SPARSE BAYESIAN METHODS FOR CONTINUOUS SPEECH RECOGNITION

By

Jonathan E. Hamaker

A Dissertation Proposal
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

February 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)	James C. Harden Graduate Coordinator of Computer Engineering in the Department of Electrical and Computer Engineering
G. Marshall Molen 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: February 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: 76

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 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 (RVMs), 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

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

STATISTICAL APPROACH TO 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. However, for our limited purpose of computer speech recognition, we will consider only the problem of converting an acoustic signal (i.e. the speaker's voice) into a stream 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* (see [3] for examples of multimodal recognition technology).

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 thesis.

1.1. 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 is at the core of most acoustic front-ends for speech recognition and is used in this work [6]. This front-end is depicted in Figure 1. While this front-end is not the only possibility (see, for example [7]), it has been widely used in speech recognition applications.

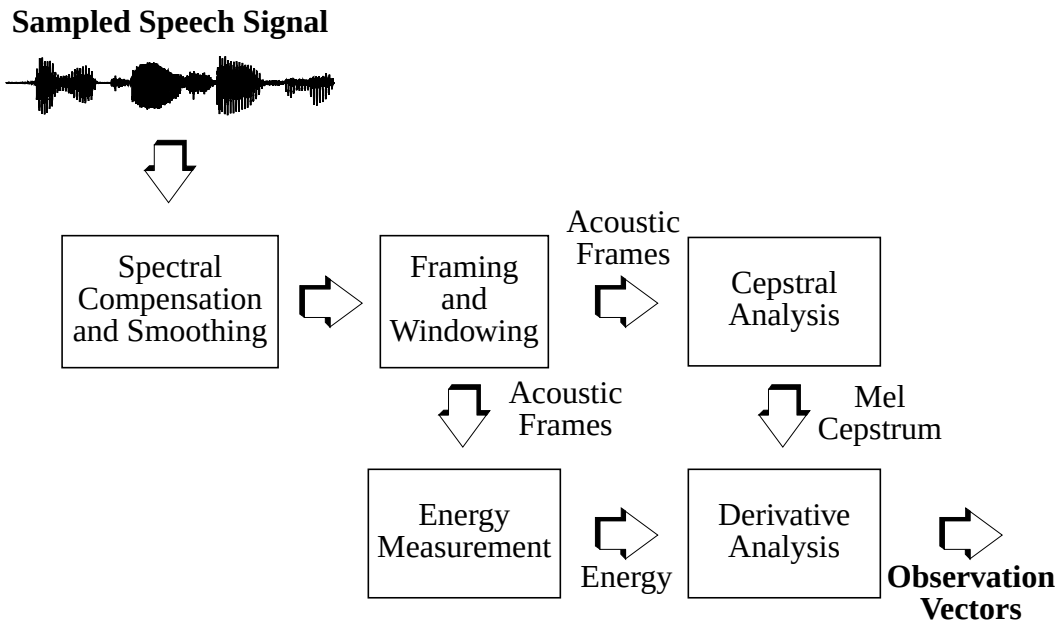

Figure 1. Typical Mel-Cepstral acoustic front-end.

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 2. 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 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].

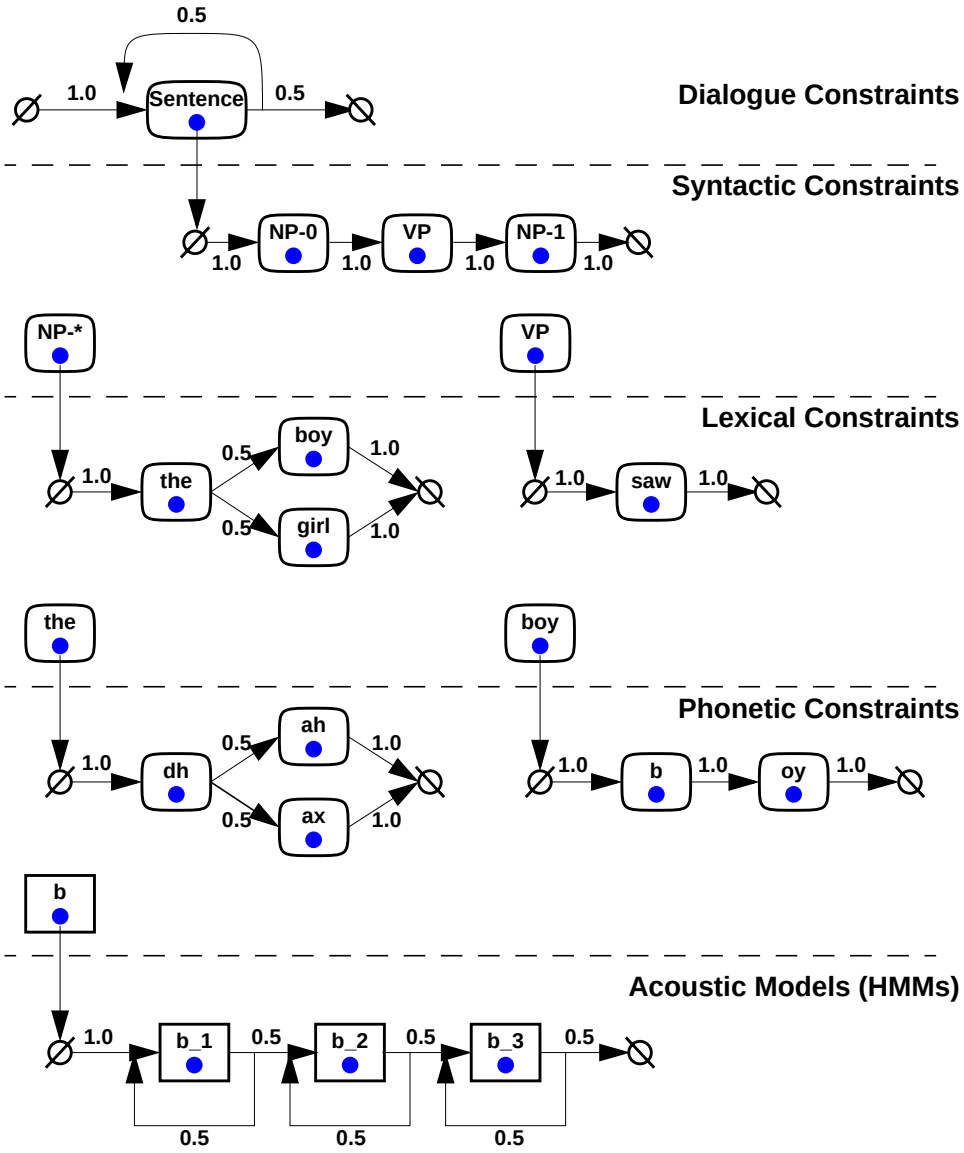

Figure 2. Speech is roughly modeled as a hierarchical constraint system. At each level of the hierarchy, a different knowledge source is applied. The job of a speech recognition system is to combine these knowledge sources in an optimal manner. Often the lowest level in the hierarchy is modeled by hidden Markov models and is responsible for the acoustic match (i.e. modeling the observations sequences generated by the acoustic front-end).

To model these coarticulation effect, 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, $\boldsymbol{O}$:

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

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 (1) as

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

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 (2) because the observation sequence, $\boldsymbol{O}$, is constant during the maximization. This yields

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

The terms in (3) 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 acoustic model [15,16,17,18]. The HMM (an example of which is shown in Figure 3) 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

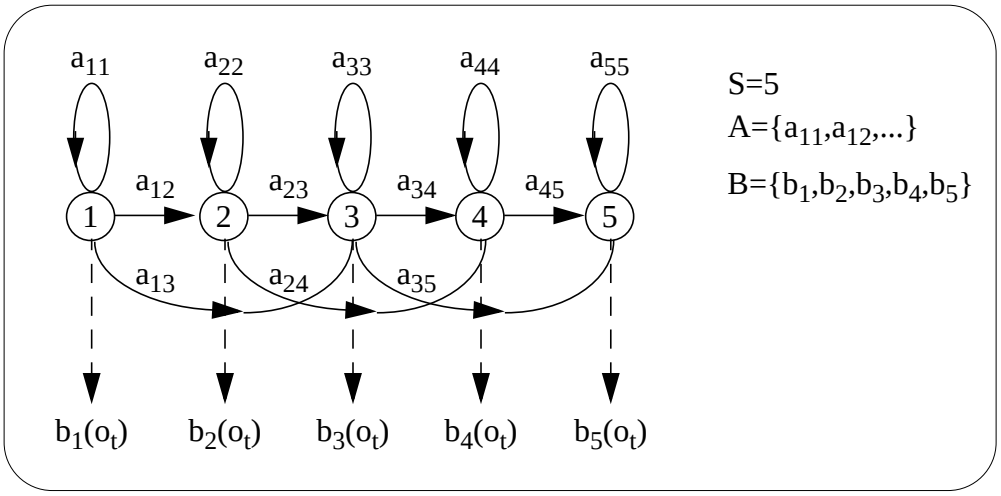

Figure 3. A simple HMM featuring a five state topology with skip transitions. Each state has a stochastic emission distribution.

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

$$b_j(\boldsymbol{o}_t) = \sum_{i=1}^{K} C_{ij} N(\boldsymbol{o}_t | \mu_{ij}, \Sigma_{ij}), \quad \sum C_{ij} = 1, \text{ where} \tag{4}$$

$$N(\boldsymbol{o}_t | \mu_{ij}, \Sigma_{ij}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_{ij}|}} \exp\left(-\frac{1}{2}(\boldsymbol{o}_t - \mu_{ij})^{\mathrm{T}} \Sigma_{ij}^{-1} (\boldsymbol{o}_t - \mu_{ij})\right). \tag{5}$$

In (4) and (5), 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.

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 2 shows how the HMM fits into the hierarchical model described previously.

1.2. Closed-loop, Supervised Parameter Estimation

The answer to the third question above (how do we train these models?) comes from taking an account of the tunable parameters in the hierarchical HMM system described previously. These are the language model probabilities, pronunciation model probabilities and the HMM state transitions ($\{a_{ij}\}$), mixture weights (C_{ij}), means (μ_{ij}) and covariances (Σ_{ij}). We ignore the first two of these in this dissertation and point the reader to [13] regarding training procedures for language models and pronunciation models. Instead, we concentrate on the HMM parameters which are directly related to the acoustic model. Typically, this approach involves finding the HMM parameter set that maximizes the likelihood of the data given the model — the maximum likelihood (ML) approach.

As with most machine learning tasks, training acoustic models begins with some 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 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”, but we do not know in which time interval each word occurred. Therefore, we can not immediately determine which acoustic observation should be used to train the individual emission probabilities. This is known as the *segmentation problem*.

A simple two-step approach can be taken to alleviate the segmentation problem. First, hypothesize the sequence of HMM states which were most likely to have generated the sequence of acoustic observations given the current parameter set; then 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 [2].

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 or that the iterative procedure will converge. Baum and colleagues [19] addressed these problems by defining a soft-decision training paradigm which is a special case of the expectation-maximization (EM) algorithm [20]. The EM formulation has the desirable property of guaranteed convergence to a local maximum.

Baum [19] defined an EM-type auxiliary function as

$$Q(\lambda, \lambda') = \sum_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda') \log P(\boldsymbol{O}, \boldsymbol{q}|\lambda) \tag{6}$$

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). Maximizing $Q(\lambda, \lambda')$ with respect to λ insures that

$$Q(\lambda, \lambda') \geq Q(\lambda', \lambda') \tag{7}$$

which implies that

$$P(\boldsymbol{O}|\lambda) \geq P(\boldsymbol{O}|\lambda'). \tag{8}$$

Thus, maximizing the auxiliary function monotonically increases the likelihood of the data given the model [19,20,2] until a critical point is reached. Note that the sum over all $\boldsymbol{q}$ in (6) implies a soft decision as to which is the true alignment of states. Contrast this to the Viterbi training algorithm where a single alignment was assumed to be the true alignment.

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:

$$\alpha_j(t) = P(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, q_t = j \,|\, \lambda'). \tag{9}$$

We can inductively define $\alpha_j(t)$ as a function of $\alpha_1(t-1), \ldots, \alpha_S(t-1)$. 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:

$$\beta_j(t) = P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T, q_t = j|\lambda'). \tag{10}$$

It can be defined inductively as a function of $\beta_1(t+1), \ldots, \beta_S(t+1)$. These inductive representations provide an extremely efficient method for estimating $\alpha_j(t)$ and $\beta_j(t)$. Note that, in Viterbi training, the $\alpha_j(t)$ and $\beta_j(t)$ were all unity for the states in the assumed alignment and zero for all other state alignments.

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

$$P(\boldsymbol{O}, q_t = j|\lambda') = \alpha_j(t)\beta_j(t). \tag{11}$$

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

$$P(\boldsymbol{O}|\lambda') = \sum_{j=1}^{S} \alpha_j(t)\beta_j(t). \tag{12}$$

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

$$P(\boldsymbol{O}, q_{t-1}= i, q_t= j|\lambda') = \alpha_i(t-1)a_{ij}b_j(\boldsymbol{o}_t)\beta_j(t). \tag{13}$$

The above three probabilistic equations amount to the expectation step of the EM algorithm. With (11), (12) and (13) in place, we can substitute them into the auxiliary function, (6), 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 are fully derived in [2,13].

While the combination of HMMs and Gaussian mixture models (HMM/GMM) has been extremely successful, there are some key assumptions made that are not appropriate for speech modeling.

1. The assumption of conditional independence (i.e. that all probabilities in the system are conditioned only on the current state) is clearly false. The probability of an acoustic observation given a particular state is highly correlated with both past and future observations. Most HMM systems account for this by including derivative features in the observation vector [5], thus breaking the model of conditional independence. Ideally, one would want to condition the distribution itself on the acoustic context, but that is impractical in conventional systems.

2. The HMM/GMM system makes assumptions about the parametric form of the underlying distribution which may lead to a poor match to the true underlying distribution.

3. Maximum likelihood approaches do not improve the discriminative abilities of the model. In other words, the ML approach maximizes the probability of the correct model while implicitly ignoring the probability of the incorrect model. 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. Methods such as maximum mutual information [21,22] and minimum classification error [23] have been developed to incorporate discriminative training

directly into the standard HMM/GMM framework. However, their success has been limited due primarily to their considerable computational costs [22].

1.3. Connectionist Speech Recognition

The weaknesses of the HMM/GMM system have led researchers to seek models which mitigate some or all of them [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 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. The outputs of a 1-of-N classifier trained under the mean-squared error criteria are known to approximate the posterior class 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 (2). 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, $P(\boldsymbol{W}|\boldsymbol{O})$. Thus, the posterior class probabilities need to be converted to likelihoods using Bayes' rule

$$\frac{P(c|\boldsymbol{o})}{P(c)} = \frac{P(\boldsymbol{o}|c)}{P(\boldsymbol{o})}. \tag{14}$$

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

Using (14), 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 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:

$$\gamma_j(t) = P(q_t = j|\boldsymbol{O}, \lambda) = \frac{\alpha_j(t)\beta_j(t)}{\sum_{k \in \boldsymbol{S}} \alpha_k(T)}. \tag{15}$$

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.

1.4. Summary

This chapter has reviewed the most common acoustic modeling framework for speech recognition systems — HMMs with GMM emission probability distributions. 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 2

SUPPORT VECTOR MACHINES FOR SPEECH RECOGNITION

Given a training corpus, $\boldsymbol{O} = \{(\boldsymbol{o}_1, y_1), (\boldsymbol{o}_2, y_2), \ldots\}$ where $\boldsymbol{o}_i$ is the i'th input observation and y_i is the corresponding target (e.g. class assignment or class probability), the goal of a learning machine is to learn the mapping $y = f(\boldsymbol{o})$ 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

$$y(\boldsymbol{o};\boldsymbol{w}) = w_o + \sum_{i=1}^{M} w_i \phi_i(\boldsymbol{o}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{o}). \tag{16}$$

A special form of (16) 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 two such models: the Support Vector Machine (SVM) [37,38,39,40,41,42,43] and Relevance Vector Machine (RVM) [44,45,46].

2.1. 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

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \; R(\alpha) = \underset{\alpha}{\operatorname{argmin}} \int Q(\boldsymbol{o}, y, \alpha) dP(\boldsymbol{o}, y) \tag{17}$$

where $Q(\boldsymbol{o}, y, \alpha)$ 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 $P(\boldsymbol{o}, y)$ is the joint distribution of the observations and targets. Finding a minimum for (17) is usually impossible because $P(\boldsymbol{o}, y)$ can not be found *a priori*. Thus, we look for a simplification of (17) 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,

$$R_{emp}(\alpha) = \frac{1}{l} \sum_{i = 1 \ldots l} Q(\boldsymbol{o}_i, y_i, \alpha). \tag{18}$$

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 $P(\boldsymbol{o}, y)$. Finding the α which minimizes (18) gives the *empirical risk minimization (ERM)* solution and is one of the most commonly used optimization procedures in machine learning. 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,

$$R(\alpha) \leq R_{emp}(\alpha) + f(h) \tag{19}$$

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

$$\frac{\varepsilon(l)}{2}\left(1 + \sqrt{1 + \frac{4R_{emp}(\alpha_l)}{\varepsilon(l)}}\right) \tag{20}$$

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,

$$\varepsilon(l) = 4\frac{h(\log(2l/h + 1)) - \log\eta/4}{l}. \tag{21}$$

From (21), 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 (21) 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 4 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 $\{\boldsymbol{o}_i, y_i\}, i = 1, \ldots l$ be the training set of length l where $y_i = \pm 1$ 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

$$\boldsymbol{w} \cdot \boldsymbol{o} + b = 0 \tag{22}$$

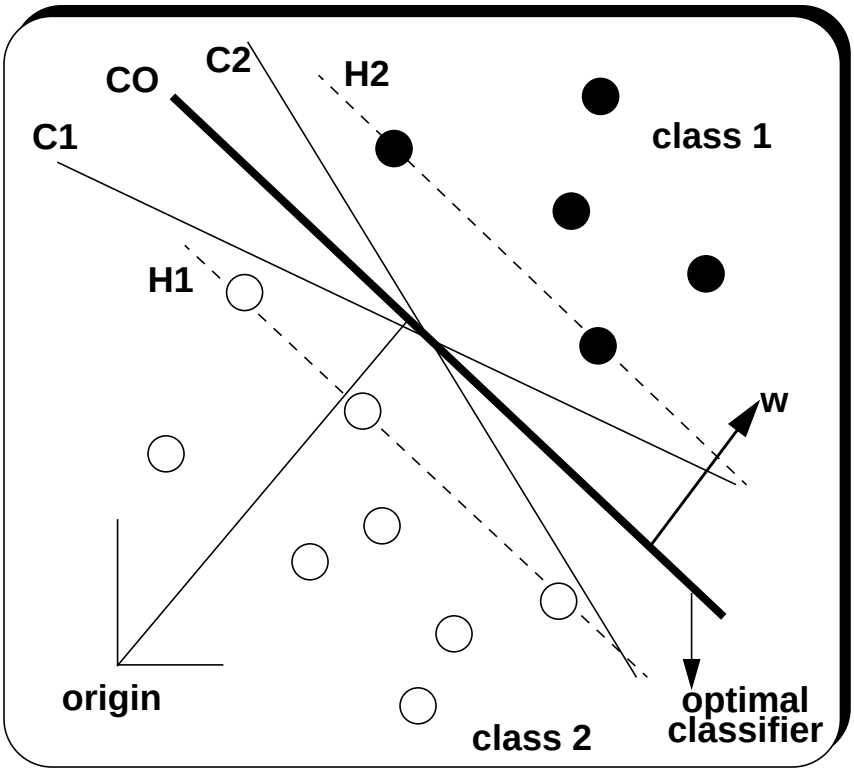

Figure 4. Difference between empirical risk minimization and structural risk minimization for a simple example involving a hyperplane classifier. Each hyperplane ($C0$, $C1$ and $C2$) achieves perfect classification and, hence, zero empirical risk. However, $C0$ is the optimal hyperplane because it maximizes the margin — the distance between the hyperplanes $H1$ and $H2$. Maximizing the margin indirectly results in better generalization.

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

$$\boldsymbol{o}_i \cdot \boldsymbol{w} + b \geq +1 \qquad \text{for } y_i = +1 \tag{23}$$

$$\boldsymbol{o}_i \cdot \boldsymbol{w} + b \leq -1 \qquad \text{for } y_i = -1\,. \tag{24}$$

These can be combined into a single set of inequalities,

$$y_i(\boldsymbol{o}_i \cdot \boldsymbol{w} + b) - 1 \geq 0 \qquad \forall i\,. \tag{25}$$

Vectors for which the equality condition in (25) holds are known as *support vectors*.

We can require that all points satisfying the equality condition in (23) lie on the hyperplane $H_1: \boldsymbol{o}_i \cdot \boldsymbol{w} + b = 1$ with normal vector $\boldsymbol{w}$ and distance from the origin of $|1 - b|/\|\boldsymbol{w}\|$. Similarly, all points satisfying the equality condition in (24) 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 (25). 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

$$L_P \equiv \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \alpha_i y_i (\boldsymbol{o}_i \cdot \boldsymbol{w} + b) + \sum_{i=1}^{N} \alpha_i \tag{26}$$

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

$$\boldsymbol{w} = \sum_j \alpha_j y_j \boldsymbol{o}_j, \text{ and} \tag{27}$$

$$\sum_i \alpha_i y_i = 0. \tag{28}$$

Equations (22) and (27) imply that the decision function can be defined as,

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{o}_i \cdot \boldsymbol{o} + b \tag{29}$$

where the sign of f can be used to classify examples as either in-class or out-of-class. This equation defines the SVM classifier. Notice the correspondence between (29) and (16): b corresponds to w_0, α_i to w_i, M to N, and $\phi_i(\boldsymbol{o}) = y_i \boldsymbol{o}_i \cdot \boldsymbol{o}$. 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 may be nonlinearly separable, or completely inseparable. In either case, we must find a method which maximizes the margin 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 of 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, (23) and (24), become

$$\boldsymbol{o}_i \cdot \boldsymbol{w} + b \geq +1 - \xi_i \qquad \text{for } y_i = +1, \tag{30}$$

$$\boldsymbol{o}_i \cdot \boldsymbol{w} + b \leq -1 + \xi_i \qquad \text{for } y_i = -1, \text{ and} \tag{31}$$

$$\xi_i \geq 0 \qquad \forall i, \tag{32}$$

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* trick [51]. Notice that, in the optimization problem formulated in (26), 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

$$K(\boldsymbol{o}_i, \boldsymbol{o}_j) = \phi(\boldsymbol{o}_i) \cdot \phi(\boldsymbol{o}_j). \tag{33}$$

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, (29), then becomes

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x_i}) + b. \tag{34}$$

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

$$K_{poly}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y} + 1)^d \tag{35}$$

$$K_{RBF}(\boldsymbol{x}, \boldsymbol{y}) = \exp\{-\Upsilon|\boldsymbol{x} - \boldsymbol{y}|^2\}. \tag{36}$$

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 1. 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].

2.2. Support Vector Methods

Initial attempts to add discriminative information to speech recognition used discriminative training of HMM/GMM systems using 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 the 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).

2.3. 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 y be the class label where $y = \pm 1$, then we can write the posterior probability $P(y = 1 | f)$ as

$$P(y = 1 | f) = \frac{P(f | y = 1)P_1}{P(f | y = 1)P_1 + P(f | y = -1)P_{-1}}. \quad (37)$$

It remains, then, to define the form of the likelihood functions, $P(f | y = 1)$ and $P(f | y = -1)$, 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

$$P(f | y = 1) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(f-u_1)^2}{2\sigma_1^2}} \text{ and} \quad (38)$$

$$P(f | y = -1) = \frac{1}{\sqrt{2\pi\sigma_{-1}^2}} e^{-\frac{(f-u_{-1})^2}{2\sigma_{-1}^2}}. \quad (39)$$

Normalizing (37) by its numerator and combining exponential terms yields

$$p(y = 1 | f) = \frac{1}{1 + \frac{P(f | y = -1)}{P(f | y = 1)} \frac{P_{-1}}{P_1}} = \frac{1}{1 + \frac{P_{-1}}{P_1} \frac{\frac{1}{\sqrt{2\pi\sigma_{-1}^2}} e^{-\frac{(f-u_{-1})^2}{2\sigma_{-1}^2}}}{\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(f-u_1)^2}{2\sigma_1^2}}}}, \quad (40)$$

which, after simplification gives the form

$$p(y = 1 \mid f) = \frac{1}{1 + \frac{P_{-1}}{P_1}\frac{\sigma_1}{\sigma_{-1}} e^{-\frac{1}{2}\left[\left(\frac{f-u_1}{\sigma_1}\right)^2 - \left(\frac{f-u_{-1}}{\sigma_{-1}}\right)^2\right]}}. \tag{41}$$

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

$$p(y = 1 \mid f) = \frac{1}{1 + \frac{P_{-1}}{P_1} e^{-\frac{1}{2\sigma^2}((u_1^2 - u_{-1}^2) + 2f(u_{-1} - u_1))}} = \frac{1}{1 + e^{(Af + B)}}. \tag{42}$$

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 1 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 (42), 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. Connectionists 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 still 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 a into 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 5. While the percentage of the segment that is allocated to each subsection can be manipulated, the performance of the HMM/SVM system is invariant to changes in the proportions [27,63].

System architecture

The hybrid HMM/SVM system is built using the rescoring paradigm shown in Figure 6. 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

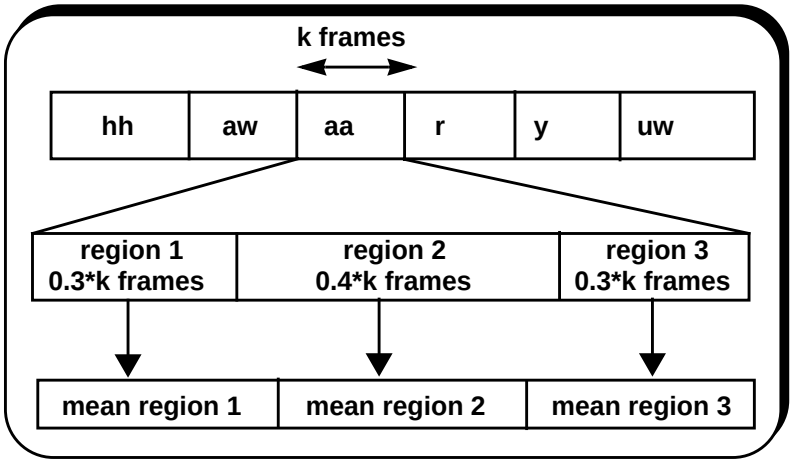

Figure 5. Composition of the segment level feature vector assuming a 3-4-3 proportion for the three sections.

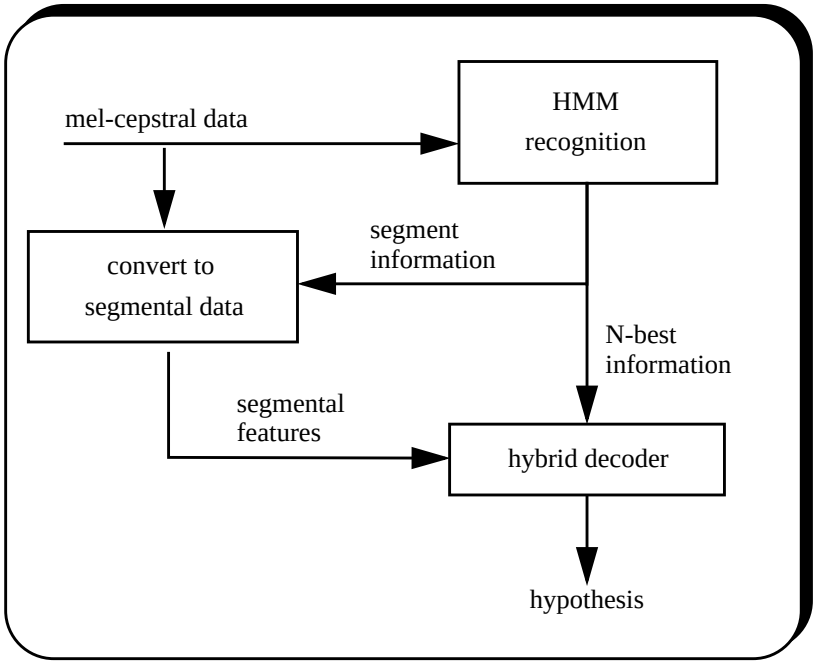

Figure 6. Flow graph for hybrid HMM/SVM system [27].

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/

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.

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 (unpublished) 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.

2.4. 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 machines (RVMs) which are 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.

CHAPTER 3

BAYESIAN METHODS AND THE RELEVANCE VECTOR MACHINE

While the SVMs presented in the previous chapter provide an excellent classification paradigm, they suffer from two serious drawbacks that hamper their effectiveness in speech recognition. First, while sparse, the size of the SVM models (number of non-zero weights) tends to scale linearly with the quantity of training data. For a large speaker-independent corpus such as SWB this effect becomes prohibitive. Techniques have been developed to overcome these problems [83], but they typically involve approximations which can only attempt to insure that the location of the model on the error surface remains reasonably close to optimal. We prefer methods where this sparse optimization is implicit in the training of the model. As will be explained shortly, there are a class of Bayesian methods that provide just such a framework.

Second, the SVMs are binary classifiers which are only capable of producing a yes/no decision. In speech recognition this is an important disadvantage since there is significant overlap in the feature space which can not be modeled by a yes/no decision boundary [59]. Further, the combination of disparate knowledge sources (such as linguistic models, pronunciation models, acoustic models, etc.) requires a method for combining the scores produced by each model so that alternate hypotheses can be

compared. Thus, we require a probabilistic classification which reflects the amount of uncertainty in our predictions. Efforts [27,71,72] have been made to build posterior probability estimates from the SVM models by mapping the SVM distances to a sigmoid function. While this does build a posterior estimate, Tipping [44, Appendix D] argues quite effectively that the sigmoid estimate is unreliable and that it tends to overestimate the model's confidence in its predictions.

In this chapter, we introduce a Bayesian approach due to MacKay [47,48] that incorporates an automatic relevance determination (ARD) prior over each model parameter. This tends to force most of the parameters to zero, leading to a sparse model representation. A kernel-based learning technique termed the *Relevance Vector Machine* (RVM) [44,45] is an application of ARD methods that is explored in this dissertation. Key to the RVM approach is the fact that only those parameters which are truly relevant to accurate modeling are retained. Thus, sparseness in the RVM model is automatically produced. In many cases, the RVM requires over an order of magnitude fewer parameters than the SVM [44,84,85,86] under equal conditions while producing generalization performance on par with the SVM. Further, the RVM approach is built from a fully probabilistic framework. This avoids the rather clumsy coupling of the model to the probability space as was necessary with the SVM.

3.1. Bayesian Methods

In the speech problem defined earlier, the task of learning amounted to finding the values of the parameters in our model that best matched the training data. The hope was

that, given sufficient training data, the model would generalize to unseen test sets. Implicit in this problem was choosing a model that was best suited to the speech task. We examined two three possible models thus far: the HMM/GMM, HMM/ANN and HMM/SVM systems. Embodied in this discussion are the two primary inference tasks of data modeling [47]. First, assuming that a particular model is true, we seek to infer the values for the parameters of the model that best fit the data at hand. This is exactly the training process given earlier — e.g. we presume the ANN topology and proceed to use back propagation to find the optimal weights. The second level of inference is one that we have not addressed closely to this point. That is the problem of inferring which model is most appropriate given the data at hand, or model comparison.

A first-cut approach to model comparison might dictate that we simply choose the model that fits the data best — the maximum likelihood solution. However, a more complex model can always fit the data better. Jaynes [87] describes an extreme interpretation of this problem where we would always choose the so-called *Sure Thing* hypothesis, under which exactly the training set and only training set is possible. Though it is the maximum likelihood solution, the Sure Thing hypothesis is intuitively displeasing and is counter to our desire for a solution which generalizes. We avoid choosing the Sure Thing hypothesis by expressing an *a priori* preference for simpler solutions. This preference for simple theories is given by a rather famous principle of modeling known as Occam's Razor.

Models studied previously, such as ANNs and SVMs have developed methods for dealing with generalization — cross-validation for instance. Bayesian

methods [87,88,89,90], on the other hand, provide a natural and quantitative embodiment of Occam's razor [48] as will be demonstrated shortly. First, the notation for Bayesian methods needs to be developed (we will follow the notation of MacKay [48]). The first level of inference requires that we find the best-fit parameters. We can write this probabilistically as $P(\boldsymbol{w}|D, H_i)$, where $\boldsymbol{w}$ is the set of adjustable parameters, D is the data from which we will make all inferences, and H_i is the overall model of the world including the form of the model, etc. Using Bayes' rule, we can rewrite this as

$$P(\boldsymbol{w}|D, H_i) = \frac{P(D|\boldsymbol{w}, H_i)P(w|H_i)}{P(D|H_i)} \tag{43}$$

Gradient methods are typically applied to find a optimal setting of $\boldsymbol{w}$. The denominator, termed the *evidence* for the hypothesis H_i, is usually ignored during the first level of inference because it is not needed in finding the most probable parameter settings, $\hat{\boldsymbol{w}}$.

The second level of inference requires the comparison of competing hypotheses, H_1 and H_2, by finding which of $P(H_1|D)$ and $P(H_2|D)$ is maximum. Setting this problem as a ratio of probabilities and using Bayes' rule gives

$$\frac{P(H_1|D)}{P(H_2|D)} = \frac{P(D|H_1)P(H_1)}{P(D|H_2)P(H_2)}. \tag{44}$$

If we assume that the competing hypotheses are *a priori* equiprobable (i.e. $P(H_1) = P(H_2)$), then the best hypothesis is chosen by evaluating the evidence, $P(D|H_i)$. The evidence is computed by marginalization across the model parameters:

$$P(D|H_i) = \int P(D|\boldsymbol{w},H_i)P(\boldsymbol{w}|H_i)d\boldsymbol{w}. \quad (45)$$

It is usually impractical to compute the integration, so MacKay [47,48] prescribes an analytical approximation to the evidence computation. Under the assumption that the posterior probability in (43), $P(\boldsymbol{w}|D,H_i) \approx P(D|\boldsymbol{w},H_i)P(\boldsymbol{w}|H_i)$, is well-approximated by a Gaussian, the integrand in (45) can be assumed to have a strong peak at the most probable value of the parameters, $\hat{\boldsymbol{w}}$. The evidence can then be approximated by multiplication of the height of the integrand and the width of the posterior, $\Delta\boldsymbol{w}$. This is depicted in Figure 7.

The evidence is approximated by

$$P(D|H_i) \approx P(D|\hat{\boldsymbol{w}},H_i)P(\hat{\boldsymbol{w}}|H_i)\Delta\boldsymbol{w}. \quad (46)$$

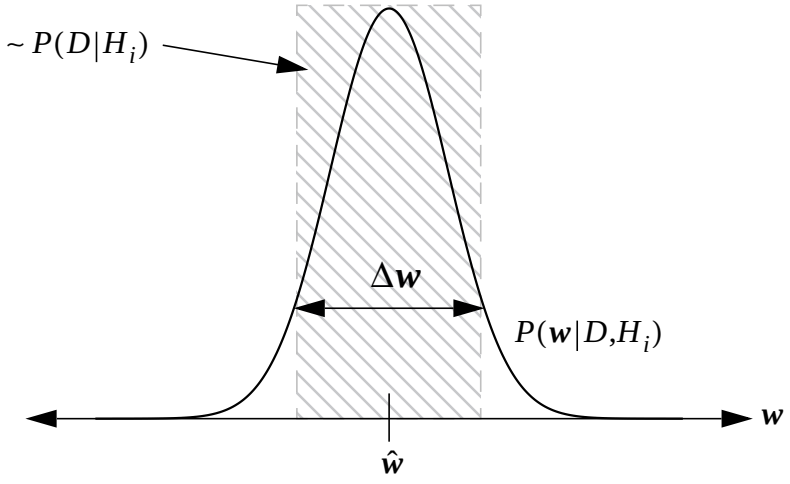

Figure 7. Evidence approximation for a single hypotheses. If the Gaussian assumption for the posterior, peaked about $\hat{\boldsymbol{w}}$, is not a good one then other methods must be employed. The width, $\Delta\boldsymbol{w}$, is the posterior uncertainty in our estimate of $\hat{\boldsymbol{w}}$ and can be determined by computing the error bars from the posterior.

where the term $P(D|\hat{\boldsymbol{w}},H_i)$ is the likelihood of the data given the best-fit parameter set and $P(\hat{\boldsymbol{w}}|H_i)\Delta\boldsymbol{w}$ is a penalty on the range of $[0, 1]$ which measure of how well our posterior distribution on $\boldsymbol{w}$ fits with our prior specification. As shown in Figure 8, a more complex model would be expected to have a smaller prior probability for $\hat{\boldsymbol{w}}$, $P(\hat{\boldsymbol{w}}|H_i)$, than a less complex model and thus would be penalized more. This is precisely how the evidence embodies Occam's razor — all other things being equal, a less complex model is preferred. The evidence provides a natural trade-off between the best-fit likelihood and the Occam factor. This concept is closely related to other 'penalizing' methods such as the Minimum Description Length [91] and the Bayesian Information Criteria [92] where the model is directly penalized by the number of parameters used. A similar idea was also

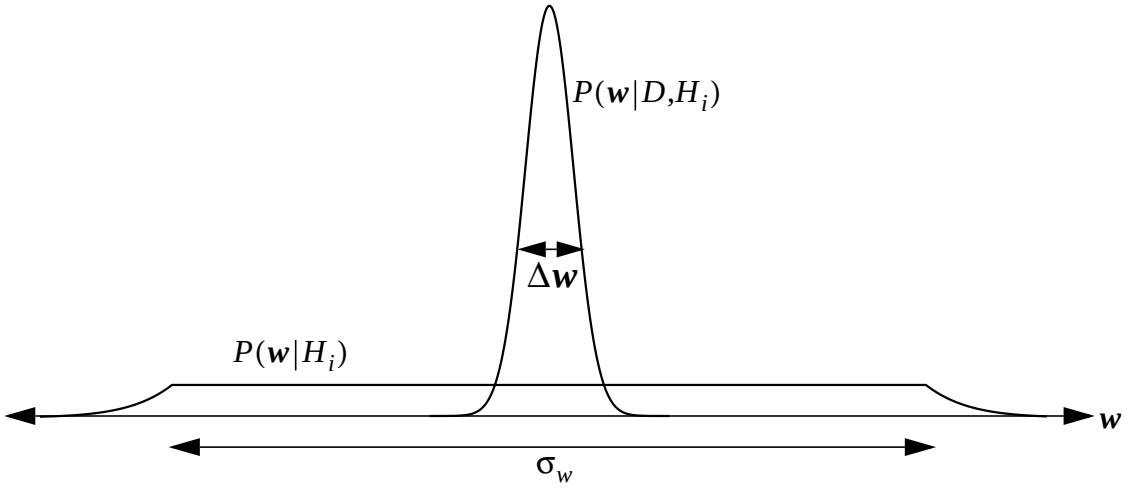

Figure 8. The prior distribution on the parameters in conjunction with the posterior distribution width determine the Occam factor. $\Delta w/\sigma_w$ determines the penalty incurred for choosing the model, H_i. A model with with more paramters will tend to have a larger σ_w. Thus, the penalty for such a model will be larger. The evidence defines the trade-off between posterior likelihood and model complexity (generalization) in the Bayesian framework.

seen in the SVM models which penalized models with too large a capacity (VC dimension) [37].

A impediment for the acceptance of Bayesian methods in the past has been the belief that they required the specification of subjective priors, thus making their results meaningless. While necessary, the priors are rarely subjective. Rather they are meant to represent our prior state of belief in the nature of the problem. The strength of Bayesian methods is that they allow us to quantitatively explore our prior beliefs. The failure of a model can be analyzed in terms of the priors and the priors can be adjusted appropriately. The Bayesian methodology also prescribes a principled manner of dealing with our lack of prior knowledge using maximum entropy arguments [87,90]. This prescription is best stated by Jaynes [87] when he says that

> "out of all distributions, p_i, that agree with the constraints, the one that maximizes the Shannon entropy represents the 'most honest' description of our state of knowledge".

This principle will be used often in the following sections when we seek to define prior probabilities over the parameters, $\boldsymbol{w}$, where $\boldsymbol{w} \in (-\infty, \infty)$. In these cases a zero-mean Gaussian (maximum entropy) prior will be used which indicates our prior belief that most parameters should go to zero, yielding a smooth model.

3.2. MacKay's Evidence Framework and Automatic Relevance Determination

MacKay [47] was the first to apply the evidence framework to regression and classification problems using ANNs. A summarization of his analysis is given now.

Defining the training data set as $D = \{\boldsymbol{o}, \boldsymbol{t}\}$, our goal in neural network learning is to find the set of weights, $\boldsymbol{w}$, such that a global error term, $E_D(\boldsymbol{w})$, is minimized. Typically $E_D(\boldsymbol{w})$ takes the form of a squared error as in the sum squared error in back propagation

$$E_D(\boldsymbol{w}) = \sum_{j=1}^{|D|} \sum_i (t_i^{(j)} - y_i(\boldsymbol{o}^{(j)};\boldsymbol{w}))^2, \tag{47}$$

where $t_i^{(j)}$ is the i'th component of the j'th target output and $y_i(\boldsymbol{o}^{(j)};\boldsymbol{w})$ is the i'th output of the ANN when presented with training sample $\boldsymbol{o}^{(j)}$ when the weights are set to $\boldsymbol{w}$. To discourage overfitting, a weight decay or regularizer term may be added which penalizes large w_i, for example

$$E_W = \frac{1}{2}\sum_i w_i^2. \tag{48}$$

The objective function for learning thus becomes

$$M(\boldsymbol{w}) = \beta E_D + \alpha E_W. \tag{49}$$

We can give a probabilistic interpretation for (49) if we consider the neural network outputs to be perturbed by a Gaussian noise process so that

$$t_i^{(j)} = y_i(\boldsymbol{o}^{(j)};\boldsymbol{w}) + \varepsilon \tag{50}$$

where ε is a zero-mean Gaussian noise process with variance equal to $1/\beta$. Then,

$$p(t_i^{(j)} \big| \boldsymbol{w}, \beta, H) = N(t_i^{(j)} \big| y_i(\boldsymbol{o}^{(j)};\boldsymbol{w}), 1/\beta) \tag{51}$$

specifies a Gaussian distribution over $t_i^{(j)}$ with mean $y_i(\boldsymbol{o}^{(j)};\boldsymbol{w})$ and variance $1/\beta$. The total probability of the data given the model (using the log of the sum-squared error condition) can then be written as

$$P(D|\boldsymbol{w}, \beta, H) = \frac{1}{Z_D(\beta)} e^{-\beta E_D}, \tag{52}$$

where $Z_D(\beta)$ is the Gaussian normalization term. Likewise, the log of the weight decay term, E_W, can be interpreted as a prior probability over the parameters so that

$$P(\boldsymbol{w}|\alpha, H) = \frac{1}{Z_W(\alpha)} e^{-\alpha E_W}. \tag{53}$$

With E_W as given in (48), $P(\boldsymbol{w}|\alpha, H)$ is a zero-mean Gaussian whose width is defined by $1/\alpha$. Finally, we have that

$$P(\boldsymbol{w}|D, \alpha, \beta, H) = \frac{P(D|\boldsymbol{w}, \beta, H)P(\boldsymbol{w}|\alpha, H)}{P(D|\alpha, \beta, H)}, \tag{54}$$

where, substituting (52) and (53) gives us

$$P(\boldsymbol{w}|D, \alpha, \beta, H) = \frac{\frac{1}{Z_D(\beta)} e^{-\beta E_D} \frac{1}{Z_W(\alpha)} e^{-\alpha E_W}}{P(D|\alpha, \beta, H)} = \frac{1}{Z_M} e^{-M(\boldsymbol{w})}. \tag{55}$$

It should be noted that binary and multi-class classification networks can be handled in a similar manner [47]. We simply replace the sum-square error function by a log-likelihood function, $G(\boldsymbol{w})$. The parameter, β, is not necessary in this case.

Application of (55) has the expected consequence that, by minimizing the objective function, (49), we are maximizing the probability of the weights given the

constraints. Finding the most probable weights, however, is not the end of the problem. Two parameters, α and β, have been introduced which need to be estimated. Note that as α is increased, the probability distribution of the decay terms becomes peaked about zero and smoother interpolants are favored. However, a value of α that is too large (i.e. too narrow a Gaussian) may limit the ability of the system to model a complex data set. As α is decreased, more complex interpolants are allowed. Here we have the first application of the Occam factor as described above — we must find the α that provides sufficient flexibility to model the training data set without allowing so complex a model that overfitting is encouraged. A similar argument can be made for β.

Under typical statistical methods, we might turn to cross-validation to find suitable values for these two parameters, but Bayesian methods provide a natural and principled approach for estimating them using the data at hand. We can write down the probability of the two parameters given our state of knowledge as

$$P(\alpha, \beta | D, H) = \frac{P(D|\alpha, \beta, H)P(\alpha, \beta|H)}{P(D|H)}. \qquad (56)$$

Note that $P(D|\alpha, \beta, H)$ is the evidence for α and β and is the denominator in (54). Assuming we have no prior knowledge that would cause us to favor a particular value of α or β, we can find the optimal values for α and β by evaluating the evidence (if we did have prior knowledge, we would simply repeat the inference over α and β using the prior, $P(\alpha, \beta|H)$, similar to what was done in the optimization of $\boldsymbol{w}$. At some level of the

inference, we will arrive at a point where our prior knowledge is too weak to apply and then we evaluate the evidence).

Unfortunately, a maximum for $P(D|\alpha, \beta, H)$ can not be found analytically in this case, so we proceed with an approximation due to MacKay [47] and Gull [90]. Under the assumption that the posterior distribution, (55), can be adequately approximated as a Gaussian, α and β can be updated as

$$\hat{\beta} = \frac{N - \gamma}{2E_D} \text{ and} \tag{57}$$

$$\hat{\alpha} = \frac{\gamma}{\sum_i (\hat{w}_i)^2}, \tag{58}$$

where N are the number of training points, $\hat{\boldsymbol{w}}$ are the most probable weights found by maximizing (55), γ is a measure of the number of parameters which are well-determined by the training data and is given by

$$\gamma = k - \alpha \text{Trace}(\Sigma). \tag{59}$$

Σ is the covariance of the assumed posterior Gaussian and defines error bars on the parameters, $\boldsymbol{w}$. Σ is found by computing the Hessian of the objective function given in (55). Iterative application of (55) and (56) provides optimal values for the system parameters, $\hat{\boldsymbol{w}}$, $\hat{\alpha}$, and $\hat{\beta}$, under the set of Gaussian assumptions.

For the explanation above, it was assumed that only one parameter, α, was used to control the complexity of all parameters in the system. In practice, we may want to group parameters of the system and control the complexity of each group separately. This

requires little change to the above formulation. We now assume a Gaussian prior for each class, c, of parameters so that

$$P(\{w_i\}|\alpha_c, H) = \frac{1}{\prod Z_{W(c)}} e^{\left(-\sum_c \alpha_c E_{W(c)}\right)}, \tag{60}$$

where

$$E_{W(c)} = \sum_{i \in c} w_i^2/2 \tag{61}$$

and proceed with the optimization as above. In the extreme case, a control parameter, α_i, can be assigned to each weight, w_i. This extreme application of the Bayesian prior as a control parameter is known as the method of *automatic relevance determination* (ARD). It is so named because the prior over the input unit weights in a neural network can 'shut-off' those input dimensions which are irrelevant to the problem at hand. ARD is at the heart of the relevance vector machines that will be described next.

3.3. Relevance Vector Machines

All of the above analysis has been done in terms of neural network training. An application of the evidence framework to kernel machines is the relevance vector machine (RVM) [44,45]. As with SVMs, the RVMs are formed by defining a vector-to-scalar mapping as a weighted linear combination of basis functions,

$$y(\boldsymbol{o};\boldsymbol{w}) = w_o + \sum_{i=1}^{M} w_i \phi_i(\boldsymbol{o}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{o}), \tag{62}$$

where $\boldsymbol{w} = \left[w_o, w_1, \ldots, w_M\right]^T$ and $\Phi = \left[1, \phi_1(\boldsymbol{o}), \ldots, \phi_M(\boldsymbol{o})\right]^T$ is a set of M functions that each form a, generally nonlinear, mapping of the observed vector, $\boldsymbol{o}$, to a scalar. The weights, $\boldsymbol{w}_i$, are the parameters to be tuned to produce an accurate model (under some appropriate measure) of the phenomena we desire to learn. At this stage, it is important to note the form of the basis functions, ϕ_i. Since SVMs are optimizing a distance measure in the transform space, they require that the basis functions take the form of a so-called Mercer kernel [38] (i.e. a kernel which acts as a dot-product in some space). No such restriction is placed on the basis functions that can be employed by the RVM. However, the power demonstrated by kernel machines gives compelling reason to pursue this special form of the basis function.

We reformulate (62) as

$$y(\boldsymbol{o};\boldsymbol{w}) = w_o + \sum_{i=1}^{M} w_i K(\boldsymbol{o};\boldsymbol{o}_i), \tag{63}$$

where there is one weight, w_i, associated with each training vector and $K(\boldsymbol{o};\boldsymbol{o_i})$ defines a kernel function (not necessarily a Mercer kernel). Due to the large number of parameters in this model — one per observation — we must guard against overfitting of the model to the training data. SVMs use the control parameter, C, to implicitly balance the trade-off between training error and generalization. RVMs take a Bayesian approach and explicitly define an ARD prior distribution over the weights

$$p(\boldsymbol{w}|\alpha) = \prod_{i=0}^{N} N\left(w_i|0, \frac{1}{\alpha_i}\right) = \frac{1}{\sqrt{(2\pi)^{N+1}|\boldsymbol{A}^{-1}|}} e^{-\frac{1}{2}\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w}} \tag{64}$$

where we have defined $\boldsymbol{A} = diag(\alpha_o, \alpha_1, \ldots, \alpha_N)$. This prior acts to force weak components of the model toward a weight of zero, thus finding the inputs that are relevant to modeling.

Each weight in the RVM model has an individual hyperparameter, α_i, that is iteratively reestimated as part of the optimization process. As the α_i grows larger, the prior on w_i becomes infinitely peaked around zero, forcing w_i to go to zero and, thus, contributing nothing to the summation in (63). This process automatically embodies the principle of Occam's Razor because it explicitly seeks the simplest model that satisfies the data constraints. In practice, the majority of the weights are pruned, resulting in an exceedingly sparse model with generalization abilities on par with SVMs [44]. To complete the Bayesian specification of the model, we have to specify a prior probability over the α_i. In practice we use a non-informative (flat) prior to indicate a lack of preference [44].

With SVMs the form of (63) arises from the need to optimize the classification margin in a high-dimensional space. With RVMs, however, the goal is to directly model the posterior probability distribution. The posterior is, thus, formed by generalizing the linear model to a probability distribution with a sigmoid link function,

$$\sigma(y) = \frac{1}{1 + e^{-y}}, \tag{65}$$

and adopting the two-class Bernoulli distribution for $P(\mathrm{t}|\boldsymbol{o})$ to give

$$\mathrm{P}(\mathrm{t}_i|\boldsymbol{w}, \boldsymbol{o}_i) = [\sigma\{\mathrm{y}(\boldsymbol{o}_i;\mathbf{w})\}]^{\mathrm{t}_i} \; [1 - \sigma\{\mathrm{y}(\boldsymbol{o}_i;\mathbf{w})\}]^{1-t_i} \tag{66}$$

where $t_i \in \{0, 1\}$ — an integrated multiple-class approach is also defined but due to computational concerns is less favorable than a set of one-versus-all classifiers. Under the assumption that each data sample is drawn independently, the likelihood of the training data set can be written as

$$P(\boldsymbol{t}|\boldsymbol{w}, \boldsymbol{O}) = \prod_{n=1}^{N} \sigma_n^{t_n}(1-\sigma_n)^{1-t_n} \tag{67}$$

where $\sigma_n = \sigma\{y(\boldsymbol{o}_n;\boldsymbol{w})\}$.

The objective of training is to find a parameter set which yields a model that is well-matched to the training data. In mathematical terms we want to find

$$(\hat{\boldsymbol{w}}, \hat{\alpha}) = \underset{\boldsymbol{w},\, \alpha}{\operatorname{argmax}} \; p(\boldsymbol{w}, \alpha|\boldsymbol{t}, \boldsymbol{O}). \tag{68}$$

Using Bayes' rule and (67), we can form (68) as finding $\boldsymbol{w}$ and α that maximize

$$p(\boldsymbol{w}, \alpha|\boldsymbol{t}, \boldsymbol{O}) = \frac{p(\boldsymbol{t}|\boldsymbol{w}, \alpha, \boldsymbol{O})\, p(\boldsymbol{w}, \alpha|\boldsymbol{O})}{p(\boldsymbol{t}|\boldsymbol{O})}. \tag{69}$$

A closed form solution to this maximization is not possible so we use the iterative approximation used by MacKay [47] which was described earlier.

1. For a fixed α, find the locally most probable weights $\hat{\boldsymbol{w}}$. In other words, we want to find the $\boldsymbol{w}$ that maximizes $p(w|\boldsymbol{t}, \alpha, \boldsymbol{O})$. This is equivalent to maximizing $P(\boldsymbol{t}|\boldsymbol{w}, \boldsymbol{O})\, p(\boldsymbol{w}|\alpha)$. Taking the logarithm of this quantity and ignoring the scale factor on $p(\boldsymbol{w}|\alpha)$ which is a constant due to the fixed α we can write

$$
\begin{aligned}
L &= \log\{P(\boldsymbol{t}|\boldsymbol{w}, \boldsymbol{O})p(\boldsymbol{w}|\alpha)\} \\
&= \log\left[\prod_{n=1}^{N} \sigma_n^{t_n}(1-\sigma_n)^{1-t_n} \prod_{i=0}^{N} N\left(w_i|0, \frac{1}{\alpha_i}\right)\right] \quad . \\
&= \sum_{n=1}^{N} [t_n \log(\sigma_n) + (1-t_n)\log(1-\sigma_n)] - \frac{1}{2}\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w}
\end{aligned} \tag{70}
$$

The gradient and Hessian of L are found by differentiating with respect to $\boldsymbol{w}$

$$\nabla_{\boldsymbol{w}} L = \Phi[\boldsymbol{t} - \boldsymbol{y}] - \boldsymbol{A}\boldsymbol{w} \tag{71}$$

$$\nabla_{\boldsymbol{w}}\nabla_{\boldsymbol{w}} L = -(\Phi^{\boldsymbol{T}}\boldsymbol{B}\Phi + \boldsymbol{A}), \tag{72}$$

where $\boldsymbol{B} = diag(\beta_1, \beta_2, \ldots, \beta_N)$ with $\beta_n = \sigma_n(1-\sigma_n)$, and Φ is the N x N+1 matrix defined by $\Phi = [\phi(x_1), \phi(x_2), \ldots, \phi(x_N)]^T$ with

$$\phi(x_n) = [1, K(x_n, x_1), K(x_n, x_2), \ldots, K(x_n, x_N)]^T. \tag{73}$$

The Hessian defined in (72) is negative-definite everywhere, and therefore defines a unimodal, log-concave surface — this is easy to see since $\boldsymbol{B}, \Phi^T\Phi$ and $\boldsymbol{A}$ contain only positive entries. We can, thus, use second-order Newton methods to solve for the $\boldsymbol{w}$ that maximizes L with an assuredness that the process will converge.

2. The Hessian is negated and inverted and to give an approximation to the covariance of a Gaussian posterior over the weights, centered about $\hat{\boldsymbol{w}}$

$$\Sigma = (\Phi^{\boldsymbol{T}}\boldsymbol{B}\Phi + \boldsymbol{A})^{-1} \tag{74}$$

3. Using Σ and $\hat{\boldsymbol{w}}$ as the covariance and mean, respectively, of the Gaussian approximation, we can follow MacKay's approach [47] to update the $\{\alpha_i\}$ by

$$\alpha_i = \frac{\gamma_i}{\hat{w}_i^2}, \qquad \gamma_i = 1 - \alpha_i \Sigma_{ii}. \tag{75}$$

This iterative procedure is repeated until suitable convergence criteria are met. Central to this iterative method is the second-order Newton maximization of $P(\boldsymbol{t}|\boldsymbol{w}, \boldsymbol{O})p(\boldsymbol{w}|\alpha)$ requiring an $O(N^3)$ inversion operation. As the quantity of training data increases, this becomes prohibitive. SVMs have a similar problem with scaling up that has been addressed through iterative refinement of the training set [40]. Current research is focusing on similar methods for RVMs [93].

3.4. Summary

In this chapter, sparse Bayesian methods have been examined. These methods use the evidence framework to compare potential models. Sparsity is explicitly encouraged in the model through the invocation of an ARD prior. The RVM is a kernel machine that is formed as a special case of this methodology. The form of the RVM is similar to the SVM but it overcomes some of the drawbacks of the SVM paradigm. Namely, the RVM provides superior sparsity with little to no degradation in generalization. The RVM also operates as a purely probabilistic model so there is no need to coax probabilities out of the model as was necessary with the SVM. In this dissertation, we will exploit the advantages

of the RVM on the speech recognition problem. The next chapter will describe the framework, data, and experiments that will be employed by this dissertation.

CHAPTER 4

PROPOSED WORK AND EXPERIMENTS

The HMM/SVM hybrid framework defined by Ganapathiraju and colleagues [27,63] gave improved performance for continuous speech recognition tasks. In particular, the oracle results showed the promise of discriminative kernel methods. However, the results also exposed the shortcomings of the presented framework. The nature of the SVM model necessitated the use of ad hoc procedures for estimating the segmentation of the speech data as well as the posterior probability distributions, neither of which seemed to reap the full power of the SVM model. Further, the HMM/SVM hybrid framework was not able to make full use of the power of HMMs to automatically find segmentations which maximize the likelihood of the data given the SVM model. Nor was it able to leverage the existing methods common to HMM/GMM systems such as iterative EM training.

In this work, we seek to use the HMM/SVM work as a springboard to an integrated solution which follows the form of standard HMM/GMM systems. Yet, we desire to retain the power to define nonlinear decision regions by discriminatively training in a high-dimensional space via kernels. The Bayesian formulation of the RVM provides exactly these benefits. The primary contribution of this thesis work will be to motivate, investigate, define and implement a set of theoretically well-founded techniques for

estimating and evaluating an HMM/RVM continuous speech recognition system. This new methodology will address three primary issues:

1. **Integrated, iterative HMM/RVM training**: The results of the HMM/SVM hybrid system [27,63] indicate a need to automatically incorporate segmentation variation into the training process. HMMs offer a principled approach to this problem via the EM-based Baum-Welch algorithm [19]. In this work we propose to create a similar algorithm for training HMM/RVM systems. The RVM will replace the Gaussian as the frame-level emission distribution in the HMM state. Iterative reestimation formulae which describe cycles of Baum-Welch statistical accumulation (the expectation step) followed by Bayesian RVM training (maximization step) will be derived. In building this training algorithm we must address issues of iterative and monotonic convergence and stopping criteria. Similar work that has been developed for connectionist HMM/ANN systems [35,36] will serve as reference.

2. **Practical optimization methods**: As with SVMs, the process to train an RVM classifier is computationally expensive even for small problems. For the RVM, though, this is primarily due to the need for inversion of the Hessian matrix which is an $O(M^3)$ operation requiring $O(M^2)$ memory, where M is the number of basis functions (also the number of non-zero multipliers). At initialization M is set to the size of the training corpus, N. Since our new training paradigm would replace the HMM emission distribution by an RVM, the RVM would be exposed to every

frame of data in the training corpus. For even small speech corpora the number of frames in the training set is on the order of 10^6. The usual RVM training methods described previously are rendered impractical.

There are three immediate avenues for research on this problem. The first is to define a technique analogous to the *chunking algorithm* [40] used in efficient SVM optimization. With this, the RVM optimization problem can be decomposed into a set of smaller problems whose respective solutions can be combined to form a solution for the full problem, all while insuring optimality (or near-optimality). Faul and Tipping [44,93] refer briefly to a second constructive method in which the optimization begins with a single basis function and others are added or deleted as the optimization proceeds. Finally, an active data selection mechanism akin to that described by MacKay [47] may be defined. Similar to Tipping's method, MacKay's is constructive in the sense that only those data points which are expected to add significant new information (e.g. those which are likely to have non-zero multipliers) are added to the optimization.

3. **Integrated hierarchical Viterbi-HMM/RVM decoder**: Finally, we will build upon the ISIP hierarchical HMM decoder [94] to create an HMM/RVM decoder. Construction of this HMM/RVM decoder primarily requires the replacement of the Gaussian core with the trained RVM models. The remainder of the decoder machinery remains unchanged, though some tuning of parameters will be necessary for each experimental task.

4.1. Corpora

The work in this thesis will incorporate three corpora which cover the full range of continuous speech corpora:

1. The Deterding vowel [95] set is a publicly available vowel classification task. This is a good data set to evaluate the efficacy of static RVM classifiers and to compare their performance to SVM classifiers on speech data since it has been used as a standard benchmark for several non-linear classifiers for several years. In this evaluation, the speech data was collected at a 10 kHz sampling rate and low pass filtered at 4.7 kHz. The signal was then transformed to 10 log-area parameters, giving a 10 dimensional input space. A window duration of 50 msec. was used for generating the features. The training set consists of 528 observations from eight speakers and the test set consists of 462 observations from a different set of seven speakers. The speech data consisted of 11 vowels uttered by each speaker in a h*d context. This data set is one of the most widely used for benchmarking non-linear classifiers. Though it appears to be a simple task, the small training set and significant confusion in the vowel data make it a very challenging task.

2. The TIDigits corpus [96] consists of more than 25 thousand digit (“zero” through “nine” and “oh”) sequences spoken by over 300 men, women, and children. The data was collected in a quiet studio environment and digitized at 20 kHz. However, most experiments begin by downsampling the data to 8 kHz. The typical word-error rates on TIDigits is close to 1% so this corpus will not serve to prove

the superiority of the new methods. Instead we will use this corpus to provide a benchmark where experiments can be completed quickly — until the HMM/RVM system approaches state-of-the-art on this task there is no reason to continue on to more complicated tasks.

3. The OGI Alphadigits [81] corpus is a collection of about 78,000 examples from 3031 speakers saying strings of letters ("a"-"z") and digits ("zero"-"nine" and "oh") over the telephone. The data was recorded directly off of a digital T1 phone line without digital-to-analog or analog-to-digital conversion at the recording end. An 8kHz sampling rate was used. Experimentation on the Alphadigits corpus will follow directly from the TIDigits experiments since the form of the task is identical (open grammar on a small domain without a probabilistic language model). This will also give us our first comparison point with the hybrid HMM/SVM system. State-of-the-art word-error rates on this task are near 10%.

4. Switchboard [82] corpus consists of spontaneous conversations averaging six minutes in length. Over 500 speakers of both genders from every major dialect of American English are represented. The data is a digital version of speech signals collected directly from the telephone network over T1 lines by automatic switching software. The added confusability and conversational style as well as the addition of a high-perplexity stochastic language model makes this one of the most difficult tasks being tackled in recognition research today. Results from a basic

HMM/GMM system hover near 40%, while state-of-the-art systems are able to achieve error rates near 20%.

For the TIDigits task the standard, speaker-independent, open-loop training and test sets will be used. A proposed segmentation into training and test sets for the Alphadigits corpus has been defined [97] and will be used in all experimentation with that corpus. For the SWITCHBOARD corpus, the training and test sets created during the 1997 LVCSR Summer workshop at Johns Hopkins University [98] will be used since they were used in the HMM/SVM work [27] and will provide a point of comparison.

4.2. Preliminary Experiments

Preliminary experiments using RVMs have been run on a large number of data sets, both synthetic and real. A representative example of these is the Deterding vowel classification data. Table 2 gives the results and compares them to the SVM classifiers trained in [59]. Importantly, the RVM classifiers achieve superior performance to the SVM classifiers while utilizing nearly an order of magnitude fewer parameters. While we do not expect the superior error performance to be typical (on pure classification tasks) we do expect the superior sparseness to be typical. This sparseness property will be particularly important when attempting to build systems which are practical to train and test.

4.3. Planned Experiments

Before the integrated/iterative training methods can be tested on any (reasonably-sized) speech corpus, we must address the issue of practical training methods

Classifier	Error Rate	Average number non-zero weights
SVM	35.0%	82.8
RVM	30.3%	12.6

Table 2. Comparison of SVM and RVM classifiers on Deterding vowel data [95]. Each classifier type was trained as a set of 11 1-vs-all classifiers. The training and test set sizes for each classifier was 532 examples and 462 examples respectively. Both the SVM and RVM system used an RBF kernel with the variance parameter set to 0.7. For the SVM system, the trade-off parameter, C, was set to 10. The best performance reported thus far on this data is 29% using a speaker adaptation scheme called Separable Mixture Models [99].

for the static RVM classifiers. For this task, we will mirror the alphadigit segmental-modeling experiments performed using the hybrid HMM/SVM system [27]. This is a reasonable benchmark point for the proposed methods since these segmental models are trained on as many as 350 thousand training vectors. A key difference between the HMM/RVM and HMM/SVM segmental systems will be the posterior estimate. While the HMM/SVM system relied on an ML-fit of the posterior probabilities to a sigmoid, the HMM/RVM system will directly predict the posterior probability.

Our initial recognition experiments will use the TIDigits corpus. We first propose to build two sets of 5-state left-to-right word models with GMM emission probabilities and RVM emission probabilities respectively. The HMM/GMM models will be trained using the standard Baum-Welch algorithm on all of the TIDigits training data. The number of mixtures in the GMMs will be increased up to 16 mixtures. The HMM/RVM models will be trained using the training algorithm developed as part of this dissertation. The power of the RVM model should obviate the need for the mixtures, so no mixture model

will be generated for the RVM system. This initial training experiment will be key to the development of convergence criteria for the HMM/RVM training system.

Further experiments with the TIDigits corpus will build context-independent phone models as well as context-dependent cross-word phone models. Due to insufficient training data for some models, we will have to face the issue of parameter tying. Initial attempts at parameter tying will use the same tied state mapping as determined by the HMM/GMM system. Because the RVM model describes a probability distribution we can also examine a decision tree methodology which uses a cross-entropy measure between two RVM models to generate the tree of similar states. The test set will be evaluated by decoding a loop grammar (any number of words is possible and any word sequence is possible).

Extending the techniques to the alphadigits and SWB tasks should prove to be a trivial extension of the lessons learned in the TIDigits experiments, with the added inconvenience of an order of magnitude longer training and decoding time. To abbreviate the SWB decoding experiments, we will rescore lattices generated from an HMM/GMM cross-word triphone system. These lattices will have an inherent error rate of approximately 10-15%. Both the alphadigits and SWB systems will be built as cross-word, context-dependent triphone systems. The alphadigits system language model will be modeled as a loop grammar while the SWB system will use a trigram language model.

With all of these experiments, the question of what feature set to use arises. One objection we have to many of the previous results with discriminative models is that it is

difficult if not impossible to decouple the improvement due to the new learning machine from the improvement due to the increased feature set dimension. The authors would likely argue (and rightly so) that the ability to avoid the curse of dimensionality is an important feature of their model. However, when comparing two modeling paradigms, we would like to as much as possible be able to make an apples-to-apples comparison. For instance, if GMM models are no worse than the proposed discriminative models, then perhaps more effort should go into ways to train GMM models with larger feature sets.

Thus, we will initially use the same features for the HMM/RVM system as is used for the HMM/GMM system. These will include 12 FFT-derived cepstral coefficients and one energy coefficient along with the first and second derivatives of those 13 to constitute a single 39-dimensional feature vector generated for each 10 milliseconds of speech data. Using these features will allow us to determine if the RVM system is truly learning some modality of the data beyond what the GMM is able to learn. In further experiments, we will use an extended feature set to determine what additional information can be gained by the RVM. We will use a sliding window of frames of data along the lines of connectionist systems. The window size will vary from 5 frames to 15 frames.

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.