Applications of Support Vector Machines to Speech Recognition

Aravind Ganapathiraju, *Member, IEEE*, Jonathan E. Hamaker, *Member, IEEE*, and Joseph Picone, *Senior Member, IEEE*

***Abstract*—Recent work in machine learning has focused on models, such as the support vector machine (SVM), that automatically control generalization and parameterization as part of the overall optimization process. In this paper, we show that SVMs provide a significant improvement in performance on a static pattern classification task based on the Deterding vowel data. We also describe an application of SVMs to large vocabulary speech recognition and demonstrate an improvement in error rate on a continuous alphadigit task (OGI Alphadigits) and a large vocabulary conversational speech task (Switchboard). Issues related to the development and optimization of an SVM/HMM hybrid system are discussed.**

***Index Terms*—Machine learning, speech recognition, statistical modeling, support vector machines.**

I. Introduction

SPEECH recognition systems have become one of the premier applications for machine learning and pattern recognition technology. Modern speech recognition systems, including those described in this paper, use a statistical approach [1] based on Bayes' rule. The acoustic modeling components of a speech recognizer are based on hidden Markov models (HMMs) [1], [2]. The power of an HMM representation lies in its ability to model the temporal evolution of a signal via an underlying Markov process. The ability of an HMM to statistically model the acoustic and temporal variability in speech has been integral to its success. The probability distribution associated with each state in an HMM models the variability that occurs in speech across speakers or phonetic context. This distribution is typically a Gaussian mixture model (GMM) since a GMM provides a sufficiently general parsimonious parametric model as well as an efficient and robust mathematical framework for estimation and analysis.

Widespread use of HMMs for modeling speech can be attributed to the availability of efficient parameter estimation procedures [1], [2] that involve maximizing the likelihood (ML) of the data given the model. 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, although it does not guarantee finding the global maximum [3].

There are, however, problems with an ML formulation for applications such as speech recognition. A simple example, which is shown in Fig. 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 Bayes' rule is used to classify the data. We see that the decision threshold occurs inside the range of class 2. This results in a significant probability of error. 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, ML training of a Gaussian model will never achieve perfect classification. Learning decision regions discriminatively will improve classification performance. The important point here is not that Gaussian models are necessarily an incorrect choice, but rather that discriminative approaches are a key ingredient for creating robust and more accurate models. Many promising techniques [4], [5] have been introduced for using discriminative techniques to improve the estimation of HMM parameters.

Artificial neural networks (ANNs) represent an interesting and important class of discriminative techniques that have been successfully applied to speech recognition [6]–[8]. Although ANNs attempt to overcome many of the problems previously described, their shortcomings with respect to applications such as speech recognition are well documented [9], [10]. Some of the most notable deficiencies include design of optimal model topologies, slow convergence during training, and a tendency to overfit the data. However, it is important to note that many of the fundamental ideas presented in this paper (e.g., soft margin classifiers) have similar implementations within an ANN framework. In most classifiers, controlling a tradeoff between overfitting and good classification performance is vital to the success of the approach.

In this paper, we describe the application of one particular discriminative approach—support vector machines (SVMs) [11]—to speech recognition. We review the SVM approach in Section II, discuss applications to speech recognition in Section III, and present experimental results in Section IV.

Manuscript received July 16, 2003; revised March 18, 2004. This work was supported in part by the National Science Foundation under Grant IIS-0095940. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Andrew Singer.

A. Ganapathiraju is with Conversay, Redmond, WA 98052 USA (e-mail: aganapathiraju@conversay.com).

J. Hamaker is with Microsoft Corporation, Redmond, WA 98052-6399 USA (e-mail: jonham@microsoft.com).

J. Picone is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762 USA (e-mail: picone@isip.msstate.edu).

Digital Object Identifier 10.1109/TSP.2004.831018

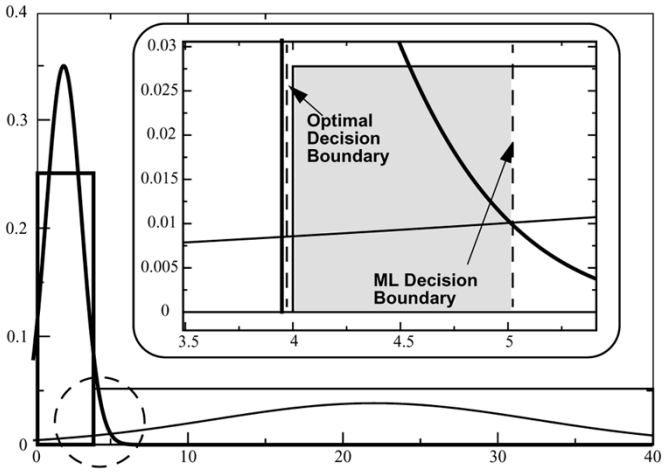

Fig. 1. Example of a two-class problem where a maximum likelihood-derived decision surface is not optimal (adapted from [4]). In the exploded view, the shaded region indicates the error induced by modeling the separable data by Gaussians estimated using maximum likelihood. This case is common for data, such as speech, where there is overlap in the feature space or where class boundaries are adjacent.

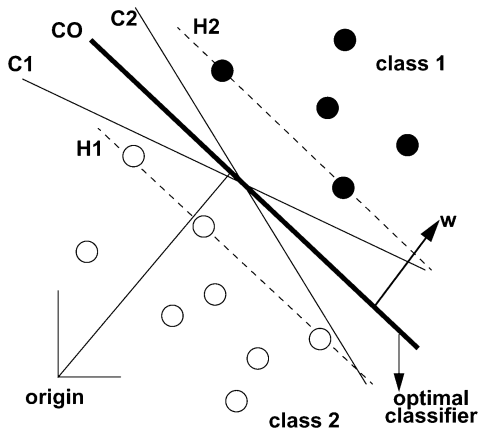

Fig. 2. Difference between empirical risk minimization and structural risk minimization for a simple example involving a hyperplane classifier. Each hyperplane achieves perfect classification and, hence, zero empirical risk. However, C0 is the optimal hyperplane because it maximizes the margin—the distance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

More comprehensive treatments of fundamental topics such as risk minimization and speech recognition applications can be found in [11]–[20].

II. Support Vector Classifiers

An SVM [11] is one example of a classifier that estimates decision surfaces directly rather than modeling a probability distribution across the training data. SVMs have demonstrated good performance on several classic pattern recognition problems [13]. Fig. 2 shows a typical two-class problem in which the examples are perfectly separable using a linear decision region. H1 and H2 define two hyperplanes. The distance separating these hyperplanes is called the *margin*. The closest in-class and out-of-class examples lying on these two hyperplanes are called the *support vectors*.

Empirical risk minimization (ERM) [11] can be used to find a good hyperplane, although this does not guarantee a unique solution. Adding an additional requirement that the optimal hyperplane should have good generalization properties can help choose the best hyperplane. The structural risk minimization (SRM) principle imposes structure on the optimization process by ordering the hyperplanes based on the margin. The optimal hyperplane is the one that maximizes the margin while minimizing the empirical risk. This indirectly guarantees better generalization [11]. Fig. 2 illustrates the differences between using ERM and SRM.

An SVM classifier is defined in terms of the training examples. However, all training examples do not contribute to the definition of the classifier. In practice, the proportion of support vectors is small, making the classifier sparse. 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.

Real-world classification problems typically involve data that can only be separated using a nonlinear decision surface. Optimization on the input data in this case involves the use of a kernel-based transformation [11]:

$$K(x_i, x_j) = \Phi(x_i) \bullet \Phi(x_j). \tag{1}$$

Kernels allow a dot product to be computed in a higher dimensional space without explicitly mapping the data into these spaces. A kernel-based decision function has the form

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b. \tag{2}$$

Two commonly used kernels explored in this study are

$$K(x_i, x_j) = (x \bullet y + 1)^d, \quad \text{(polynomial)} \tag{3}$$

$$K(x_i, x_j) = \exp\left\{-\Psi\left(|x - y|^2\right)\right\}. \quad \text{(radial basis)} \tag{4}$$

Radial basis function (RBF) kernels are extremely popular, although data-dependent kernels [21] have recently emerged as a powerful alternative. Although convergence for RBF kernels is typically slower than for polynomial kernels, RBF kernels often deliver better performance [11]. Since there are N dot products involved in the definition of the classifier, where N is the number of support vectors, the classification task scales linearly with the number of support vectors.

Nonseparable data is typically addressed by the use of soft margin classifiers. Slack variables [11] are introduced to relax the separation constraints:

$$x_i \bullet w + b \geq +1 - \xi_i, \quad \text{for } y_i = +1, \tag{5}$$

$$x_i \bullet w + b \leq -1 + \xi_i, \quad \text{for } y_i = -1, \text{ and} \tag{6}$$

$$\xi_i \geq 0, \quad \forall\, i \tag{7}$$

where y_i are the class assignments, w represents the weight vector defining the classifier, b is a bias term, and the ξ_i's are the slack variables. Derivation of an optimal classifier for this nonseparable case exists and is described in detail in [11] and [12].

III. Applications to Speech Recognition

Hybrid approaches for speech recognition [6] provide a flexible paradigm to evaluate new acoustic modeling techniques. These systems do not entirely eliminate the HMM framework because classification models such as SVMs cannot model the temporal structure of speech effectively. Most contemporary connectionist systems use neural networks only to estimate posterior probabilities and use the HMM structure to model temporal evolution [6], [10]. In integrating SVMs into such a hybrid system, several issues arise.

Posterior Estimation: One drawback of an SVM is that it provides an m-ary decision. Most signal processing applications, however, need a posterior probability that captures our uncertainty in classification. This issue is particularly important in speech recognition because there is significant overlap in the feature space. SVMs provide a distance or discriminant that can be used to compare classifiers. This is unlike connectionist systems, whose output activations are estimates of the posterior class probabilities [6], [7].

One of the main concerns in using SVMs for speech recognition is the lack of a clear relationship between distance from the margin and the posterior class probability. A variety of options for converting the posterior to a probability were analyzed in [18], including Gaussian fits and histogram approaches. These methods are not Bayesian in nature in that they do not account for the variability in the estimates of the SVM parameters. Ignoring this variability in the estimates often results in overly confident predictions by the classifiers on the test set [22].

Kwok [14] and Platt [15] have extensively studied the use of moderated SVM outputs as estimates of the posterior probability. Kwok's work also discusses the relationship between the SVM output and the evidence framework. We chose unmoderated probability estimates based on ML fitting as a tradeoff between computational complexity and error performance. We used a sigmoid distribution to map the output distances to posteriors:

$$p(y=1|f) = \frac{1}{1+\exp(Af+B)}. \tag{8}$$

As suggested by Platt, the parameters A and B can be estimated using a model-trust minimization algorithm [23]. An example of the fit for a typical classifier is shown in Fig. 3.

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 by the training data. In order to avoid biased estimates, a cross-validation set must be used to estimate the parameters of the sigmoid [15]. The size of this data set can be determined based on the amount of training data that is available for the classifier.

Classifier Design: A fundamental issue in classifier design is whether the classifiers should be one-versus-one classifiers, which learn to discriminate one class from another class, or one-versus-all, which learn to discriminate one class from all other classes. One-versus-one classifiers are typically smaller and less complex and can be estimated using fewer resources than one-versus-all classifiers. When the number of classes is N, we need to estimate $N(N-1)/2$ one-versus-one classifiers as compared to one-versus-all classifiers. On several standard classification tasks, it has been proven that one-versus-one classifiers are marginally more accurate than one-versus-all classifiers [16], [17]. Nevertheless, for computational efficiency, we chose to use one-versus-all classifiers in all experiments reported here.

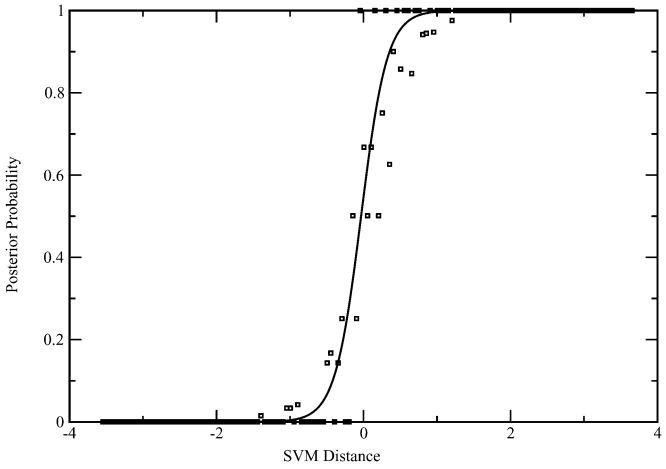

Fig. 3. Sigmoid fit to the SVM distance-based posterior.

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 prevents this at the current time. Although very efficient optimizers are used to train an SVM, it is still not computationally feasible to train on all data available in the large corpora used in this study. In our work, we have chosen to use a segment-based approach to avoid these issues. This allows every classifier to use all positive examples in the corpora and an equal number of randomly selected negative examples.

The other motivating factor for most segment-based approaches [24] is that the acoustic model needs to capture both the temporal and spectral structure of speech that 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 speech data.

A related problem is the variable length or duration problem. Segment durations are correlated with the word choice and speaking rate but are difficult to exploit in an SVM-type framework. A simple but effective approach motivated by the three-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 [10], [24]. The first and third sections model the transition into and out of the segment, whereas 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. The segment vector is then augmented with the logarithm of the duration of the phone instance to explicitly model the variability in duration.

Fig. 4 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 based on the alignments from a baseline three-state Gaussian-mixture 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 three sections per segment, the composite vector has a dimensionality of 117. SVM classifiers are trained on these composite vectors, and recognition is also performed using these segment-level composite vectors.

N-best List Rescoring: As a first step toward building a standalone hybrid SVM/HMM system for continuous speech recognition, we have explored a simple N-best list rescoring paradigm. 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 previously discussed, which are then 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. An overview of the resulting hybrid system is shown in Fig. 5.

The above framework allows the SVM to rescore each N-best entry against the corresponding segmentation derived by a Viterbi alignment. It is also instructive to fix the segmentation so that each N-best entry is rescored against a single, common, segment-level feature stream. One approach to accomplishing this, which represents our second segmentation method, is to define a single feature stream by Viterbi-aligning the one-best hypothesis generated by the baseline HMM system. A third segmentation approach uses a feature stream derived from the reference transcription to investigate the lower error bound when a perfect segmentation is available. These three segmentation methods are explored in the experiments below.

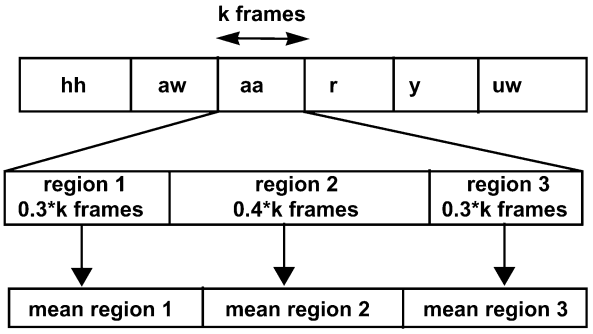

Fig. 4. Overview of a hybrid HMM/SVM system based on an N-best list rescoring paradigm.

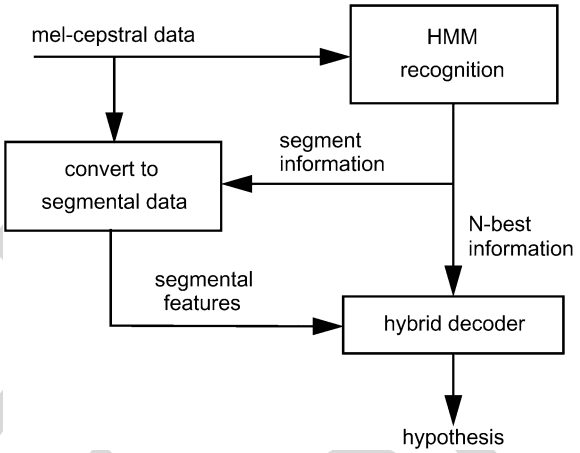

Fig. 5. Composition of the segment level feature vector assuming a 3-4-3 proportion for the three sections.

IV. Experimental Results

We evaluated our SVM approach on three popular tasks: Deterding vowel classification [25], OGI Alphadigits [26], and Switchboard [27]. In these experiments, we compared the performance of the SVM approach to a baseline HMM system that was used to generate segmentations. In this section, we also present a detailed discussion of the classifier design in terms of data distribution and parameter selection.

A. Static Pattern Classification

The Deterding vowel data [25] is a relatively simple but popular static classification task used to benchmark nonlinear classifiers. In this evaluation, the speech data was collected at a 10-kHz sampling rate and lowpass filtered at 4.7 kHz. The signal was then transformed to 10 log-area parameters. A window duration of 50 ms was used to generate 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 an $h*d$ context.

A traditional method for estimating an RBF classifier involves finding the RBF cluster centers for each class separately using a clustering mechanism such as K-MEANS [28]. We estimate weights corresponding to each cluster center to complete the definition of the classifier. The goal of the optimization process is typically not improved discrimination but better representation. The training process requires using heuristics to determine the number of cluster centers. In contrast, the SVM approach for estimating RBF classifiers is more elegant, where the number of centers (support vectors in this case) and their weights are learned automatically in a discriminative framework.

The parameters of interest in tuning an RBF kernel are Ψ, which is the variance of the kernel, and C, which is the parameter used to penalize training errors during SVM estimation [11], [18]. Table I shows the performance of SVMs using RBF kernels for a wide range of parameter values. The results clearly indicate that the performance of the classifiers is very closely tied to the parameter setting, although there exists a pretty wide range of values for which performance is comparable. Another interesting observation is the effect of C on the performance.

TABLE I
EFFECT OF THE KERNEL PARAMETERS ON THE CLASSIFICATION PERFORMANCE OF RBF KERNEL-BASED SVMS ON THE DETERDING VOWEL CLASSIFICATION TASK

gamma (C=10)	classification error %	C (gamma=0.5)	classification error %
0.2	45	1	58
0.3	40	2	43
0.4	35	3	43
0.5	36	4	43
0.6	35	5	39
0.7	35	8	37
0.8	36	10	37
0.9	36	20	36
1.0	37	50	36
		100	36

Note that for values of C greater than 20, the performance does not change. This suggests that a penalty of 20 has already accounted for all overlapped data, and a larger value of C will have no additional benefit.

Table II presents results comparing two types of SVM kernels to a variety of other classification schemes. A detailed description of the conditions for this experiment are described in [18]. Performance using the RBF kernel was better than most nonlinear classification schemes. The best performance we report is 35%. This is worse than the best performance reported on this data set (30% using a speaker adaptation scheme called separable mixture models [29]) but significantly better than the best neural network classifiers (e.g., Gaussian Node Network) [8]. We have observed similar trends on a variety of static classification tasks [30] and recent publications on applications of SVMs to other speech problems [31] report similar findings.

TABLE II
SVM CLASSIFIER USING AN RBF KERNEL PROVIDES SIGNIFICANTLY BETTER PERFORMANCE THAN MOST OTHER COMPARABLE CLASSIFIERS ON THE DETERDING VOWEL CLASSIFICATION TASK

Approach	Error Rate
K-Nearest Neighbor	44%
Gaussian Node Network	44%
SVM: Polynomial Kernels	49%
SVM: RBF Kernels	35%
Separable Mixture Models	30%
RVM: RBF Kernels	30%

B. Spoken Letters and Numbers

We next evaluated this approach on a small vocabulary continuous speech recognition task: OGI Alphadigits (AD) [26]. This database consists of spoken letters and numbers recorded over long distance telephone lines. These words are extremely confusable for telephone-quality speech (e.g., "p" versus "b"). AD is an extremely challenging task from an acoustic modeling standpoint since the language model (a "self-loop" in which each letter or number can occur an arbitrary number of times and can follow any other letter or number) provides no predictive power.

A baseline HMM system described in [18] was used to generate segmented training data 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 improve the convergence properties of the quadratic optimizers used as part of the SVM estimation utilities in SVMLight [32]. For each phone instance in the training transcription, a segment was extracted. This segment was divided into three parts, as shown in Fig. 5. An additional parameter describing the log of the segment duration was added to yield a composite vector of size $3 * 39$ features $+ 1$ log duration $= 118$ features. Once the training sets were generated for all the classifiers, the SVMLight utilities were used to train each of the 29 phone SVM models.

For each phone, an SVM model was trained to discriminate between this phone and all other phones (one-versus-all models), generating a total of 29 models. In order to limit the number of samples (especially the out-of-class data) that is required by each classifier, a heuristic data selection process was used that required the training set to consist of equal amounts of within-class and out-of-class data. All within-class data available for a phone is by default part of the training set. The out-of-class data was randomly chosen such that one half of the out-of-class data came from phones that were phonetically similar to the phone of interest, and one half came from all other phones [18].

Table III gives the performance of the hybrid SVM system as a function of the kernel parameters. These results were generated with ten-best lists whose total list error (the error inherent in the lists themselves) was 4.0%. The list error rate is the best performance any system can achieve by postprocessing these

TABLE III
COMPARISON OF WORD ERROR RATES ON THE ALPHADIGITS TASK 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. THE WER FOR THE BASELINE HMM SYSTEM IS 11.9%

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			

N-best lists. Although we would ideally like this number to be as close to zero as possible, the constraints placed by the limitations of the knowledge sources used in the system (acoustic models, language models etc.) force this number to be nonzero. In addition, the size of the N-best list was kept small to minimize computational complexity.

The goal in SRM is to build a classifier that balances generalization with discrimination on the training set. Table III shows how the RBF kernel parameter is used as a tuning parameter to achieve this balance. As gamma increases, the variance of the RBF kernel decreases. This, in turn, produces a narrower support region in a high-dimensional space. This support region 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 support vectors decreases, which leads to a smoother decision surface. Eventually, we reduce the number of support vectors to a point where the decision region is overly smooth (gamma $= 0.1$), and performance degrades.

As with the vowel classification data, RBF kernel performance was superior to the polynomial kernel. In addition, as we observed in the vowel classification task, the generalization performance was fairly flat for a wide range of kernel parameters. The segmentation derived from the one-best hypothesis from the baseline HMM resulted in an 11.0% WER using an RBF kernel. To provide an equivalent and fair comparison with the HMM system we have rescored the ten-best lists with the baseline HMM system. Using a single segmentation to rescore the ten-best lists does force a few hypothesis in the lists to become inactive because of the duration constraints. The effective average list size after eliminating hypotheses that do not satisfy the duration constraints imposed by the segmentation is 6.9. The result for the baseline HMM system using these new N-best lists remains the same, indicating that the improvements provided by the hybrid-system are indeed because of better classifiers and not the smaller search space.

As a point of reference, we also produced results in Table III that used the reference transcription to generate the segments. Using this *oracle* segmentation (generated using the reference transcription for Viterbi alignment), the best performance obtained on this task was 7.0% WER. This is a 36% relative improvement in performance over the best configuration of the hybrid system using hypothesis-based segmentations. The effective average list size after eliminating hypotheses that do not satisfy the duration constraints imposed by the *oracle* segmentation was 6.7. This experiment shows that SVMs efficiently lock on to good segmentations. However, when we let SVMs choose the best segmentation and hypothesis combination by using the N-best segmentations, the performance gets worse (11.8% WER as shown in Table IV). This apparent anomaly suggests 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.

C. Conversational Speech

The Switchboard (SWB) [27] task, which is based on two-way telephone recordings of conversational speech, is very different from the AD task in terms of acoustic confusability and classifier complexity. The baseline HMM system for this task performs at 41.6% WER. We followed a similar SVM training process to that which was described for the AD task. We estimated 43 classifiers for SWB. Since a standard cross-validation set does not exist for this task, we used 90 000 utterances from 60 h of training data. The cross-validation set consisted of 24 000 utterances. We limited the number of positive examples for any classifier to 30 000. These positive examples were chosen at random.

A summary of our SWB results are shown in Table IV. In the first experiment, we used a segmentation derived from the baseline HMM system's top hypothesis to rescore the N-best list. This hybrid setup does improve performance over the baseline, albeit only marginally—40.6% compared to a baseline of 41.6%. The second experiment was performed by using N segmentations to rescore each of the utterances in the N-best lists. From the experimental results on the AD task, we expected the performance with this setup to be worse than 40.6%. The performance of the system did indeed get worse—42.1% WER. When HMM models were used instead of SVMs with the same setup, the HMM system achieved a WER of 42.3% compared with the baseline of 41.6%. From this result, we deduce that the lack of any language modeling information when we reorder the N-best lists is the reason for this degradation in performance.

The next experiment was an *oracle* experiment. Use of the reference segmentation to rescore the N-best list gave a WER

TABLE IV
EXPERIMENTAL RESULTS COMPARING THE BASELINE HMM SYSTEM TO A NEW HYBRID SVM/HMM SYSTEM. THE HYBRID SYSTEM MARGINALLY OUTPERFORMS THE HMM SYSTEM WHEN USED AS A POSTPROCESSOR TO THE OUTPUT OF THE HMM SYSTEM WHERE THE SEGMENTATION IS DERIVED USING THE HYPOTHESIS OF THE HMM SYSTEM. HOWEVER, WHEN THE REFERENCE TRANSCRIPTION IS INCLUDED, OR WHEN THE REFERENCE-BASED SEGMENTATION IS USED, SVMS OFFER A SIGNIFICANT IMPROVEMENT

Exp.	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	11.9	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

* For the baseline HMM system, recognition is performed by rescoring a word-lattice created from the N-best list.

** For the baseline HMM system, experiments were not performed due to the lack of a time-aligned decoding mechanism.

of 36.1%, confirming our hypothesis that the segmentation issue needs further exploration. A second set of *oracle* experiments evaluated the richness of N-best lists. The N-best list error rate was artificially reduced to 0% by adding the reference to the original ten-best lists. Rescoring these new N-best lists using the corresponding segmentations resulted in error rates of 38.1% and 9.1% WER on SWB and AD, respectively. The HMM system under a similar condition improves performance to 38.6%. On AD, the HMM system does not improve performance over the baseline, even when the reference (or correct) transcription is added to the N-best list. This result suggests that SVMs are very sensitive to segmentations and can perform well if accurate segmentations are available.

Another set of experiments were run to quantify the absolute ceiling in performance improvements the SVM hybrid system can provide. This ceiling can be achieved when we use the hybrid system to rescore the N-best lists that include the reference transcription and the reference-based segmentation. Using this setup, the system gives a WER of 5.8% on SWB and 3.3% on AD. This huge improvement should not be mistaken to be a real improvement in performance for two reasons. First, we cannot guarantee that the reference segmentation is available at all times. Second, generating N-best lists with 0% WER is extremely difficult, if not impossible, for conversational speech. This improvement should rather be viewed as an indication of the fact that by using good segmentations to rescore good N-best lists, the SVM/HMM hybrid system has a potential to improve performance. In addition, using matched training and test segmentations seems to improve performance dramatically.

Table IV summarizes the important results in terms of the various segmentations and N-best lists that were processed to arrive at the final hypothesis. The key point to be noted here is that experiments 2 and 4 are designed such that both the hybrid system and the HMM system are operating under the same conditions and offer a fair comparison of the two systems. For these experiments, since we reorder N-best lists by using segmentations corresponding to each of the hypothesis in the list, both systems have the opportunity to evaluate the same segments. On the other hand if we were to run the experiments using a single segmentation (experiment 1 for example), the HMM system cannot use the segmentation information, whereas the hybrid system can. Experiments 2 and 4 are key in order to compare both systems from a common point of reference. Experiment 4 suggests that when the HMM and hybrid system process good segmentations and rich N-best lists, the hybrid system outperforms the HMM system—significantly in the case of AD and marginally on SWB.

V. CONCLUSIONS

This paper addresses the use of a support vector machine as a classifier in a continuous speech recognition system. The technology has been successfully applied to two speech recognition tasks. A hybrid SVM/HMM system has been developed that uses SVMs to postprocess data generated by a conventional 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 oracle experiments reported here clearly show the potential of this hybrid system while highlighting the need for further research into the segmentation issue.

Our ongoing research into new acoustic modeling techniques for speech recognition is taking two directions. First, it is clear that for a formalism such as SVMs, there is a need for an iterative SVM classifier estimation process embedded within the learning loop of the HMM system to expose the classifier to the same type of ambiguity observed during recognition. This should be done in a way similar to the implicit integration of optimal segmentation and MCE/MMI-based discriminative training of HMMs [4], [5]. Mismatched training and evaluation conditions are dangerous for recognition experiments. Second, the basic SVM formalism suffers from three fundamental problems: scalability, sparsity, and Bayesian-ness. Recent related research [20] based on relevance vector machines (RVMs) directly addresses the last two issues.

Finally, the algorithms, software, and recognition systems described in this work are available in the public domain as part of our speech recognition toolkit (see *http://www.isip.msstate.edu/projects/speech/software*).

ACKNOWLEDGMENT

The inspiration for this work grew out of discussions with V. Vapnik at the 1997 Summer Workshop at the Center for Speech and Language Processing at the Johns Hopkins University. The authors are also grateful to many students at the Institute for Signal and Information Processing at Mississippi State University, including N. Parihar and B. Jelinek, for their contributions to the software infrastructure required for the experiments.

REFERENCES

[1] F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE*, vol. 64, pp. 532–537, 1976.

[2] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood estimation from incomplete data," *J. R. Statist. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[4] E. McDermott, "Discriminative training for speech recognition," Ph.D. dissertation, Waseda Univ., Tokyo, Japan, 1977.

[5] P. Woodland and D. Povey, "Very large scale MMIE training for conversational telephone speech recognition," in *Proc. Speech Transcription Workshop*, 2000.

[6] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition—A Hybrid Approach*. Boston, MA, USA: Kluwer, 1994.

[7] J. S. Bridle and L. Dodd, "An alphanet approach to optimizing input transformations for continuous speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, Toronto, ON, Canada, Apr. 1991., pp. 277–280.

[8] A. J. Robinson, "Dynamic error propagation networks," Ph.D. dissertation, Cambridge Univ., Cambridge, U.K., 1989.

[9] G. D. Cook and A. J. Robinson, "The 1997 ABBOT system for the transcription of broadcast news," in *Proc. DARPA BNTU Workshop*, Lansdowne, VA, 1997.

[10] N. Ström, L. Hetherington, T. J. Hazen, E. Sandness, and J. Glass, "Acoustic modeling improvements in a segment-based speech recognizer," in *Proc. IEEE ASR Workshop*, Keystone, CO, Dec. 1999.

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

[12] ——, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[13] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Knowledge Discovery Data Mining*, vol. 2, no. 2, pp. 121–167, 1998.

[14] J. Kwok, "Moderating the outputs of support vector machine classifiers," *IEEE Trans. Neural Networks*, vol. 10, pp. 1018–1031, Sept. 1999.

[15] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. Cambridge, MA: MIT Press, 1999.

[16] E. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *J. Machine Learning Res.*, vol. 1, pp. 113–141, Dec. 2000.

[17] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," in *Proc. Seventh Eur. Symp. Artificial Neural Networks*, Bruges, Belgium, 1999, pp. 219–224.

[18] A. Ganapathiraju, "Support vector machines for speech recognition," Ph.D. dissertation, Mississippi State University, Mississippi State, MS, 2002.

[19] A. Ganapathiraju, J. Hamaker, and Picone, "A hybrid ASR system using support vector machines," in *Proc. Int. Conf. Spoken Language Processing*, vol. 4, Beijing, China, Oct. 2000, pp. 504–507.

[20] J. Hamaker, J. Picone, and A. Ganapathiraju, "A sparse modeling approach to speech recognition based on relevance vector machines," in *Proc. Int. Conf. Spoken Language Processing*, vol. 2, Denver, CO, Sept. 2002, pp. 1001–1004.

[21] N. Smith and M. Niranjan, "Data-dependent kernels in SVM classification of speech patterns," in *Proc. Int. Conf. Spoken Language Processing*, Beijing, China, Oct. 2000, pp. 297–300.

[22] M. E. Tipping, "The relevance vector machine," in *Advances in Neural Information Processing Systems 12*. Cambridge, MA: MIT Press, 2000, pp. 652–665.

[23] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. London, U.K: Academic, 1981.

[24] J. Chang and J. Glass, "Segmentation and modeling in segment-based recognition," in *Proc. Eurospeech*, Rhodes, Greece, Sept. 1997, pp. 1199–1202.

[25] D. Deterding, M. Niranjan, and A. J. Robinson. (2004) Vowel recognition (Deterding Data), Berkeley, CA. [Online]. Available: http://ftp.ics.uci.edu/pub/machine-learning-databases/undocumented/connectionist-bench/vowel/

[26] R. Cole. (1998) Alphadigit Corpus v1.0. CSLU, OGI, Portland, OR. [Online]. Available: http://www.cse.ogi.edu/CSLU/ corpora/alphadigit

[27] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. ICASSP*, vol. 1, San Francisco, CA, Mar. 1992, pp. 517–520.

[28] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. San Diego, CA: Academic, 1990.

[29] J. Tenenbaum and W. T. Freeman, "Separating style and content," in *Advances in Neural Information Processing Systems 9*. Cambridge, MA: MIT Press, 1997.

[30] J. Picone and D. May. (2003) A Visual introduction to pattern recognition. Mississippi State Univ., Mississippi State, MS. [Online]. Available: http://www.isip.msstate.edu/projects/speech/software/demonstrations/applets/util/pattern_recognition/current/index.html

[31] E. Singer, P. A. Torres-Carrasquillo, T. P. Gleason, and W. M. Campbell, "Acoustic, phonetic, and discriminative approaches to automatic language identification," in *Proc. Eurospeech*, Geneva, Switzerland, Sept. 2003, pp. 1345–1348.

[32] T. Joachims. (1999) SVMLight: support vector machine. Cornell Univ., Ithaca, NY. [Online]. Available: http://www.cs.cornell.edu/People/tj/svm_light/

[33] J. Hamaker and J. Picone, "Advances in speech recognition using sparse Bayesian methods," IEEE Trans. Speech Audio Processing, submitted for publication.

Aravind Ganapathiraju (M'92) was born in Hyderabad, India, in 1973. He received the B.E. degree from Regional Engineering College, Trichy, India, in 1995, the M.S. degree in electrical engineering in 1997, and the Ph.D. degree in computer engineering in 2002, both from Mississippi State University, Mississippi State, MS.

He is currently the VP of Core Technology at Conversay, Redmond, WA. He has been working on various aspects of speech recognition technology for the past eight years. His research interests include providing technology for limited-resource platforms using statistical techniques derived from various fields, including pattern recognition and neural networks. He is also interested in nascent technology areas such as support vector machines and cognitive science. He has served as a reviewer for several journals in the signal processing area.

Jonathan E. Hamaker (M'96) was born in Birmingham, AL, in 1974. He received the B.S. degree in electrical engineering in 1997 and the M.S. degree in computer engineering in 2000, both from Mississippi State University, Mississippi State, MS.

He is currently a member of technical staff at Microsoft, Redmond, WA, where he is focused on the development of core acoustic modeling techniques for speech platforms. His primary research interest is in the application of machine learning to speech recognition in constrained environments. Further interests include noise-robust acoustic modeling and discriminative modeling. He has served as a reviewer for several journals in the signal processing and machine learning areas.

Joseph Picone (M'80–SM'90) was born in Oak Park, IL, in 1957. He received the B.S. degree in 1979, the M.S. degree in 1980, and the Ph.D. degree in 1983, all in electrical engineering from the Illinois Institute of Technology, Chicago.

He is currently a Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS. He has previously been with Texas Instruments, Dallas, TX, and AT&T Bell Laboratories, Murray Hill, NJ. His primary research interests are the application of machine learning techniques to speech recognition and the development of public domain speech technology. He is a registered Professional Engineer, been awarded eight patents, and has published more than 120 papers in the area of speech processing.

Dr. Picone has served in several capacities with the IEEE.