for Adaptive Models*, Ph. D. Thesis, California Institute of Technology, Pasadena, California, USA, 1991.

[48] D. J. C. MacKay, “Probable networks and plausible predictions --- a review of practical Bayesian methods for supervised neural networks,” *Network: Computation in Neural Systems*, 6, pp. 469-505, 1995.

[49] V.N. Vapnik, *Statistical Learning Theory*, John Wiley, New York, NY, USA, 1998.

[50] P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization*, Academic Press, New York, 1981.

[51] B. Boser, I. Guyon and V. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144-152, Pittsburgh, Pennsylvania, USA, 1992.

[52] O. L. Mangasarian, W. Nick Street and W. H. Wolberg: ‘‘Breast cancer diagnosis and prognosis via linear programming", *Operations Research*, 43(4), pp. 570-577, July-August 1995.

[53] Y. Le Cun, L.D. Jackel, L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, U.A. Muller, E. Sackinger, P. Simard, and V. Vapnik, “Learning algorithms for classification: A comparison on handwritten digit recognition,” in *Neural Networks: The Statistical Mechanics Perspective,* J.H. Kwon and S. Cho, eds., pp. 261--276, World Scientific, 1995.

[54] T. Joachims, “Text Categorization with Supprt Vector Machines: Learning with Many Relevant Features,” *Technical Report 23, LS VIII, University of Dortmund*, Germany, 1997.

[55] P. Clarkson and P. Moreno, "On the Use of Support Vector Machines for Phonetic Classification," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Phoenix, Arizona, USA, 1999.,

[56] K.-R. Muller, A. J. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen and V. Vapnik, "Predicting Time Series with Support Vector Machines", in *Proceedings of the International Conference on Articial Neural Networks*, W. Gerstner, A. Germond, M. Hasler and J.-D. Nicoud, eds., pp. 999-1004, Springer, 1997.

[57] I. Bazzi and D. Katabi, “Using Support Vector Machines for Spoken Digit Recognition,” *Proceedings of the 6th International Conference on Spoken Language Processing*, Beijing, China, October 2000.

[58] S. Chakrabartty, G. Singh and G. Cauwenberghs, "Hybrid support vector machine / hidden markov model approach for continuous speech recognition," *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems*, Volume: 2, pp. 828-831, 2000.

[59] A. Ganapathiraju, J. Hamaker and J. Picone, “Support Vector Machines for Speech Recognition,” *Proceedings of the International Conference on Spoken Language Processing*, Sydney, Australia, November 1998.

[60] A. Ganapathiraju, J. Hamaker and J. Picone, “A Hybrid ASR System Using Support Vector Machines,” submitted to the *International Conference of Spoken Language Processing*, Beijing, China, October, 2000.

[61] A. Ganapathiraju and J. Picone, "Hybrid SVM/HMM architectures for speech recognition,” submitted to *Neural Information Processing Systems - 2000*, Denver, Colorado, USA, November 2000.

[62] A. Ganapathiraju, J. Hamaker and J. Picone, “Hybrid HMM/SVM Architectures for Speech Recognition,” *Proceedings of the Department of Defense Hub 5 Workshop*, College Park, Maryland, USA, May 2000.

[63] A. Ganapathiraju, J. Hamaker and J. Picone, “Continuous Speech Recognition Using Support Vector Machines,” submitted to *Computer, Speech and Language*, November 2001.

[64] T. Joachims, *SVMLight: Support Vector Machine*, http://www-ai.informatik.uni-dortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light.eng.html, University of Dortmund, November 1999.

[65] M. Schmidt, H. Gish, “Speaker Identification Via Support Vector Classifiers,” *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 105-108, Atlanta, GA, USA, May 1996

[66] T. Jaakkola and D. Haussler, “Exploiting generative models in discriminative classifiers,” *Advances in Neural Information Processing Systems*, 11, 1998.

[67] N. Smith and M.J.F. Gales, “Speech Recognition using SVMs” *Proceedings of Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 2001.

[68] N. Smith and M. Gales. “Using SVMs to classify variable length speech patterns,” *Technical Report CUED/F-INFENG/TR.412*, Cambridge University Eng. Dept., June 2001.

[69] Ostendorf, M., Digalakis, V. and Kimball, O (1996). From HMM’s to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 360-378.

[70] Ostendorf, M. and Roukos, S. (1989). A Stochastic Segment Model for Phoneme-Based Continuous Speech Recognition. *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 37, pp. 1857-1867.

[71] J. Kwok, "Moderating the Outputs of Support Vector Machine Classifiers," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, 1999.

[72] J. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," in *Advances in Large Margin Classifiers*, MIT Press, Cambridge, MA, USA, 1999.

[73] M.J. Russell and R.K., Moore, “Explicit Modeling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition,” *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 5-8, Tampa, USA, 1985.

[74] W. Holmes, *Modelling Segmental Variability for Automatic Speech Recognition*, Ph. D. dissertation, University of London, UK, 1997.

[75] M.J.Russell and W.J. Holmes, “Linear Trajectory Segmental Models,” *IEEE Signal Processing Letters*, vol. 4, no. 3, pp. 72-74, 1997.

[76] G. Saon, M. Padmanabhan, R. Gopinath and S. Chen, "Maximum Likelihood Discriminant Feature Spaces," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, 2000.

[77] S. Greenberg and S. Chang, "Linguistic Dissection of Switchboard-Corpus Automatic Speech Recognition Systems," *Proceedings of the ISCA Workshop on Automatic Speech Recognition: Challenges for the New Millennium*, Paris, France 2000.

[78] J. Chang and J. Glass, "Segmentation and Modeling in Segment-based Recognition," *Proceedings of Eurospeech*, pp. 1199-1202, Rhodes, Greece, 1997.

[79] N. Ström, L. Hetherington, T. J. Hazen, E. Sandness and J. Glass, "Acoustic Modeling Improvements in a Segment-Based Speech Recognizer," *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*, Keystone, CO, USA, 1999.

[80] A. K. Halberstadt, *Heterogeneous Acoustic Measurements and Multiple Classifiers for Speech Recognition,* Ph. D. Thesis, MIT Department of Electrical Engineering and Computer Science, Cambridge, Massachusetts, USA, 1998.

[81] R. Cole, "Alphadigit Corpus v1.0". *http://www.cse.ogi.edu/CSLU/corpora/alphadigit*, Center for Spoken Language Understanding, Oregon Graduate Institute, USA, 1998.

[82] J. Godfrey, E. Holliman and J. McDaniel, "SWITCHBOARD: Telephone Speech Corpus for Research and Development," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 517-520, San Francisco, California, USA, 1992.

[83] C. J. C. Burges and B. Schölkopf, "Improving the Accuracy and Speed of Support Vector Machines," *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan and T. Petsche, eds., vol. 9, The MIT Press, 1997.

[84] V. Roth, "Sparse Kernel Regressors," *Proceedings of ICANN,* pp. 339-346, 2001.

[85] S. Chen, S. R. Gunn and C. J. Harris, "The Relevance Vector Machine Technique for Channel Equalization Applications," *IEEE Transactions on Neural Networks,* vol. 12, pp. 1529-1532, 2001.

[86] J. B. Gao, S. R. Gunn, C. J. Harris and M. Brown, "Regression with Input-dependent Noise: a Relevance Vector Machine Treatment," *IEEE Transactions on Neural Networks,* March 2001.

[87] E. T. Jaynes, "Bayesian Methods: General Background," *Maximum Entropy and Bayesian Methods in Applied Statistics,* J. H. Justice, ed., pp. 1-25, Cambridge University Press, Cambridge, UK, 1986.

[88] H. Jeffreys, *Theory of Probability,* Oxford University Press, 1939.

[89] T. J. Loredo, "From Laplace to Supernova SN 1987A: Bayesian Inference in Astophysics," *Maximum Entropy and Bayesian Methods,* P. Fougere, ed, Kluwer Publishing, 1989.

[90] S. F. Gull, "Bayesian Inductive Inference and Maximum Entropy," *Maximum Entropy and Bayesian Methods in Science and Engineering, vol. 1, Foundations,* G. J. Erickson and C. R. Smith, eds., Kluwer Publishing, 1988.

[91] J. Rissanen, "Modeling by Shortest Data Description," *Automatica,* 14, pp. 465-471, 1978.

[92] G. Schwartz, "Estimating the Dimension of a Model," *Annals of Statistics,* vol. 6, no. 2, pp. 461-464, 1978.

[93] A. Faul and M. Tipping, "Analysis of Sparse Bayesian Learning," *Proceedings of the Conference on Neural Information Processing Systems*, preprint, 2001.

[94] N. Deshmukh, et. al., “A Public Domain Speech-to-Text System,” *Proceedings of. Eurospeech,* vol. 5, Budapest, Hungary, September 1999.

[95] D. Deterding, M. Niranjan and A. J. Robinson, "Vowel Recognition (Deterding data)," Available at *http://www.ics.uci.edu/pub/machine-learning-databases/ undocumented/connectionist-bench/vowel, 2000.*

[96] R. G. Leonard, "A Database for Speaker Independent Digit Recognition," *Proceedings of the International Conference for Acoustics, Speech and Signal Processing*, vol. 3, pp. 42-45, San Diego, California, USA, 1984.

[97] J. Hamaker, A. Ganapathiraju, J. Picone and J. Godfrey, “Advances in Alphadigit Recognition Using Syllables,” *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 421-424, Seattle, Washington, USA, May 1998.

[98] A. Ganapathiraju et. al., “WS97 Syllable Team Final Report,” P*roceedings of the 1997 LVCSR Summer Research Workshop*, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, Maryland, USA, December 1997.

[99] J. Tenenbaum and W.T. Freeman, "Separating Style and Content," *Advances in Neural Information Processing Systems 9*, MIT Press, Cambridge, Massachusetts, USA, 1997.