CONTINUOUS SPEECH RECOGNITION USING SUPPORT VECTOR MACHINES

Aravind Ganapathiraju, Jonathan Hamaker and Joseph Picone

Institute for Signal and Information Processing (ISIP)
Department of Electrical and Computer Engineering
Mississippi State University, Mississippi State, MS 39762
{ganapath, hamaker, picone}@isip.msstate.edu

Direct correspondence to:

Jonathan Hamaker
429 Simrall, Hardy Road
Box 9571
Mississippi State, MS 39762
USA

Email: hamaker@isip.msstate.edu
Phone: (662) 325-8335
Fax: (662) 325-3149

ABSTRACT

Hidden Markov models (HMMs) with Gaussian mixture observation densities are the dominant approach in speech recognition. These systems typically use a representational model based on maximum likelihood decoding and expectation maximization-based training. Though powerful, this paradigm is prone to overfitting and does not directly incorporate discriminative information. We propose a new paradigm centered on principles of structural risk minimization and using a discriminative framework for speech recognition based on support vector machines (SVMs). SVMs are a family of discriminative classifiers that provide significant advantages over other discriminatively trained classifiers. Chief among these advantages is the ability to simultaneously optimize the representational and discriminative ability of the acoustic classifier — a necessity for acoustic units such as phonemes which have a high degree of overlap in the feature space.

As a proof of concept, we present an SVM-based large vocabulary speech recognition system. This system achieves a state-of-the-art word error rate of 10.6% on a continuous alphadigit task. Through the introduction of this system, we provide insight into the many issues one faces when moving from an HMM framework to an SVM framework. These include the application of temporal constraints to the static support vector classifier, generation of a posterior probability from the binary support vector classifier and balancing the need for a robust training set with pragmatic efficiency issues. We conclude with a discussion of open research issues that are crucial to the successful application of SVMs in speech recognition.

1. INTRODUCTION

The goal in a statistically-based speech recognition system is to find the most likely word sequence given the acoustic data. If A is the acoustic evidence that is provided to the system and $W = w_1, w_2, \ldots, w_N$ is a sequence of words, then the recognition system must choose a word string $\hat{W}$ such that

$$\hat{W} = \underset{W}{\operatorname{argmax}}\ p(W/A) . \qquad (1)$$

The term $p(W/A)$ is known as the *a posteriori* probability and it is typically impossible to maximize this term due to the infinite number of possible word strings. Thus, we apply Bayes rule to give the form

$$\hat{W} = \underset{W}{\operatorname{argmax}}\ p(A/W)p(W) . \qquad (2)$$

The probability, $P(A/W)$, is typically provided by an *acoustic model* while $p(W)$ gives the *a priori* probability of the word sequence W being spoken as predicted by a *language model*.

In most speech recognition systems, the acoustic modeling components of the recognizer are based on hidden Markov models (HMMs) (Deller, Proakis & Hansen, 1993; Picone, 1990; Rabiner & Juang, 1993; Jelinek, 1997). The ability of the HMM to statistically model the acoustic and temporal variability in speech has been the main reason for their success. HMMs provide an elegant statistical framework for modeling speech patterns by modeling the temporal evolution of speech via an underlying Markov process (Picone, 1990). The probability distribution associated with each state in an HMM models the variability which occurs in speech across speakers or even different speech contexts. This distribution is typically a Gaussian mixture model since it provides a sufficiently general parametric model as well as a well-developed mathematical framework for estimation and analysis. On the other hand, non-parametric models are attractive because they can automatically accommodate unforeseen modalities in the data without having to assume the properties of an underlying distribution. However, non-parametric models have not found favor in speech recognition systems due to their computational complexity and their inefficient use of resources.

A key to the widespread use of HMMs to model speech can be attributed to the availability of efficient parameter estimation procedures (Rabiner & Juang, 1993). If we assume that parameters of an HMM are fixed but unknown, we can pose the problem of parameter estimation as one that maximizes the probability that the model generates the observed data (*a posteriori* probability). The approach for a typical HMM parameter estimation process then becomes maximization of the likelihood of the data given the model, traditionally known as maximum likelihood (ML) estimation (Jelinek, 1997). One of the most compelling reasons for the success of ML and HMMs has been the existence of iterative methods to estimate the parameters that guarantee convergence. The expectation maximization (EM) algorithm provides an iterative framework for ML estimation with good convergence properties, though it does not guarantee finding the global maximum (Dempster, Laird & Rubin, 1977; McLachlan, 1997; Redner & Walker, 1984).

There are, however, problems with an ML formulation for speech recognition. Key among these is that maximizing likelihood does not necessarily translate to better classifiers. The goal in building a classifier is to learn to distinguish between two, or more, distinct classes of data. Maximum likelihood improves a classifier's ability to *represent* a specific class. In this light, ML-based estimation is tangential to the goal of classifiers in general and speech recognizers in particular. The ML estimation process tries to optimize the modeling ability of the acoustic models without access to a measure of their classification ability. In reality, better classification is the ultimate goal of the speech recognizer. ML estimation is unable to guarantee this because it is not discriminatively trained — the model parameters are estimated based on in-class data alone without considering the out-of-class data.

A simple example, shown in Figure 1 illustrates this problem. The two classes shown are derived from completely separable uniform distributions. ML is used to fit Gaussians to these classes and a simple Bayes classifier is built. However, we see that the decision threshold occurs inside the range of class 2. This means that the probability of error is significant. However if we were to simply recognize that the range of data points in class 1 is less than 3.3 and that no data point in class 2 occurs within this range, we can achieve perfect classification. In this example any amount of effort expended in learning a better Gaussian model will not help achieve perfect classification. More dramatic examples can be constructed to show that learning decision regions discriminatively will help improve classification. The conclusion from the above example is not necessarily that using a Gaussian is an incorrect choice. However, it is clear from such examples that discrimination is a key ingredient for creating robust and more accurate acoustic models.

The primary difference between ML-based HMM parameter estimation and discriminative techniques is that the objective criterion in the latter includes the probability of the data given that the wrong model was used (McDermott, 1997). Under discriminative-based estimation, the optimization process can effectively trade-off rejection of out-of-class examples while simultaneously learning to optimally represent in-class examples. There has already been significant progress in the area of discriminative techniques for the estimation of HMM parameters. In HMM-based systems, the discrimination ability of the Gaussians is improved by using optimizing criteria like Maximum Mutual Information (MMI) (Woodland & Povey, 2000) and Minimum Classification Error (MCE) (McDermott, 1997) which include both in-class and out-of-class data in the estimation process. MCE is especially elegant in that it uses the fact that in order to achieve good classification, the estimation of the posterior probabilities is not as important as it appears. ML and MMI suffer from the fact that both the estimation procedures expend effort in modeling posteriors while not guaranteeing improved classification performance. Though both MCE and MMI estimation have had significant success in terms of improvements in recognition performance their use has thus far been limited because they require immense resources during training (Woodland & Povey, 2000).

Artificial neural networks (ANNs) represent a class of discriminative techniques that have been successfully applied to speech recognition (Bridle & Dodd, 1991; Bridle, 1989; Richard & Lippmann, 1991; Renals, 1990; Ström, 1997; Robinson, 1989). ANNs can learn very complex non-linear decision surfaces effectively. However, the estimation process for an ANN is significantly more computationally expensive than an HMM. Also, ANNs are typically constrained to classification problems involving static data. This has led to the development of several connectionist approaches in which neural networks are embedded in a HMM framework (Bourlard & Morgan, 1994; Cook & Robinson, 1997; Tebelskis, 1995). The performance of these hybrid systems has been competitive with many HMM-based systems with the added benefit that the ANN hybrids typically require a significantly fewer parameters (Bourlard & Morgan, 1994). The hybrid connectionist systems also provide a means for circumventing some of the assumptions made in HMM systems such as the assumption of independence of observations across frames (Holmes, 1997; Russell & Holmes, 1997; Ostendorf, Digalakis & Kimball, 1996; Ostendorf & Roukos, 1989; Austin, Zavaliagkos, Makhoul & Schwartz, 1992). Hybrid systems mitigate this problem by allowing the classifiers to operate on several frames of acoustic data (Russell & Holmes, 1997).

Despite the advantages of the ANN approach, their use has been limited for several reasons:

1. **Generalization**: ANNs have been known to overfit data unless cross-validation is applied. This can be restrictive when the amount of training data is limited.

2. **Model Topology**: In most connectionist hybrid systems the topology of the neural network classifiers needs to be fixed prior to the estimation process. Finding a near-optimal topology is rarely possible without expert knowledge of the data. Techniques exist to learn connections automatically but are prohibitive for large-scale tasks (Kirkpatrick, Gellatt & Vecchi, 1983, Bodenhausen, Manke, & Waibel, 1993).

3. **Optimality**: ANN learning is based on the principle of empirical risk minimization via the back-propagation algorithm (Rosenblatt, 1957; Rumelhart, Hinton & Williams, 1986; Ackley, Hinton & Sejnowski, 1985). Though this guarantees good performance on the training data, obtaining a bound on the performance on the test data is not easy.

4. **Convergence**: Convergence properties of the optimization process has been the biggest drawback of neural networks. Convergence is typically an order of magnitude slower than ML estimation of HMM parameters. Neither ML estimation using the EM methods nor ANN learning guarantee reaching a global maximum unless measures are taken to perturb the system from time to time (Lawrence, Giles & Tsoi, 1996; Lawrence, Giles & Tsoi, 1997).

At a high level, speech recognition can be viewed as a classification problem. In that respect one would expect better performance with classifiers that estimate decision surfaces directly rather than those that estimate a probability distribution across the training data. A Support Vector Machine (SVM) is one such machine learning technique that directly estimates the decision surface using a discriminative approach (Vapnik, 1998). Unlike other discriminative techniques, SVMs have demonstrated good generalization and proven to be successful classifiers on several classical pattern recognition problems (Burges, 1999). We argue that, through the principle of structural risk minimization (SRM), the SVM framework provides significant advantages over the ML techniques currently prevalent in HMM-based speech recognition systems. Unlike ML techniques, the SRM paradigm provides for control of the trade-off between generalization and closed-loop optimality.

In the next section we introduce the principles of structural risk minimization and large margin classifier design. Further, we discuss the use of kernels in SVMs for the development of non-linear decision surfaces. We present a hybrid SVM/HMM system in Section 3 that combines the temporal modeling power of the HMM with the superior classification performance of the SVM. Section 4 provides experimental evidence that an SVM/HMM combination is able to learn modalities in the decision surface that are not learned by Gaussian mixture models. We conclude with suggestions for future research on extensions to the hybrid SVM/HMM system. For brevity, a significant level of theoretical detail has been omitted but can be found in (Ganapathiraju, 2001).

2. SUPPORT VECTOR PARADIGM

The design of a classifier is essentially a process of learning a mapping of the input data to class labels. This is achieved using an optimization process (such as risk minimization) constrained by knowledge relevant to a specific classification problem. Two commonly used minimization techniques are empirical risk minimization (ERM) and structural risk minimization (SRM). Support vector machines are estimated based on SRM. The next few sections describe SRM by comparing and contrasting it with ERM. The theory of SVMs is also discussed in detail.

2.1. Risk Minimization

Let us assume that the training data consists of pairs, $(x_1, y_1), (x_2, y_2), \ldots$ where the x 's are the input observations and y 's are the target classes. The goal of a learning machine is to learn the mapping $y = f(x)$. We assume that the training data has been drawn randomly and independently based on the joint distribution $P(x, y)$. To learn the unknown mapping, we can either estimate a function that is close to the joint distribution under an appropriate metric or we can learn an optimal predictor of the system's output. In the former case, it is not sufficient for us to estimate a good predictor of the output. The goal is to estimate P, the joint distribution. However, for the purposes of data classification, we pursue the latter approach where the goal is to learn an optimal predictor.

The learning process is therefore a process of choosing a function from a set of functions defined by the construction of the learning machine. For example, in a neural network classifier, the problem reduces to that of finding the weights of the connections in a predefined network. Since the network structure has been defined *a priori,* the set of networks from which the optimal network needs to be chosen is a finite set. The optimal network is chosen based on some optimality criterion that measures the quality of the learning machine. Let us define a term, *risk,* which measures the quality of the chosen function. The operating space is a subset Z of an n-dimensional vector space R^n, which is the union of the input vector space and the output space. Let us also assume the existence of a set of functions $\{g(z)\}, z \in Z$ and a functional R, that measures the risk as,

$$R(g(z)) = \int L(z, g(z)) dP(z) \tag{3}$$

where L is an appropriately defined loss function and P is the previously defined joint probability distribution.

The problem of *risk minimization* can then be defined as one that minimizes the functional given by (3) for a specific training data set. In reality the minimization process involves finding the optimal parameterization for the function $g(z)$ which can be parametrically represented as $g(z, \alpha), \alpha \in \Lambda$. The minimization involves finding the best parameterization α^* such that $g(z, \alpha^*)$ is the optimal function from the set of functions $\{g(z, \alpha)\}$. The above parameterization does not necessarily imply that the problem is restricted to parametric forms since α can be a scalar, a vector or any other abstract functional element. With the above modifications to the definition of the minimization problem, the risk can be rewritten as,

$$R(\alpha) = \int Q(z, \alpha) dP(z), \qquad \alpha \in \Lambda, \tag{4}$$

where,

$$Q(z, \alpha) = L(z, g(z, \alpha)). \tag{5}$$

The function Q is now called the *loss function.* Choosing an appropriate value for α to minimize the functional defined in (4) is called *risk minimization.*

Minimizing the risk functional is not a trivial problem because the form or the parameterization of the joint distribution $P(z)$ is not known *a priori*. The problem can be simplified significantly if we minimize a variation of (4). For example, we can minimize the measured mean risk, or *empirical risk,* defined as,

$$R_{emp}(\alpha) = \frac{1}{l}\sum Q(z_i, \alpha), \qquad \alpha \in \Lambda. \tag{6}$$

We assume that we have access to l training observations $z_1, z_2, \ldots z_l$. R_{emp} is therefore the mean error computed from a fixed number of training samples under the assumption that the training samples are uniformly distributed. Minimization of (6) is called *empirical risk minimization (ERM)* and is one of the most commonly used optimization procedures in machine learning. ERM is computationally simpler than attempting to minimize the actual risk as defined in (4) since it circumvents the need for the estimation of the joint probability density function P. A variety of loss functions can be used for the optimization process. One such example is,

$$Q(x, y) = |y - f(x, \alpha)|, \tag{7}$$

where y is the output of the classifier and x is the input vector. This form of a loss function is common in learning binary classifiers. For example, to estimate the parameters of a multi-layered perceptron using the back-propagation algorithm, a loss function representing the squared error is used.

The issue of the quality of the learning machine is not addressed in its entirety when we consider ERM. There could be several configurations of the learning machine which give us the same empirical risk as indicated by Figure 3. How does one choose the best configuration? To better answer this question we need to analyze the relationship between the actual risk defined by (4) and the empirical risk defined by (6). Suppose that the minimum empirical risk is obtained using the function $Q(z, \alpha_l)$, where the subscript l is equal to the size of the training sample. Let the minimum actual risk be obtained using the function $Q(z, \alpha_0)$. There are two issues that need to be addressed. First, we need to know the risk achieved using $Q(z, \alpha_l)$. Second, we need to know how close this risk is to the risk obtained using $Q(z, \alpha_0)$.

Vapnik (Vapnik, 1995) proved that bounds exist for the actual risk such that,

$$R(\alpha) \le R_{emp}(\alpha) + f(h) \tag{8}$$

where h is the Vapnik-Chervonenkis (VC) dimension (Vapnik, 1995; Vapnik, 1998) and is a measure of the capacity of a learning machine to learn any training set. Thus, the VC dimension is directly related to the generalization ability of a learning machine (Burges, 1999) and is typically proportional to the number of free parameters (weights, mixture components, etc.) in the system. For example, if $f(h)$ is small, the machine generalizes well because the actual risk is guaranteed to be close to the empirical risk which has been already minimized via the principle of ERM.

To better qualify the above statement, suppose that the empirical risk obtained using the training data set is zero. This fixes R_{emp} and the actual risk is now bounded by $f(h)$. Suppose we now receive a new set of data that does not include any of the l examples used previously. For a machine that generalizes well, we should be able to predict with a high degree of confidence that the empirical risk obtained using this new data, $R_{emp}{}^{*}$, will also be close to zero. However, from equation (8), we note that the actual risk over the data can be as high as $f(h)$. Therefore when $f(h)$ is large, $R_{emp}{}^{*}$ can be as high as $f(h)$ and R_{emp} cannot be used to effectively predict the performance of the machine on an unseen data set. In other words, the machine fails to generalize well when $f(h)$ is large as shown in Figure 3.

For loss functions in the form of indicator functions, which is the true in the case of classifiers, the second term in the r.h.s. of equation (8) is,

$$\frac{\varepsilon(l)}{2}\left(1 + \sqrt{1 + \frac{4R_{emp}(\alpha_l)}{\varepsilon(l)}}\right) \qquad (9)$$

where α_l is the parameter set that defines the learning machine and $\varepsilon(l)$ is the measure of the difference between the expected and empirical risk (Vapnik, 1998). The error term, $\varepsilon(l)$, can be written in terms of the VC dimension and the size of the training set as,

$$\varepsilon(l) = 4\frac{h(\log(2l/h + 1)) - \log\eta/4}{l}, \qquad (10)$$

where h is the VC dimension (Vapnik, 1995).

Equation (10) provides us with a good method for comparing system configurations optimized using empirical risk minimization. When l/h is large, ε and $f(h)$ are both small. This implies that the expected risk tends towards the empirical risk. With this, we can guarantee both a small empirical risk (training error) and good generalization — an ideal situation for a learning machine. On the other hand, when l/h is small, both ε and $f(h)$ are large. Under this condition, a small empirical risk does not guarantee a small expected risk. Thus, the system is not guaranteed to generalize well. In this case, both terms on the r.h.s. of (8) need to be minimized simultaneously. From (10), we see that the error term, ε, monotonically increases with the VC dimension. This implies that the confidence in the empirical risk decreases monotonically with the VC dimension as does the generalization ability. These observations are depicted in Figure 3.

The principle of structural risk minimization (SRM) (Vapnik, 1995; Vapnik, 1998) is an attempt to identify the optimal point on the curve describing the bound on the expected risk. “The SRM principle defines a trade-off between the quality of the approximation of the given data and the complexity of the approximating function.” (Vapnik, 1995). From the above discussion, making the VC dimension a controlling variable for the generalization ability of the learning machine is a natural choice. In practice the principle of SRM can be implemented in two distinct ways:

1. for a fixed confidence interval optimize the empirical risk

2. for a fixed empirical risk optimize (or minimize) the confidence interval

The appropriate scheme to use is problem/classifier dependent.

Neural network learning uses the former procedure by first fixing the network structure and then minimizing the empirical risk using gradient descent. SVMs implement 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. In other words, SRM is an extension of ERM with the additional constraint that a *structure* be added to the space containing the optimal function. For example, structure can be imposed on the problem of function estimation using neural networks by associating the number of hidden units or their connections to each subset. In the case of optimal hyperplane classifiers, which will be discussed in the next section, structure is imposed by the width of the margin of the separating hyperplane.

2.2. Support Vector Classifiers

The following formulation is based on the fact that among all hyperplanes separating the data, there exists a unique hyperplane that maximizes the margin of separation between the classes (Vapnik, 1995). Figure 3 shows a typical 2-class classification example where the examples are perfectly separable using a linear decision region. H_1 and H_2 define two hyperplanes. The distance separating these hyperplanes is called the *margin*. The closest in-class and out-of-class examples lie on these two hyperplanes. As noted earlier, the SRM principle imposes structure to the optimization process by ordering the hyperplanes based on this margin. The optimal hyperplane is the one that maximizes the margin while minimizing the empirical risk. Figure 3 illustrates the difference between using ERM and SRM to estimate a simple hyperplane classifier. Using SRM results in the optimal hyperplane classifier.

Let $\boldsymbol{w}$ be a vector that is normal to the decision region. Let the l training examples be represented as the tuples $\{\boldsymbol{x}_i, y_i\}, i = 1, \ldots, l$ where $y = \pm 1$. The points that lie on the hyperplane separating the data satisfy

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0 \qquad (11)$$

where b is the distance of the hyperplane from the origin. Let the “margin” of the SVM be defined as the distance from the separating hyperplane to the closest positive and negative examples. The SVM training paradigm finds the separating hyperplane which gives the maximum margin. Once the hyperplane is obtained, all the training examples satisfy the following inequalities.

$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \geq +1 \qquad \text{for } y_i = +1 \qquad (12)$$

$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \leq -1 \qquad \text{for } y_i = -1 . \qquad (13)$$

The above equations can be compactly represented as a single inequality,

$$y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1 \geq 0 \qquad \forall i . \qquad (14)$$

Looking at the above equations with respect to Figure 4, we see that all points satisfying the equality condition in (12) lie on H_1. Similarly, all points satisfying the equality condition (13) lie on H_2. The distance between H_1 and H_2, also called the margin, is therefore two units. Since the normal to the hyperplane is not constrained to be of unit-norm, we need to normalize this margin by the norm of the normal vector to the hyperplane, $\boldsymbol{w}$. Therefore the margin as defined by the hyperplane is $2/\|\boldsymbol{w}\|$. For a completely separable data set, no points fall between H_1 and H_2. To maximize the margin we need to therefore maximize $1/\|\boldsymbol{w}\|$. Elegant techniques exist to optimize convex functions with constraints (Gill, Murray & Wright, 1981) so we choose to minimize $\|\boldsymbol{w}\|^2$, a convex function, instead. The training points for which the equality in (14) holds are called *support vectors*.

The theory of Lagrange multipliers (Gill, Murray & Wright, 1981) can be used to solve optimization problems involving convex functionals with constraints. The functional for the optimization problem in this discussion, called the Lagrangian, can be written as,

$$L_P = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \alpha_i y_i (\boldsymbol{x}_i \cdot \boldsymbol{w} + b) + \sum_{i=1}^{N} \alpha_i \tag{15}$$

The above is called the *primal* formulation of the optimization problem. Since we are minimizing L_P, its gradient with respect to $\boldsymbol{w}$ and b should be equal to zero. This gives the system of equations,

$$\boldsymbol{w} = \sum_j \alpha_j y_j x_j \text{, and} \tag{16}$$

$$\sum_i \alpha_i y_i = 0 . \tag{17}$$

Equations (11) and (16) imply that the decision function can be defined as,

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i \cdot \boldsymbol{x} + b \tag{18}$$

where the sign of f can be used to classify examples as either in-class or out-of-class. The above equation defines the SVM classifier. This definition of the classifier is worth a closer look. The classifier is defined in terms of the training examples. However all training examples do not contribute to the definition of the classifier. Only those with non-zero multipliers, the support vectors, define the classifier. In practice, the proportion of support vectors is small, making the classifier sparse. In other words, the data set itself defines how complex the classifier needs to be. This is in stark contrast to systems such as neural networks and HMMs where the complexity of the system is typically predefined or chosen through a cross-validation process. Since there are M dot products involved in the definition of the classifier, where M is the number of support vectors, the classification task scales linearly with the number of support vectors.

2.3. Solving Nonlinear, Non-separable Decision Problems

Thus far we have seen the case where the training data is completely separable using a linear margin. However, we know that this is not the case with most real-world data. Classification problems typically involve data which is non-linearly separable or completely non-separable. Given such a training set, we still need to estimate the classifier that maximizes the margin and minimizes the errors on the training set. Two modifications to the linear SVM classifier make this possible.

Optimization of non-separable data is typically accomplished by the use of soft decision classifiers where the classification of an example is tagged with a probability. However, in the case of optimal margin classifiers, we use the concept of *slack variables* to find the optimal solution. The optimal-margin classifier can be extended to this non-separable case by using a set of slack variables that account for training errors. In this situation, the inequality constraints to be satisfied by the hyperplane become,

$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \geq +1 - \xi_i \qquad \text{for } y_i = +1 , \tag{19}$$

$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \leq -1 + \xi_i \qquad \text{for } y_i = -1 \text{, and} \tag{20}$$

$$\xi_i \geq 0 \qquad \forall i , \tag{21}$$

where ξ ‘s are the slack variables. With these slack variables comes the need to estimate a trade-off parameter, C, which is the penalty incurred by the optimizer for accepting a training error. The higher the value of C, the harder the optimization process will try to minimize training errors. However this could mean increased time for convergence and in some cases, a larger support vector set. C is typically chosen using a cross-validation procedure.

The power of SVMs lies in transforming data to a high dimensional space and constructing a linear binary classifier via the optimization described earlier in this high dimensional space. Figure 3 illustrates this idea using a simple example. The two classes can be separated by a decision region in the form of circle which cannot be modeled by classifiers based on PCA or LDA (Vapnik, 1995). The data in this 2-dimensional space is transformed to a 3-dimensional space via the transformation,

$$(x, y) \Rightarrow (x^2, y^2, \sqrt{2}xy)\,. \tag{22}$$

From the figure it is clear that the two classes can be separated in the transformed space by defining a hyperplane passing through the points corresponding to the circular decision region.

In all previous formulations of the optimization process, observe that the only place the data points occur in (15) is as a dot product. We can define a transformation, Φ, such that $\Phi : \Re^n \to \Re^N$ where N is the dimensionality of the new feature space. In this new space we can still construct optimal margin classifiers with the only difference being that the simple dot product defined by substituting (16) into (15) will now have to be replaced by $\Phi(\boldsymbol{x_i}) \cdot \Phi(\boldsymbol{x_j})$. It would be advantageous if we could define a *kernel* function K,

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \Phi(\boldsymbol{x_i}) \cdot \Phi(\boldsymbol{x_j})\,, \tag{23}$$

that could compute this dot product directly without knowing the explicit form of Φ. In this new formulation, the decision function will take the form,

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x_i}) + b\,. \tag{24}$$

A function needs to satisfy certain conditions for it to be a valid kernel. The most important property the function must exhibit is that it needs to be a dot product in some feature space. In order to affirm that a function does indeed represent a dot product in a higher dimensional space, Mercer’s theorem can be used (Vapnik, 1995). Some commonly used kernels include

$$K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y} + 1)^d \qquad \text{(polynomial)} \tag{25}$$

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\{-\Upsilon|\boldsymbol{x} - \boldsymbol{y}|^2\} \qquad \text{(RBF).} \tag{26}$$

3. APPLYING SVMS TO SPEECH RECOGNITION

Hybrid approaches for speech recognition have been in use for several years now with connectionist speech recognition systems performing comparable to traditional speech recognition systems. Hybrid approaches provide a flexible paradigm to evaluate new acoustic modeling techniques. In connectionist systems neural networks replace Gaussians as the probabilistic model at the core of an acoustic model. We are not able to eliminate the HMM framework entirely because most new classification models, including SVMs and MLPs, cannot model the temporal structure of speech effectively. Most contemporary connectionist systems use the neural networks only to estimate posterior probabilities and use the HMM structure to model temporal evolution (Ström, 1997; Cook & Robinson, 1997; Tebelskis, 1995). We develop a similar hybrid SVM/HMM framework with the HMM structure being used to handle the temporal evolution of speech and SVMs being used to discriminatively classify frames of speech. The end result is a first successful application of SVMs to continuous speech recognition (Ganapathiraju, 2001; Ganapathiraju & Picone, 2000; Ganapathiraju, Hamaker & Picone, 1998; Ganapathiraju, Hamaker & Picone, 2000a; Ganapathiraju, Hamaker & Picone, 2000b).

SVMs have been applied successfully on several kinds of classification problems and have consistently performed better than other non-linear classifiers like neural networks and mixtures of Gaussians (Robinson, 1989; Joachims, 1999). The data set that propelled SVMs to prominence in the early 90's was the US Postal Service digit data on which the SVMs achieved the best numbers reported (LeCun et al., 1990). The development of efficient optimization schemes led to the use of SVMs for classification of larger tasks like text-categorization (Joachims, 1997; Joachims, 1999). There were some initial efforts to apply SVMs to speaker recognition in the early 90's (Schmidt & Gish, 1996). This effort had limited success because of the lack of efficient implementations of the SVM estimation process at that time. SVMs have also been applied to simple phone classification tasks and the results have been very encouraging (Ganapathiraju et al., 2000a).

All the above classification tasks have one common feature — they represent static classification tasks. SVMs are not designed to directly handle data with temporal structure. We need to address this problem in order to harness the advantages of SVMs for speech recognition. A second issue is that an SVM provides a binary decision. We would prefer a posterior probability which would capture our uncertainty in the classification. This problem is particularly important in speech recognition where there is a large degree of overlap in the feature space.

3.1. Posterior Estimation

Equation (2) describes the goal of the decoder in a speech recognition system which is to find the most likely word sequence. Depending on the acoustic models that constitute the word, this can be interpreted as finding the most likely model sequence.

$$\hat{M} = \underset{M}{\operatorname{argmax}}\ p(A/M)p(M) \tag{27}$$

where M is the acoustic model and A is the acoustic data. With HMMs, the class conditional probability $p(A/M)$ is obtained from the Gaussian evaluations. In connectionist systems, the neural networks estimate the posteriors, $p(M/A)$, and these posteriors are converted to likelihoods using the Bayes rule as,

$$P(A/M) = \frac{P(M/A)P(A)}{P(M)}. \tag{28}$$

In (28), since the classifier estimates the posteriors directly, if we assume the effect of $P(A)$ to be insignificant for recognition, simply dividing the posterior with the *a priori* probability of the model M gives the required conditional probability (scaled likelihood) $P(A/M)$ that can be used for recognition (Tebelskis, 1995).

SVMs, by definition, provide a distance or discriminant which can be used to compare classifiers and arrive at a classification as described by (24). This is unlike neural networks whose output activations are in fact estimates of the posterior class probabilities (Bridle, 1989). One of the main concerns in using SVMs for speech recognition is that there is no clear relationship between the distance of the test pattern from the margin and the posterior class probability. If we can develop such a relationship then applying SVMs to speech recognition within the HMM framework is possible.

A crude estimate of the posterior probability can be obtained by modeling the posterior class probability distribution with a Gaussian. Another possibility is to use a simple histogram approach. The above methods, however, are not Bayesian in nature in that they do not account for the variability in the estimates of the SVM parameters. Recent work on using moderated SVM outputs as estimates of the posterior probability has had success at the expense of increased computations (Kwok, 1999). We briefly discuss some of the methods that can be used to convert SVM distances to posterior probabilities.

The first option at hand is to use the SVM output directly:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x_i}) + b. \tag{29}$$

When $f > 0$, the test sample is classified as being in-class and when $f < 0$ the sample is classified as being out-of-class. In general, the value of f does not give any meaningful information. A second ad hoc method is to clip the SVM output at ± 1. This is a better approximation of the posterior. However, this output will show high confidence even in areas with low data density and far from the decision boundary. Unmoderated probability estimates based on maximum likelihood fitting is another option and is the one pursued in this work. In this method, we estimate a sigmoid defined as,

$$p(y = 1|f) = \frac{1}{1 + \exp(Af + B)} \tag{30}$$

Kwok’s definitive work on moderated SVM outputs addresses the above issues in greater detail (Kwok, 1999).

Figure 6 shows the distribution of the distances for positive and negative examples using a typical classifier. One possibility is to model these distance distributions using Gaussians and then compute the probability of the class given the SVM distance. Mathematically, that can be written as,

$$P(y = 1|f) = \frac{P(f|y = 1)P_1}{P(f|y = 1)P_1 + P(f|y = -1)P_{-1}}, \tag{31}$$

where f is the SVM distance and y is the class label which takes the value ± 1. Each of the class conditionals, $P(f|y)$ can be modeled as a Gaussian. Some simplifying assumptions can be made at this point without loss of generality.

Suppose we model each of the class-conditional probabilities with a Gaussian. Then,

$$P(f|y=1) = \frac{1}{\sqrt{2\pi\sigma_1^2}}\exp\frac{-(f-\mu_1)^2}{2\sigma_1^2} \text{ and} \tag{32}$$

$$P(f|y=-1) = \frac{1}{\sqrt{2\pi\sigma_{-1}^2}}\exp\frac{-(f-\mu_{-1})^2}{2\sigma_{-1}^2}. \tag{33}$$

If we assume the Gaussians to be of equal variance, which would typically be the case when there are enough positive and negative examples for a class, we can write the posterior probability in (31) as,

$$p(y=1|f) = \frac{1}{1+\frac{P_{-1}}{P_1}\exp\left(-\frac{1}{2\sigma^2}((f-\mu_1)^2-(f-\mu_{-1})^2)\right)}. \tag{34}$$

Expanding the quadratic terms in the denominator and simplifying gives

$$p(y=1|f) = \frac{1}{1+K\exp\left(-\frac{1}{2\sigma^2}((\mu_1^2-\mu_{-1}^2)+2(\mu_{-1}-\mu_1)f)\right)} , \tag{35}$$

which has the form of a sigmoid function

$$p(y=1|f) = \frac{1}{1+\exp(Af+B)}. \tag{36}$$

In (36), A and B are parameters that need to be estimated (using any suitable nonlinear function estimator). Note that we have assumed that the prior class probabilities are equal. An issue that arises from this formulation of estimating posteriors is that the distance estimates are heavily biased on the training data. In order to avoid biased estimates, a cross-validation set must be used to estimate the parameters of the sigmoid (Platt, 1999). The size of this data set can be determined based on the amount of training data that is available for the classifier. Figure 7 shows the posteriors and the estimated sigmoid for a typical classifier.

3.2. Classifier Design

Thus far we have not addressed a fundamental issue in classifier design — should the classifiers be one-vs-one or one-vs-all? As the name suggests, one-vs-one classifiers learn to discriminate one class from another class and one-vs-all classifiers learn to discriminate one class from all other classes. One-vs-one classifiers are typically smaller and can be estimated using fewer resources than one-vs-all classifiers. When the number of classes is N we need to estimate $N(N-1)/2$ one-vs-one classifiers as compared to N one-vs-all classifiers. On several standard classification tasks it has been proven that one-vs-one classifiers are marginally more accurate than one-vs-all classifiers (Allwein, Schapire & Singer, 2000; Weston & Watkins, 1999). Nevertheless, for computational efficiency, we chose to use one-vs-all classifiers in all experiments reported here.

3.3. Segmental Modeling

A logical step in building a hybrid system would be to replace the Bayes classifier in a traditional HMM system with an SVM classifier at the frame-level. However, the amount of training data and the confusion inherent in frame-level acoustic feature vectors is an issue worth addressing. Consider training a classifier to discriminate the phone ‘s’ from all other phones in a training set consisting of 40 hours of speech. At a frame rate of 100 frames per second, $14.4x10^6$ frames of data is available for each classifier to train on. Though very efficient optimizers are used to train the SVM, the amount of data could easily make the training process consume inordinate amount of computational resources (on the order of months even on extremely fast processors). Another aspect of using frame-level data to train SVMs is the implicit assumption that the frame-level alignments that the HMM system generates are reliable. Experiments clearly indicate this to be a flawed assumption for conversational speech corpora such as SWITCHBOARD (Godfrey, Holliman & McDaniel, 1992). An iterative training procedure where the alignments are gradually improved is an option but is not addressed in this work (Franzini, Lee & Waibel, 1990; Haffner, Franzini & Waibel, 1991; Bourlard & Morgan, 1998).

Apart from the above implementation issues that motivate looking at the data at a coarser level, there is clear evidence that it is extremely difficult to model human speech at the frame level where suprasegmental evidence such as duration cannot be used (Russell & Moore, 1985; Holmes, 1997; Russell & Holmes, 1997). Segment-based approaches to modeling speech have been pursued in the past (Ostendorf & Roukos, 1989; Austin et al., 1992). The motivation for most segment-based approaches is that the acoustic model needs to capture both temporal and spectral structure of speech which is clearly missing in frame-level classification schemes. Segmental approaches also overcome the assumption of conditional independence between frames of data in traditional HMM systems. Segmental data takes better advantage of the correlation in adjacent frames of data that is inherent in speech.

Despite their potential advantages, segment-based approaches have had limited success in the past. The inability to automatically generate reliable segment information is a primary problem. This is often circumvented through the use of a hybrid architecture for acoustic modeling. The HMM paradigm provides an elegant framework to generate the most likely segmentations via a dynamic programming approach. The new classifier can then postprocess these segmentations to hypothesize the best word sequence.

Once the segmentation problem is overcome, the next problem we face is the variable length or duration problem. Since segment duration is an important speech-related feature which is correlated with the word choice, speaking rate, etc., our classifier cannot simply discard this information. A simple but effective approach motivated by the 3-state HMMs used in most state-of-the-art speech recognition systems is to assume that the segments (phones in most cases) are composed of a fixed number of sections (Chang & Glass, 1997; Ström et al., 1999; Halberstadt, 1998). The first and third sections model the transition into and out of the segment, while the second section models the stable portion of the segment. We use segments composed of three sections in all recognition experiments reported in this work.

Figure 8 demonstrates the construction of a composite vector for a phone segment. SVM classifiers in our hybrid system operate on such composite vectors. The composite segment feature vectors are generated based on the alignments from a baseline 3-state mixture Gaussian HMM system. The length of the composite vector is dependent on the number of sections in each segment and the dimensionality of the frame-level feature vectors. For example, with a 39-dimensional feature vector at the frame level and 3 sections per segment, the composite vector has a dimensionality of 117. The SVM classifiers are trained on these composite vectors and recognition using the hybrid system is also performed using these segment-level composite vectors.

3.4. N-best List Rescoring Paradigm

As a first step towards building a complex hybrid SVM/HMM system, we have explored a simple rescoring paradigm instead of an integrated approach often used in hybrid connectionist systems (Bourlard & Morgan, 1998). Assuming that we have already trained the SVM classifiers for each phone in the model inventory, we generate N-best lists using a conventional HMM system. A model-level alignment for each hypothesis in the N-best list is then generated using the HMM system. Segment-level feature vectors are generated from these alignments. These segments are then classified using the SVMs. Posterior probabilities are computed using the sigmoid approximation discussed in the previous section. These probabilities are used to compute the utterance likelihood of each hypothesis in the N-best list. The N-best list is reordered based on the likelihood and the top hypothesis is used to calibrate the performance of the system. This scheme is shown in Figure 9.

4. EXPERIMENTAL RESULTS

SVMs have had significant success in several classification tasks (Robinson, 1989; Joachims, 1999). Most of these tasks have involved static data. However, speech recognition involves processing a temporally evolving signal. In order to gain insight into the effectiveness of SVMs for speech recognition, we explored two tasks. We first experimented on the Deterding static vowel classification task which is a common benchmark used for new classifiers. Second, we applied the techniques described above to a complete small vocabulary recognition task. With both tasks, we were able to achieve state-of-the-art results.

4.1. Deterding Vowels

In our first pilot experiment with speech data, we applied SVMs to a publicly available vowel classification task, Deterding Vowels (Deterding, Niranjan & Robinson, 2000). This was a good data set to evaluate the efficacy of static 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 consisted of 528 frames from eight speakers and the test set consisted of 462 frames from a different set of seven speakers. The speech data consisted of 11 vowels uttered by each speaker in a h*d context. Table I shows the vowel set and the corresponding words.

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. In Table II, we present results for a range of experiments using RBF and polynomial kernels with various parameter settings. Performance using both the kernels is better than most nonlinear classification schemes (Ström, 1997). The best performance we report is 35%. This is worse than the best performance reported on this data set (29% using a speaker adaptation scheme called Separable Mixture Models (Tenenbaum & Freeman, 1997). However, it is significantly better than the best neural network classifiers (Gaussian Node Network) that produce a misclassification rate of 44% (Robinson, 1989).

4.2. OGI Alphadigits

The performance of SVMs on the static classification of vowel data gave us good reason to expect the performance on continuous speech would be appreciably better than typical methods. Our initial tests of this hypothesis have been on a telephone alphadigit task. Recent work on both alphabet and alphadigit systems has taken a focus on resolving the high rates of recognizer confusion for certain word sets. In particular, the E-set (B, C, D, E, G, P, T, V, Z, THREE) and A-set (A, J, K, H, EIGHT). The problems occur mainly because the acoustic differences between the letters of the sets are minimal. For instance, the letters B and D differ primarily in the first 10-20 ms during the consonant portion of the letter (Loizou & Spanias, 1996).

The OGI Alphadigit Corpus (Cole, 1998) is a telephone database collected from approximately 3000 subjects. Each subject was a volunteer responding to a posting on the USEnet. The subjects were given a list of either 19 or 29 alphanumeric strings to speak. The strings in the lists were each six words long, and each list was “set up to balance phonetic context between all letter and digit pairs.” (Cole, 1998). There were 1102 separate prompting strings which gave a balanced coverage of vocabulary and contexts. The training, cross-validation and test sets consisted of 51544, 13926 and 3329 utterances respectively, each balanced for gender. The data sets have been chosen to make them speaker independent.

The baseline HMM system was created using our public-domain ASR system (Deshmukh, Ganapathiraju & Picone, 1999). This baseline system consists of cross-word context-dependent triphone models trained on 39-dimensional feature vectors. Each feature vector is comprised of 12 cepstral coefficients, energy, delta and acceleration coefficients. The triphones in the system are modeled by a three-state, left-to-right HMM with a mixture Gaussian emission distribution in each state. The emission distributions were trained up to 8 mixtures and a decision tree phonetic clustering procedure was used to reduce the overall parameter count in the system. A Viterbi-style search was performed over a loop grammar (any sequence of lexemes is possible) to find the final baseline hypothesis for each test utterance. This system achieved a word error rate (WER) of 11.9%

The baseline HMM system was then used to generate the segmental training data for the SVM models by Viterbi-aligning the training reference transcription to the acoustic data. The time marks derived from this Viterbi alignment were used to extract the segments. Before extraction, each feature dimension was normalized to the range [-1,1] to account for unequal variance amongst the feature dimensions. For each phoneme in the training transcription, a segment was extracted. This segment was divided into three parts as detailed previously in Figure 8. An additional parameter giving the log of the segment duration was added to yield a composite vector of size 3 * 39 features + 1 log duration = 118 features.

For each phoneme in the lexical set, shown in Table III, an SVM model was trained to discriminate between that phoneme and all other phonemes (one-vs-all models) giving a total of 29 models. From the segmental data extracted above, we chose a training set for each model. This training set consisted of equal amounts of within-class and out-of-class data. All within-class data available for that phoneme was used. The out-of-class data was randomly chosen such that one half of the out-of-class data came from phonemes that were phonetically similar to the phoneme of interest and one half came from all other phonemes. The phonetic similarity sets used are shown in Table IV. Balancing the data according to similarity allowed for more data to be used in training the SVM to learn the most confusable discriminant regions. Once the training sets were chosen, the SVMLight (Joachims, 1999) utilities were used to train each of the 29 phoneme SVM models.

The trained SVM models are only capable of making hard binary decisions whereas we require probabilistic likelihoods during recognition. Thus, we must estimate a posterior distribution from the SVM model as described in section 3.1. Estimating the sigmoid parameters from the training data would lead to severe overfitting so we use the cross-validation data instead. As before, segmental features are extracted from the cross-validation data and the SVM models are used to generate distance scores for the cross-validation set. Histograms are generated for the classifier distances and are used to estimate the parameters of a sigmoid using nonlinear optimization techniques.

In our rescoring paradigm, N-best lists are generated by the HMM system and fed along with the segmentations for each list entry to the SVM system for reordering. A problem that arises is determining how to get the segmentations to feed to the SVM system. During training and cross-validation, we were able to force-align to the reference transcription, but during testing there is no reference transcription available. We have first used the 1-best output hypothesis of the HMM baseline system as a *pseudo-reference* for segmentation. Second we experimented with using an alignment corresponding to each of the N-best paths (i.e. we end up with N segmentations for each utterance). In practice, we have found that the 1-best hypothesis segmentation performs better than the N-best segmentation.

Table V shows the performance of the hybrid SVM system as a function of the kernel parameters. These results were generated with 10-best lists whose total list error (the error inherent in the lists themselves) was 4.0%. As with the vowel classification data, the RBF kernel performance was superior to the polynomial kernel. Also like the vowel classification task, the generalization performance in the form of error rates is fairly flat for a wide range of kernel parameters. The 1-best hypothesis segmentation was used to produce a best result of 11.0% WER using an RBF kernel. To provide an equivalent and fair comparison with the HMM system we have rescored the N-best lists with the baseline HMMs. The results for the baseline system error remain the same indicating that no search errors were made by the HMM system due to search space pruning and we can have confidence that the improvement is indeed due to the SVM classifier.

As a point of reference, we also produced results using a reference segmentation. These are a set of *oracle* experiments where the segmentations are produced by force-aligning the reference transcription. The results of these experiments provide a nice analysis tool as they give us a presumptive lower bound on the achievable error (the actual lower bound is the N-best list error rate, but it is a good assumption that we won’t do better than a system with perfect knowledge of the reference segmentation). It is important to notice that using the correct segmentation for the SVM model is critical in achieving good performance. However, when we try to let the SVM decide the best segmentation and hypothesis combination by using the N-best segmentations the performance gets worse. This apparent anomaly indicates the need to incorporate variations in segmentation into the classifier estimation process. Relaxing this strong interdependence between the segmentation and the SVM performance is a point for further research. While the segmentation is critical, Table VI shows that the proportions used for the segments are not.

The goal of SRM techniques is to build a classifier which balances generalization with performance on the training set. Table VII shows how the RBF kernel parameter is used as a tuning parameter for achieving this balance. As gamma increases, the variance of the RBF kernel decreases which produces a more narrow support region in the high-dimensional space. This requires a larger number of support vectors and leads to overfitting as shown when gamma is set to 5.0. As gamma decreases, the number of SVs decreases which leads towards a smoother decision surface in the high-dimensional space. This leads to oversmoothing as shown when gamma is set to 0.1. In accordance with Figure 3, the optimal is a clear trade-off between the two extremes of overfitting (poor generalization) and oversmoothing (poor performance).

Careful analysis of the error modalities in both the baseline system and the SVM hybrid system shows the two systems have somewhat different error modalities. It appears that there are classes for which SVMs do better than HMMs and there are classes which are better modeled by HMMs than SVMs. This is indicated in Table VIII. This led us to attempt a system combination scheme where the word-likelihood score from the SVM system was combined with the word-likelihood score from the HMM baseline according to

$$likelihood = \text{SVM score} + \frac{\text{HMM Score}}{\text{norm factor}}. \qquad (37)$$

As the normalization factor increases, the likelihood is dominated by the SVM hypothesis. Likewise, as the normalization factor decreases, the HMM score dominates. Table IX shows the results of this method using the N-best segmentations. We are able to effectively gain from the disparate strengths of the two models to achieve our best overall result of 10.6% WER. The last column of Table VIII shows, that this gain is achieved for every class of data explored in this recognition task. This fact is particularly encouraging and warrants further research.

The improvements provided by the hybrid SVM system are statistically significant and very promising. The oracle experiment shows that further improvements are possible using this paradigm if certain key issues are addressed. Primary among them is the need for better integration of segmentation information into the system using concepts such as segment-graphs (Chang & Glass, 1997; Chang, 1998). Also, as noted earlier, variation in segmentations needs to be incorporated into the training process for the classifiers. One way to achieve this is to iteratively train the classifiers by going through a estimate-classify process where classification errors can be fed back into the system.

5. CONCLUSIONS

Most speech recognition systems today are based on HMMs and a few are based on hybrid HMM-Neural

Network architectures. HMMs have had significant success since they offer an elegant mechanism to model both the acoustic variability and the temporal evolution of speech. The existence of efficient iterative parameter estimation procedures such as the expectation-maximization algorithm has a significant role in the success of HMMs in speech recognition. However, HMMs suffer from a number of drawbacks — the assumption of independence of successive frames and the idea that improved representation leads to better classification being key amongst them. Our approach addresses these issues by using a segmental approach and a discriminative estimation process.

This paper addresses the use of Support Vector Machines as a viable classifier in a continuous speech recognition system. The technology has been successfully applied to a small vocabulary task — OGI Alphadigits. A hybrid SVM/HMM system has been developed which uses SVMs to post-process data generated by a conventional HMM system. The hybrid system achieves a word error rate of 10.6% on a open-loop speaker-independent test set as compared to 11.9% achieved using a context-dependent multiple mixture HMM system. The results obtained in the experiments clearly indicate the classification power of SVMs and affirm the use of SVMs for acoustic modeling. The fact that the improvements are made on all classes of sounds (some being minimal pairs), indicates that the SVM classifiers are capable of classifying even extremely confusable data better than HMMs.

Several issues that arise as a result of the hybrid framework have been addressed including estimation of posterior probabilities and the use of segment-level data. The oracle experiment reported here clearly shows the potential of this hybrid system while highlighting the need for further research into the segmentation issue. Using oracle segmentations based on the reference transcription, the hybrid system performs at a word error rate of 7.0% as compared to the 10.6% we get when the system is run in a N-best rescoring mode. This shows the dependence of the SVM classifiers on good segmentations.

To alleviate this problem, further research into defining an iterative SVM classifier estimation process is required. This would allow the classifiers to learn variations in segmentation, thereby making them less dependent on segmentation accuracy. A more elegant approach to handling the segmentation issue is to use segment-graphs generated by the HMM system to rescore N-best lists. This approach has been very useful in segmental speech recognition systems developed over the past few years.

6. REFERENCES

[1] Ackley, D.A., Hinton, G.E., and Sejnowski, T.J. (1985). A Learning Algorithm for Boltzmann Machines. *Cognitive Science*, vol. 9, pp. 147-169.

[2] Allwein, E. Schapire, R.E. and Singer, Y. (2000). Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. *Journal of Machine Learning Research*, vol. 1(Dec), pp. 113-141.

[3] Austin, S., Zavaliagkos, G., Makhoul, J., and Schwartz, R. (1992). Speech Recognition Using Segmental Neural Nets. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 625-628.

[4] Bodenhausen, U., Manke, S. and Waibel, A. (1993). Connectionist Architectural Learning for High Performance Character and Speech Recognition. *Proceedings of the International Conference on Acoustics Speech and Signal Processing,* vol. 1, pp. 625-628, Minneapolis, MN, USA.

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

[6] Bourlard, H. and Morgan, N. (1998). Hybrid HMM/ANN Systems for Speech Recognition: Overview and New Research Directions. *Adaptive Processing of Sequences and Data Structures*, C. L. Giles and M. Gori (Eds.), vol. 1387 of Lecture Notes in Artificial Intelligence, pp. 389-417, Springer-Verlag, Berlin, Germany.

[7] Bridle, J.S. (1989). Probabilistic Interpretation of Feed forward Classification Network Outputs, with Relationship to Statistical Pattern Recognition. *Neuro-Computing: algorithms, architectures and applications*, Springer-Verlag, New York, USA.

[8] Bridle, J.S. and Dodd, L. (1991). An Alphanet Approach to Optimizing Input Transformations for Continuous Speech Recognition. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 277-280.

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

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

[11] Chang, J. (1998). *Near-Miss Modeling: A Segment-Based Approach to Speech Recognition*, Ph.D. Thesis, MIT Department of Electrical Engineering and Computer Science.

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

[13] Cook, G.D. and Robinson, A.J. (1997). The 1997 ABBOT System for the Transcription of Broadcast News. *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, USA..

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

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

[16] Deshmukh, N., Ganapathiraju, A. and Picone, J. (1999). Hierarchical Search for Large Vocabulary Conversational Speech Recognition. *IEEE Signal Processing Magazine*, vol. 1, no. 5, pp. 84-107.

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

[18] Franzini, M., Lee, K.F., and Waibel, A. (1990). Connectionist Viterbi Training: A New Hybrid Method for Continuous Speech Recognition. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing,* vol. 1, pp. 425-428, Albuquerque, NM, USA.

[19] Fritsch, J. (2000). *Hierarchical Connectionist Acoustic Modeling for Domain-Adaptive Large Vocabulary Speech Recognition*, Ph. D. Thesis, University of Karlsruhe, Germany.

[20] Ganapathiraju, A. (2002). *Support Vector Machines for Speech Recognition*, Ph. D. Thesis, Mississippi State University, Mississippi State, Mississippi, USA.

[21] Ganapathiraju, A. and Picone, J. (2000). Hybrid SVM/HMM architectures for speech recognition. *Technical Report, Institute for Signal and Information Processing*, Mississippi State University, Mississippi, USA.

[22] Ganapathiraju, A., Hamaker, J. and Picone, J. (1998). Support Vector Machines for Speech Recognition. *Proceedings of the International Conference on Spoken Language Processing*, pp. 2923-2926, Sydney, Australia.

[23] Ganapathiraju, A., Hamaker, J. and Picone, J. (2000a). A Hybrid ASR System Using Support Vector Machines. *International Conference of Spoken Language Processing*, Beijing, China.

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

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

[26] Godfrey, J., Holliman, E. and McDaniel, J. (1992). 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.

[27] Haffner, P., Franzini, M. and Waibel, A. (1991). Integrating Time Alignment and Neural Networks for High Performance Continuous Speech Recognition. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 105-108.

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

[29] Hamaker, J., Ganapathiraju, A. and Picone, J. (1998). A Proposal for a Standard Partitioning of the OGI AlphaDigit Corpus," available at *http://isip.msstate.edu/projects/lvcsr/recognition_task/alphadigits/data ogi_alphadigits/trans_eval.text*. Institute for Signal and Information Processing, Mississippi State University, Mississippi, USA.

[30] Holmes, W. (1997). *Modelling Segmental Variability for Automatic Speech Recognition*, Ph. D. Thesis, University of London, UK.

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

[32] Joachims, T. (1997). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Technical Report 23, LS VIII, University of Dortmund*, Germany.

[33] Joachims, T. (1999). *SVMLight: Support Vector Machine,* http://www-ai.informatik.uni-dortmund.de/

FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light.eng.html, University of Dortmund.

[34] Kirkpatrick, S., Gellatt, C.D. and Vecchi, M.P. (1983). Optimization by Simulated Annealing. *Science*, Vol. 220, pp. 671-680.

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

[36] Lawrence, S., Giles, C.L. and Tsoi, A.C. (1996). What size neural network gives optimal generalization. *Technical Report UMIACSTR-96-22,* Institute for Advanced Computer Studies, University of Maryland, USA.

[37] Lawrence, S., Giles, C.L, and Tsoi, A.C. (1997). Lessons in neural-network training: Overfitting may be harder than expected. *Proceedings of the 14th National Conference on Artificial Intelligence*, AAAI Press, pp. 540-545.

[38] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, R. and Jackel, L.D. (1990). , Handwritten Digit Recognition with Backpropagation Network. *Advances in Neural Information Processing Systems-2*, Touretzky, D. (editor), Morgan Kaufman, pp. 396-404.

[39] Loizou, P. and Spanias, A. (1996). High-Performance Alphabet Recognition. *IEEE Transactions on Speech and Audio Processing,* pp. 430-445.

[40] McDermott, E. (1997). *Discriminative Training for Speech Recognition,* Ph. D. Thesis, Waseda University, Japan.

[41] McLachlan, G. (1997). *The EM algorithm and extensions*, John Wiley, New York, NY, USA.

[42] 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.

[43] 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.

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

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

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

[47] Redner, R.A. and Walker, H.F. (1984). Mixture Densities, Maximum Likelihood and the EM Algorithm. *SIAM Review,* vol. 26, no. 2, pp. 195-239.

[48] Renals, S. (1990). *Speech and Neural Network Dynamics*, Ph. D. Thesis, University of Edinburgh, UK.

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

[50] Robinson, A.J. (1989). *Dynamic Error Propagation Networks*, Ph.D. Thesis, Cambridge University, UK.

[51] Rosenblatt, F. (1957). The Perceptron: A Perceiving and Recognizing Automaton. C*ornell Aeronautical Laboratory Report 85-460-1*.

[52] Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986). Learning Internal Representation by Error Propagation. *Parallel Distributed Processing: Explorations in The Microstructures of Cognition, Vol. 1: Foundations*, Rumelhart, D.E. and McClelland, J.L. (Editors), MIT Press, Cambridge, MA, pp. 318-362.

[53] Russell, M.J. and Moore, R.K. (1985). 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.

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

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

[56] Ström, N. (1997). Phoneme Probability Estimation with Dynamic Sparsely Connected Artificial Neural Networks. *The Free Speech Journal*, vol. 1, no. 5.

[57] Ström, N., Hetherington, L., Hazen, T.J., Sandness, E. and Glass, J. (1999). Acoustic Modeling Improvements in a Segment-Based Speech Recognizer. *Proceedings of IEEE ASRU Workshop*, Keystone, CO, USA.

[58] Tebelskis, J. (1995). S*peech Recognition using Neural Networks*, Ph. D. Thesis, Carnegie Mellon University, Pittsburgh, USA.

[59] Tenenbaum, J. and Freeman, W.T. (1997). Separating Style and Content. *Advances in Neural Information Processing Systems 9*, MIT Press, Cambridge, MA, USA.

[60] Vapnik, V.N. (1995). *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, NY, USA.

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

[62] Weston, J. and Watkins, C. (1999). Support Vector Machines for Multi-Class Pattern Recognition. *Proceedings of the Seventh European Symposium On Artificial Neural Networks*.

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

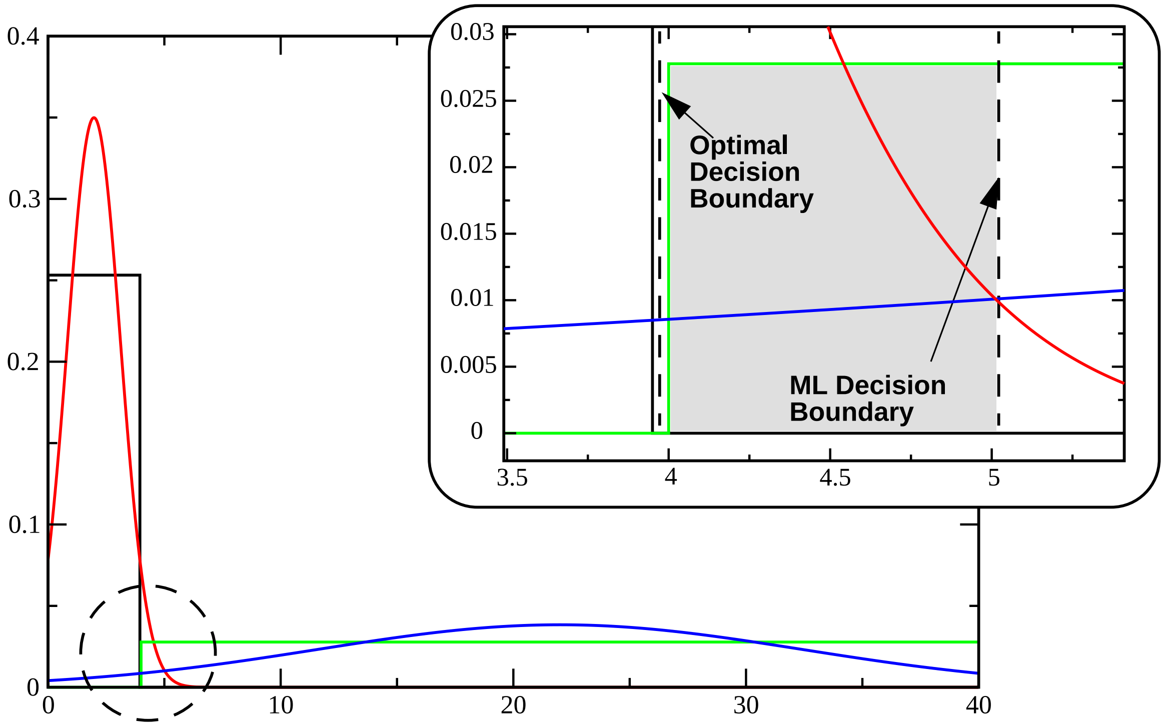


Figure 1. An example of a two-class problem where a maximum likelihood-derived decision surface is not the optimal (adapted from (McDermott, 1997)). In the exploded view, the shaded region indicates the error induced by modeling the separable data by Gaussians estimated using maximum likelihood. This case is common for data, such as speech, where there is overlap in the feature space or where class boundaries are adjacent.

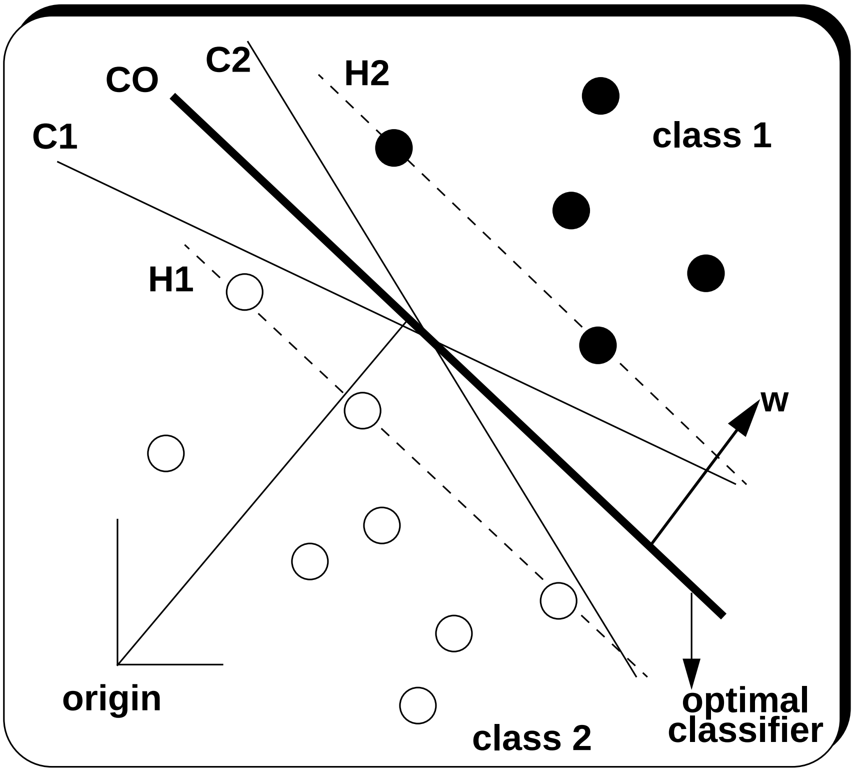


Figure 2. Difference between empirical risk minimization and structural risk minimization for a simple example involving a hyperplane classifier. Each hyperplane ($C0$, $C1$ and $C2$) achieves perfect classification and, hence, zero empirical risk. However, $C0$ is the optimal hyperplane because it maximizes the margin — the distance between the hyperplanes $H1$ and $H2$. Maximizing the margin indirectly results in better generalization.

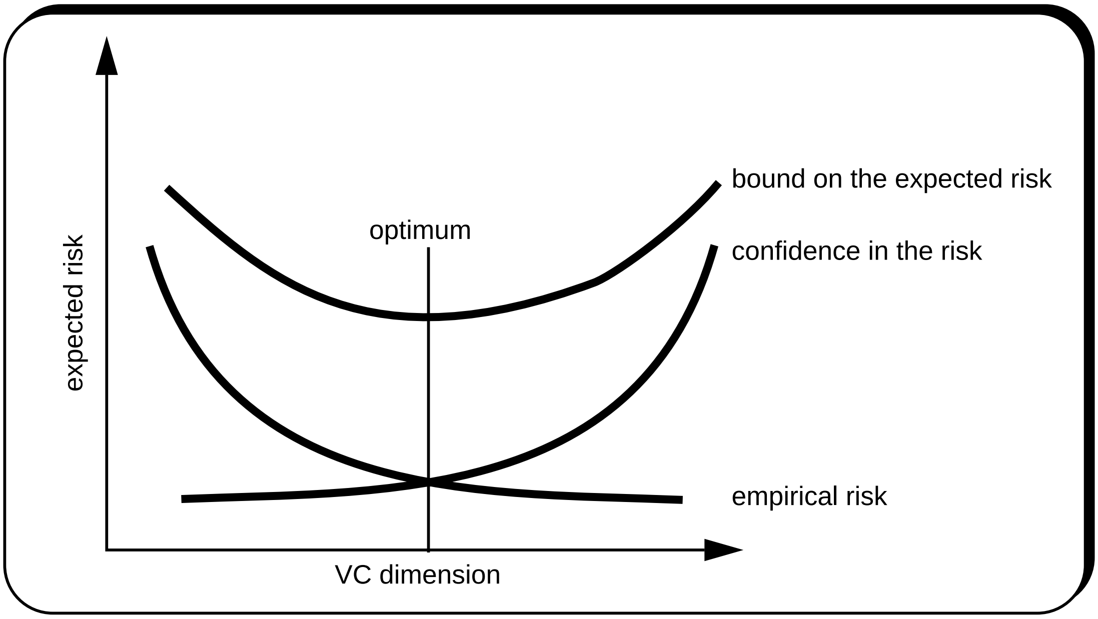


Figure 3. The goal in training a classification system is to minimize the expected risk. As the VC dimension of the classifier increases, our confidence in the generalization ability of the machine decreases. The learning machine becomes too complex and suffers from overfitting of the training set. This demonstrates the principle of Occam's razor where the simplest (lowest VC dimension) classifier is chosen that will attain a low empirical risk. The ability to automatically learn the optimal trade-off between these two factors is the most compelling feature of the SVM theory.

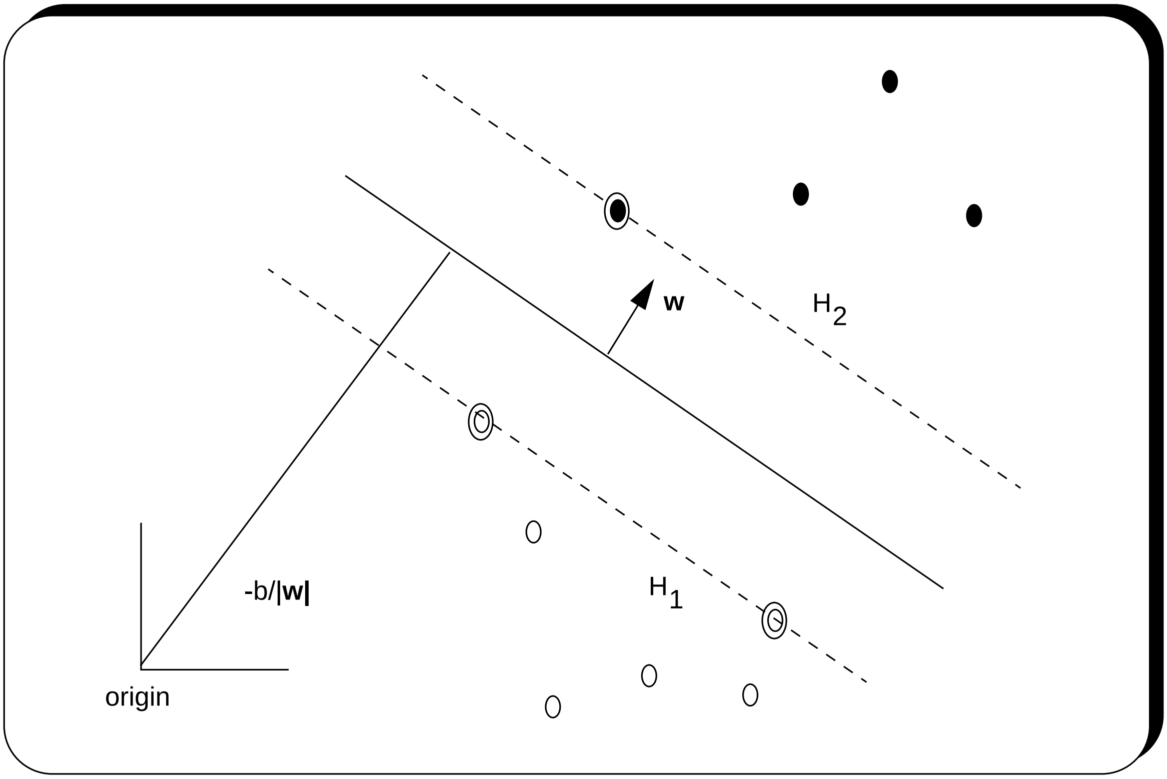
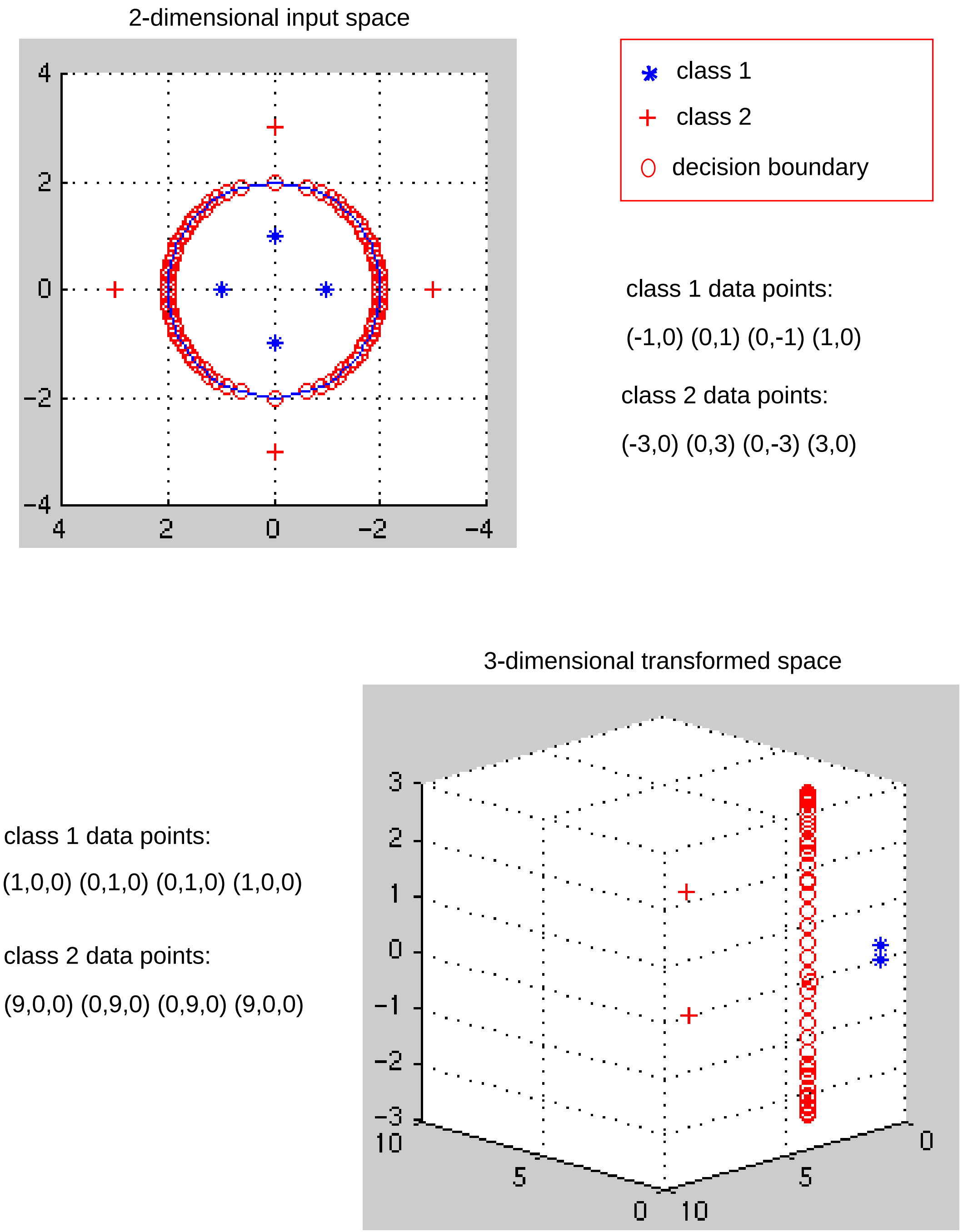


Figure 4. Definition of a linear hyperplane classifier. SVMs are constructed by maximizing the margin. The support vectors are shown as concentric circles.



class 1 data points:

(-1,0) (0,1) (0,-1) (1,0)

class 2 data points:

(-3,0) (0,3) (0,-3) (3,0)

class 1 data points:

(1,0,0) (0,1,0) (0,1,0) (1,0,0)

class 2 data points:

(9,0,0) (0,9,0) (0,9,0) (9,0,0)

Figure 5. An illustration of the fact that the construction of a simple hyperplane in a higher dimensional space is equivalent to a non-linear decision surface in a lower dimensional space. In this example a decision surface in the form of a circle in a 2-dimensional space is modeled as a hyperplane in a 3-dimensional space.

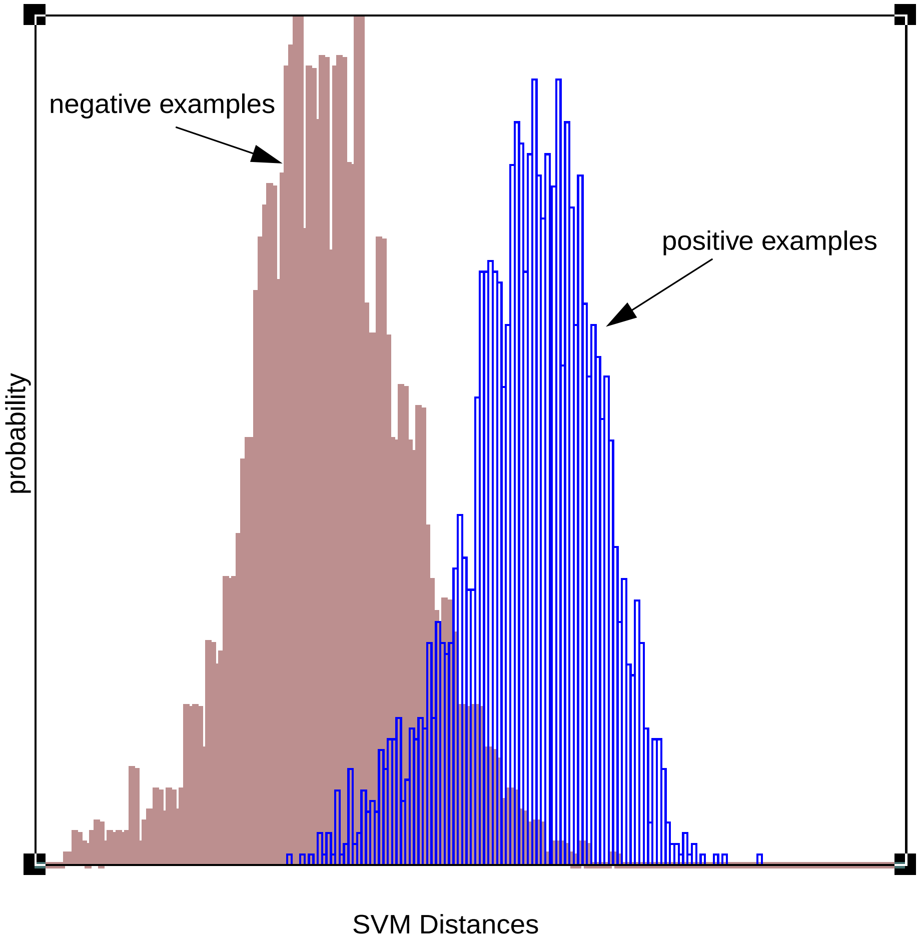


Figure 6. Histogram of SVM distances for positive and negative examples from the cross validation data.

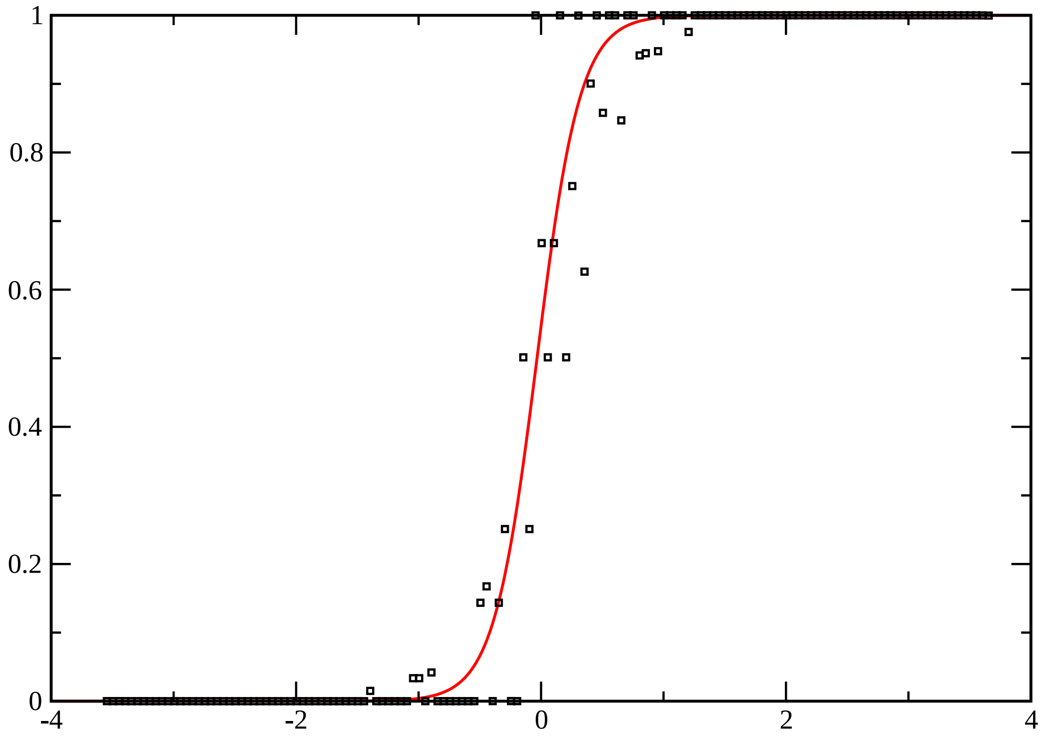


Figure 7. A sigmoid fit to the SVM distance-based posterior probability estimate.

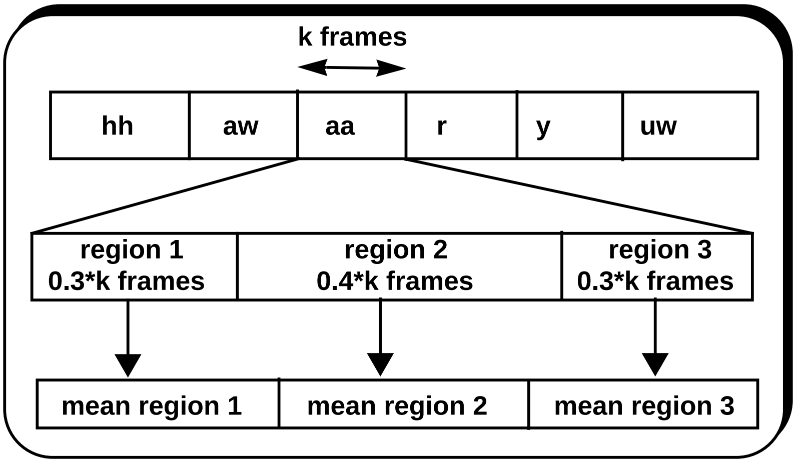


Figure 8. Composition of the segment level feature vector assuming a 3-4-3 proportion for the three sections.

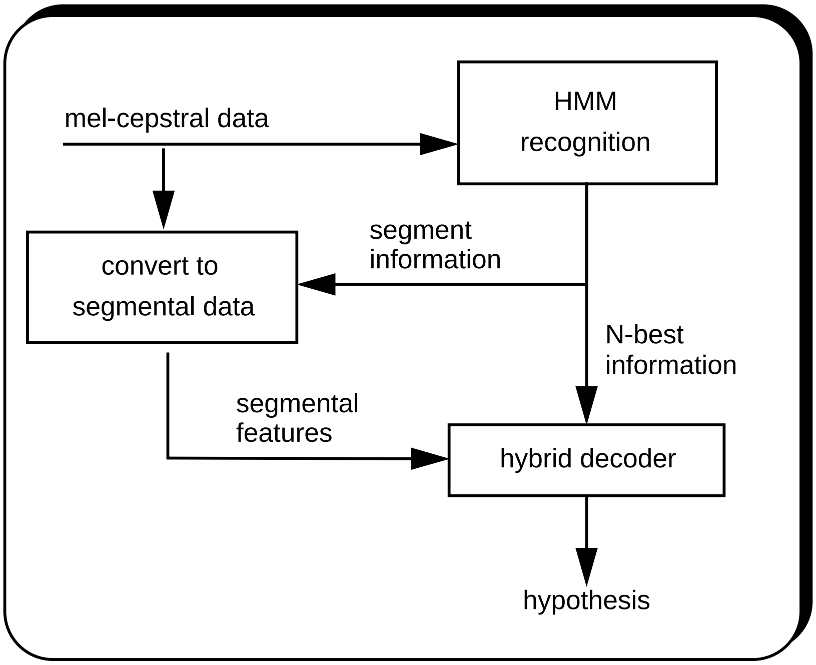


Figure 9. Flow graph for hybrid system using the N-best rescoring paradigm.

vowel	word	vowel	word
i	heed	O	hod
I	hid	C:	hoard
E	head	U	hood
A	had	u:	who’d
a:	hard	3:	heard
Y	hud		

Table I. The vowels and the corresponding words that were used in the Deterding Vowel database.

RBF gamma	Classification Error (%)	polynomial order	Classification Error(%)
0.2	45	2	49
0.3	39.6	3	52
0.4	35	4	52
0.5	35.5	5	52
0.6	35.5		
0.7	35		
0.8	36		
0.9	36.5		
1.0	37.8		

Table II. Comparison of vowel classification error rates as a function of the RBF kernel width (gamma) and the polynomial kernel order. Results are shown with the training error penalty, C, set to 10. For both kernels, there is a wide range of the kernel parameter for which the generalization capability of the SVM is equivalent. The RBF kernel performs much better than the polynomial kernel for this task.

phoneme	word	pronunciation
aa	R	aa r
ah	ONE	w ah n
ax	W	d ah b ax l y uw
ay	I	ay
b	B	b iy
ch	H	ey ch
d	D	d iy
eh	F	eh f
ey	A	ey
f	FOUR	f ow r
ih	SIX	s ih k s
iy	E	iy
jh	G	jh iy
k	K	k ey
l	L	eh l
m	M	eh m
n	N	eh n
ow	O	ow
p	P	p iy
r	FOUR	f ow r
s	SIX	s ih k s
t	TWO	t uw
th	THREE	th r iy
uw	TWO	t uw
v	FIVE	f ay v
w	ONE	w ah n
y	U	y uw
z	ZERO	z iy r ow
sil	[SILENCE]	sil

Table III. Phoneme set for alphadigit recognition. A total of 29 phoneme models were used.

set	phonemes
vowels	aa, ah, ax, ay, eh, ey, ih, iy, ow, uw
fricatives	ch, f, s, th, v, z
nasals	m, n
approximants	w, r, l, y
stops	b, d, jh, k, p, t

Table IV. Phonetic similarity sets used to build SVM training sets. This clustering is very coarse. One might be able to improve performance by making finer distinctions in the similarity classes.

RBF gamma	WER (%) Hypothesis Segmentation	WER (%) Reference Segmentation	polynomial order	WER (%) Hypothesis Segmentation	WER (%) Reference Segmentation
0.1	13.2	9.2	3	11.6	7.7
0.4	11.1	7.2	**4**	**11.4**	**7.6**
0.5	11.1	7.1	5	11.5	7.5
0.6	11.1	7.0	6	11.5	7.5
0.7	**11.0**	**7.0**	7	11.9	7.8
1.0	11.0	7.0			
5.0	12.7	8.1			

Table V. Comparison of word error rates as a function of the RBF kernel width (gamma) and the polynomial kernel order. Results are shown for a 3-4-3 segment proportion with the error penalty, C, set to 50. Both the 1-best hypothesis from the HMM system and the reference transcription have been used to generate segmentation information. The reference segmentation performance can be seen as an approximate lower-bound on the achievable error. The WER for the baseline HMM system is 11.9%.

Segmentation Proportions	WER (%) RBF kernel	WER (%) polynomial kernel
2-4-2	11.0	11.3
3-4-3	11.0	11.5
4-4-4	11.1	11.4

Table VI. Comparison of performance as a function of the segment proportions. 1-best hypothesis segmentations are used to generate the SVM segmentations and 10-best lists are rescored.

RBF gamma	WER (%) Hypothesis Segmentation	WER (%) Reference Segmentation	Average Number of Support Vectors
0.1	13.2	9.2	1313
0.4	11.1	7.2	3293
0.5	11.1	7.1	3972
0.6	11.1	7.0	4248
0.7	**11.0**	**7.0**	**4784**
1.0	11.0	7.0	6577
5.0	12.7	8.1	10236

Table VII. The number of support vectors is a good indicator of generalization ability. As the number of vectors decreases, the classifier tends toward an overly smooth decision surface, which leads to poor generalization. As the number of support vectors increases, the classifier tends toward overfitting; also leading to poor generalization.

Data Class	HMM (%WER)	SVM (%WER)	HMM+SVM (%WER)
a-set	13.5	11.5	11.1
e-set	23.1	22.4	20.6
digits	5.1	6.4	4.7
alphabets	15.1	14.3	13.3
nasals	12.1	12.9	12.0
plosives	22.6	21.0	18.9
Overall	**11.8**	**11.8**	**10.6**

Table VIII. The HMM system and the SVM system have different strengths. A combination of the two is capable of using the strengths of both systems to achieve a better overall performance. However, it does require the estimation of another nuisance parameter to normalize the respective scores. The SVM results reported in this table use the N-best segmentations. The combination of the two systems uses (37) with the normalization factor set to 200.

Normalization Factor	HMM+SVM (%WER)
100000	11.8
10000	11.4
1000	10.9
500	10.8
200	**10.6**
100	10.7
50	10.8
0	11.8

Table IX. Error rate as a function of the normalization factor. The optimal value is 200, and provides an error rate of 10.6%.