SUPPORT VECTOR MACHINES FOR SEGMENTAL MINIMUM BAYES RISK DECODING OF CONTINUOUS SPEECH

Veera Venkataramani, Shantanu Chakrabartty, and William Byrne

Center for Language and Speech Processing,
The Johns Hopkins University,
Baltimore, MD-21218, USA.
{*veera,shantanu,byrne*}*@jhu.edu*

ABSTRACT

Segmental Minimum Bayes Risk (SMBR) Decoding involves the refinement of the search space into sequences of small sets of confusable words. We describe the application of Support Vector Machines (SVMs) as discriminative models for the refined search spaces. We show that SVMs, which in their basic formulation are binary classifiers of fixed dimensional observations, can be used for continuous speech recognition. We also study the use of $Gini$SVMs, which is a variant of the basic SVM. On a small vocabulary task, we show this two pass scheme outperforms MMI trained HMMs. Using system combination we also obtain further improvements over discriminatively trained HMMs.

1. INTRODUCTION

Support Vector Machines [1] are pattern recognizers that classify data without making any assumptions about the underlying process by which the observations were generated. In their basic formulation SVMs are binary classifiers. Given a data sample to be classified, the SVM will assign it as belonging to one of two classes. In training an SVM each labeled data point is represented as a real valued vector of fixed high dimension. The SVM is defined by a hyperplane in this feature space that is constructed so as to maximize a measure of the "margin" between two classes. A new data sample is classified by the SVM according to the decision boundary defined by the hyperplane. The location of the hyperplane is usually determined by a small number of the training samples which are ideally those near the boundaries of the two classes. As a consequence, SVMs are often observed to generalize well in cases when training data is limited. It is also possible to improve classification performance by transforming the raw data into a higher dimensional feature space so that the two classes can be more easily separated by a linear classifier. Due to these and other beneficial properties, SVMs have been successfully used in many pattern recognition tasks [2] [3]. In this paper, we take the simple view that an SVM is a binary classifier of fixed-length data vectors.

In speech recognition we would like to classify a variable length sequence of fixed dimension patterns which are typically vectors of acoustic spectral energy measurements. These raw observation sequences can be expected neither to have fixed dimension nor to belong to one of only two classes. Only the simplest of word or phrase recognition tasks can be described as binary classification of fixed-duration sequences. If SVMs are to be employed in continuous ASR, their simple formulation as binary classifiers will have to be overcome or circumvented.

Smith *et al.* [4] have developed *score-spaces* [5] to represent a variable length sequence of acoustic vectors via fixed dimensional vectors. This is done by using HMMs to find the likelihood of each sequence to be classified and then computing the gradient of the likelihood with respect to the HMM parameters. Since the HMMs have a fixed number of parameters, this yields a fixed-dimension feature to which the SVMs can be applied. It has the added benefit that the features provided to the SVM can be derived from a well-trained HMM recognizer. However, the SVM is still essentially a binary classifier, so that this approach is still limited to the binary classification of variable length sequences.

To apply SVMs beyond the two-class problem we employ an approach to continuous speech recognition in which the recognition task is transformed into sequential, independent classification tasks. Each of these sub-tasks will be a binary recognition problem in which the goal is to decide which of two words were spoken. This yields a large but manageable sequence of binary decision problems and SVMs will be trained and applied to each. This is fundamentally an ASR rescoring approach. HMMs are used to generate recognition lattices in the usual way, and these lattices are post-processed to identify regions of acoustic confusability in which the first-pass HMMs were unable to distinguish between competing word hypotheses. The goal of this work is to apply SVMs to resolve the uncertainty remaining after the first-pass HMM-based recognizer. We will build on previous work in which this two-pass recognition approach was used to develop specialized discriminative training procedures for HMMs [6, 7].

We refer to this divide-and-conquer recognition strategy as *acoustic code-breaking*. The idea is first to perform an initial recognition pass with the best possible system available, which we take as HMM-based; then isolate and characterize regions of acoustic confusion encountered in the first-pass; and finally apply models to each region that are specially trained for these confusion problems. This provides a framework for incorporating models that might not otherwise be appropriate for continuous speech recognition. We observe in passing that since the first-pass HMM system provides a proper posterior distribution over sequences, this approach may be less affected by the label-bias problem that can be encountered when discriminative classifiers are applied in sequential classification [8].

To place our work in context, there have been previous appli-

This work was supported by the NSF (U.S.A) under the Information Technology Research (ITR) program, NSF IIS Award No. 0122466.

cations of SVMs to speech recognition. Ganapathiraju *et al.* [9] obtain a fixed dimension problem by using a heuristic method to normalize the durations of each variable length sequence. The distances to the decision boundary in feature space are then transformed into phone posteriors using sigmoidal non-linearities. Smith *et al.* [4] use score-spaces to train SVMs followed by a majority voting scheme among binary SVMs to recognize isolated letters. Golowich *et al.* [10] interpret multi-class SVM classifiers as an approximation to multiple logistic smoothing spline regression and use the resulting SVMs to obtain state emission densities of HMMs. Forward Decoding Kernel Machines [11] perform maximum a posteriori forward sequence decoding, where transition probabilities are regressed as a kernel expansion of acoustic features and trained by maximizing a lower bound on a regularized form of cross-entropy.

In the following sections we review the Segmental Minimum Bayes Risk framework that we use for sequence recognition. We then give brief descriptions of SVMs and scores-spaces, providing only the detail needed to describe our work. We will describe the use of $Gini$SVMs that will allow non-positive kernels to used for sequence classification. We then present our experiments and results followed by our conclusions and ideas for future work.

2. SMBR FOR SPEECH RECOGNITION

Given a suitable loss function $l(W, W')$ between two word strings W and W', the Minimum Bayes Risk (MBR) [12] decoder attempts to minimize the empirical risk. It is formulated as

$$\hat{W} = \underset{W' \epsilon \mathcal{W}}{\operatorname{argmin}} \sum_{W \epsilon \mathcal{W}} l(W, W') P(W|A) \tag{1}$$

where $\mathcal{W}$ represents all possible word strings in the grammar and A are the observed acoustics.

For this search to be practical, $\mathcal{W}$ is usually represented by the paths in a N-Best list or a lattice. However, the summation and minimization over this search space (between word strings W and W') in MBR decoders can still be prohibitively expensive. SMBR [13, 14] decoders address this issue by reducing the search problem to a sequence of smaller independent search problems, *i.e.*, the lattice is broken up or cut into a sequence of M smaller sub-lattices. Under certain assumptions [13, 15], the MBR search Equation 1 decomposes into a sequence of independent MBR searches over each of the sub-lattices. Standard MBR decoding is then performed over each of these smaller lattices

$$\hat{W}_i = \underset{W_i' \epsilon \mathcal{W}_i}{\operatorname{argmin}} \sum_{W \epsilon \mathcal{W}_i} l(W_i, W_i') P_i(W|A) \tag{2}$$

where $\hat{W}_i$ is the best path in the ith sub-lattice and $\mathcal{W}_i$ represents all possible strings in the ith sub-lattice. Finally, the sentence-level MBR hypothesis is obtained as $\hat{W} = \hat{W}_1 \cdot \hat{W}_2 \cdots \hat{W}_M$.

There are many lattice cutting schemes. In risk-based lattice cutting [16], each path in the lattice is aligned to the MAP sentence hypothesis $\hat{W}$. The path is then segmented so that the loss function relative to the MAP hypothesis remains consistent, *i.e.*, $l(\hat{W}, W') = \sum_{i=1}^{M} l(\hat{W}_i, W_i')$.

Lattice cutting produces *pinched lattices* (Fig. 1, middle). The segmentation process is designed so that the structure of the original lattice is not disrupted: new paths may be introduced, but no paths in the original lattice are lost, except possibly by pruning.

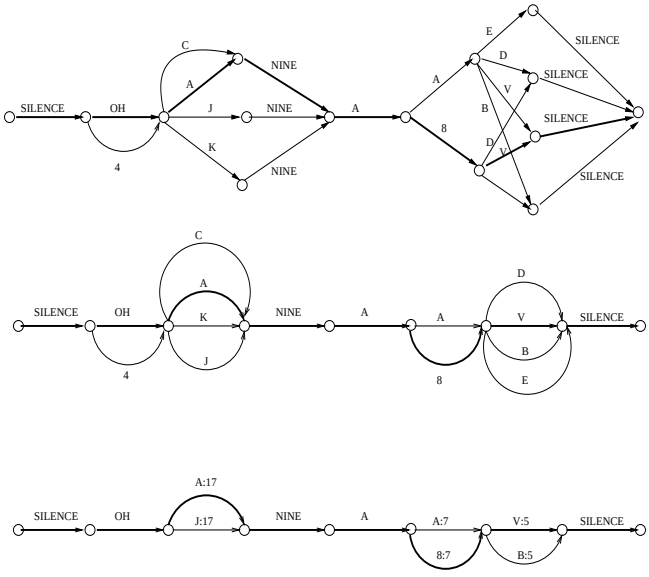

Fig. 1. Lattice Segmentation for Estimation and Search. *Top*: First-pass lattice of likely sentence hypotheses with MAP path in bold; *Middle*: Alignment of lattice paths to MAP path; *Bottom*: Refined search space $\hat{\mathcal{W}}_i$ consisting of segment sets selected for discriminative training and rescoring

Since the paths from the original lattice are preserved, we can use these pinched lattices for acoustic rescoring.

In this work we used Period-1 risk-based lattice cutting. This produces sub-lattices whose strings are at most one word long (Fig. 1, bottom). We prune these so that only the the MAP hypothesis remains in regions of high confidence; in regions of low confidence, the pinched lattice contains the MAP hypothesis along with the competing word hypotheses. We perform the pruning aggressively so that in regions of acoustic confusability there are at most two competing words - the MAP hypothesis and one other. Each of these segments is called a *confusion pair*. These are word pairs, *e.g.*, $\{V, B\}$. Associated with each instance of these pairs in the lattices are the acoustic segments that caused these confusions; these are the acoustic observations and their time boundaries provided by the lattice.

3. $GINI$SVMS

We now briefly review the $Gini$SVM [11]. Given training data $\{\mathbf{x}_i\}_{i=1}^N$ and their labels $\{y_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbf{R}^N$ and $y_i \in \{-1, +1\}$, the basic SVM searches for the hyperplane with the largest separating margin by minimizing a regularized cost function.

$Gini$SVM is a multi-class probabilistic regression machine that provides conditional probability estimates of each class. For a binary classification problem, $Gini$SVM reduces to a special case of the quadratic SVM and minimizes the following cost function

$$\frac{1}{2} \sum_{i,j} \alpha_i [\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) + \frac{2\gamma}{C} \delta_{ij}] \alpha_j - 2\gamma \sum_i \alpha_i \tag{3}$$

subject to

$$\sum_i y_i \alpha_i = 0, \quad 0 \le \alpha_i \le C, \tag{4}$$

where γ is the rate distortion factor chosen as $2log2$ and C is the SVM trade-off parameter that determines how well the SVM fits the training data. Similar to the usual SVM formulation, $Gini$SVMs employ a kernel $\mathbf{K}(\cdot,\cdot)$ to map input vectors to a higher dimension space. $Gini$SVMs have the advantage that, unlike SVMs, they can employ non positive-definite kernels.

New observations $\mathbf{x}$ are then classified as

$$y = \mathrm{sgn}(\sum_i y_i \alpha_i \mathbf{K}(\mathbf{x}, \mathbf{x}_i)) + \mathbf{b} \tag{5}$$

where $\mathbf{b}$ is the bias of the hyperplane that results from the constraints of the cost function [1].

4. SCORES AND SCORE-SPACES

Fisher scores [5] have been suggested as a means to map variable length observation sequences into fixed dimension vectors and the use of Fisher scores has been investigated for ASR [4]. Each component of the Fisher score is defined as the sensitivity of the likelihood of the observed sequence to each parameter of an HMM.

If $\mathbf{O}$ is an observation sequence and $\theta = [\theta_1^\top, \theta_2^\top]^\top$ are the parameters of two HMMs trained for the binary classes 1 and 2, the projection of the observation sequence into the score-space is given by

$$\begin{aligned} \varphi(\mathbf{O}) &= \begin{bmatrix} 1 \\ \nabla_\theta \end{bmatrix} \ln\left(\frac{p(\mathbf{O}|\theta_1)}{p(\mathbf{O}|\theta_2)}\right) \\ &= \begin{bmatrix} \ln \frac{p(\mathbf{O}|\theta_1)}{p(\mathbf{O}|\theta_2)} \\ \nabla_{\theta_1} \ln p(\mathbf{O}|\theta_1) \\ -\nabla_{\theta_2} \ln p(\mathbf{O}|\theta_2) \end{bmatrix} \end{aligned} \tag{6}$$

We first define the parameters of the j^{th} Gaussian observation distribution associated with state s in HMM i as $(\mu_{i,s,j}, \Sigma_{i,s,j})$. In this work we derive the score space solely from the means of the multiple-mixture Gaussian HMM state observation distributions, denoted via the shorthand $\theta_i[s,j,k] = \mu_{i,s,j}[k]$; the decision to focus only on the Gaussian means will be discussed in Section 7. The gradient with respect to these parameters [4] is

$$\nabla_{\mu_{i,s,j}} \ln P(O|\theta_i) = \sum_{t=1}^{T} \gamma_{i,s,j}(t) \left[(o_t - \mu_{i,s,j})^\top \Sigma_{i,s,j}^{-1} \right]^\top,$$

where $\gamma_{i,s,j}$ is the posterior for mixture component j, state s under the i^{th} HMM found via the Forward-Backward procedure; and T is the number of frames in the observation sequence.

We now discuss issues in using scores derived in this way as features to be classified by SVMs.

5. SVM IMPLEMENTATION

We first adjust the scores for each utterance via mean and variance normalization. The normalized scores are given by

$$\varphi^N(\mathbf{O}) = \hat{\Sigma}_{sc}^{-1/2}[\varphi(\mathbf{O}) - \hat{\mu}_{sc}], \tag{7}$$

where $\hat{\mu}_{sc}$ and $\hat{\Sigma}_{sc}$ are estimates of the mean and variances of the scores as computed over the training data of the SVM. Ideally, the SVM training will subsume the $\hat{\mu}_{sc}$ bias and the variance normalization would be performed by the scaling matrix $\hat{\Sigma}_{sc}$ as

$$\varphi^N(\mathbf{O}) = \hat{\Sigma}_{sc}^{-1/2}\varphi(\mathbf{O}) \tag{8}$$

where $\hat{\Sigma}_{sc} = \int \varphi(\mathbf{O})'\varphi(\mathbf{O}) P(\mathbf{O}|\theta) d\mathbf{O}$. For implementation purposes, the scaling matrix is approximated over the training data as

$$\hat{\Sigma}_{sc} = \frac{1}{N-1} \sum (\varphi(\mathbf{O}) - \hat{\mu}_{sc})^\top (\varphi(\mathbf{O}) - \hat{\mu}_{sc}) \tag{9}$$

where $\hat{\mu}_{sc} = \frac{1}{N}\sum \varphi(\mathbf{O})$, and N is the number of training samples for the SVM. However we used a diagonal approximation for Σ_{sc} since the inversion of the full matrix $\hat{\Sigma}_{sc}$ is problematic. Prior to the mean and variance normalization, the scores for each utterance are normalized by the utterance length T.

For ASR, the linear kernel ($\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i{}' \cdot \mathbf{x}_j$), has previously been found to perform best among a variety of positive-definite kernels [17]. We found that while the linear kernel does provide some discrimination, it was not sufficient for satisfactory performance. This observation can be illustrated using kernel maps. A kernel map is a matrix plot that displays kernel values between pairs of observations drawn from two classes, $\mathbf{C}_1$ and $\mathbf{C}_2$. Ideally if $\mathbf{x}, \mathbf{y} \in \mathbf{C}_1$ and $\mathbf{z} \in \mathbf{C}_2$, then $\mathbf{K}(\mathbf{x}, \mathbf{y}) \gg \mathbf{K}(\mathbf{x}, \mathbf{z})$. and the kernel map would be block diagonal. In Figs. 2 and 3, we draw 100 samples each from two classes to compare the linear kernel map to the $\tanh$ kernel ($\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \tanh(d * \mathbf{x}_i{}' \cdot \mathbf{x}_j)$) map. Visual inspection shows that the map of the $\tanh$ kernel is closer to block diagonal. We have found in our experiments with $Gini$SVM that the $\tanh$ kernel far outperformed the linear kernel; we therefore focus on $\tanh$ kernels for the rest of the paper.

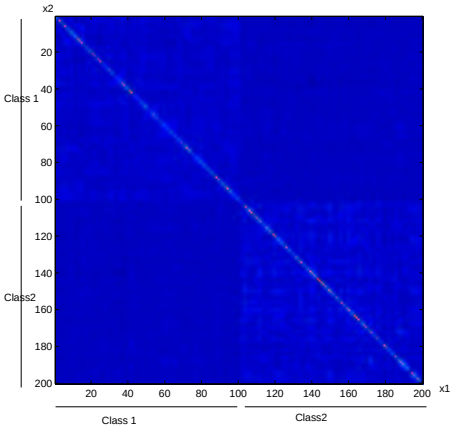

Fig. 2. Kernel Map $\mathbf{K}(\mathbf{x_i}, \mathbf{x_j})$ for the linear kernel over two class data.

The $Gini$SVM classification performance was found to be sensitive to the SVM trade-off parameter C. Unless mentioned otherwise, a value of $C = 1.0$ was chosen for all the experiments in this paper to balance between over-fitting and the time required for training.

For efficiency and modeling robustness there may be value in reducing the dimensionality of the score-space. There has been

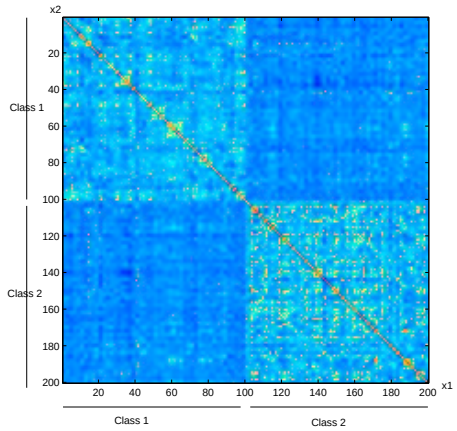

Fig. 3. Kernel Map $\mathbf{K}(\mathbf{x_i}, \mathbf{x_j})$ for $\tanh$ kernel over two class data.

research [18] [17] to estimate the information content of each dimension so that non-informative dimensions can be discarded. Assuming independence between dimensions, the goodness of a dimension can be found based on Fisher discriminant scores as [17]

$$g[d] = \frac{|\hat{\mu}_{sc[1]}[d] - \hat{\mu}_{sc[2]}[d]|}{\hat{\Sigma}_{sc[1]}[d] + \hat{\Sigma}_{sc[2]}[d]} \tag{10}$$

where $\hat{\mu}_{sc[i]}(d)$ is the dth dimension of the mean of the scores of the training data with label i and $\hat{\Sigma}_{sc[i]}[d]$ are the corresponding diagonal variances. SVMs can then be trained only in the most informative dimensions by applying a pruning threshold to $g[d]$.

6. SVMS IN AN SMBR FRAMEWORK

We now describe the steps to incorporate SVMs in the SMBR framework.

6.1. Identifying confidence sets in the training set

Initial lattices are generated using the baseline HMM system to decode the speech in the training set. The lattices produced are then aligned against the reference transcriptions [13]. Period-1 lattice cutting is performed and each sub-lattice is pruned (by the word posterior) to contain two competing words. This process identifies regions of confusion in the training set. The most frequently occurring confusion pairs (confusable words) are kept, and their associated acoustic segments are identified, retaining time boundaries and the true identity of the word spoken.

6.2. Training SVMS for each confusion pair

For each acoustic segment in every sub-lattice, likelihood-ratio scores as given by Equation 6 are generated. The dimension of these scores is equal to the sum of the number of parameters of the two competing HMMs plus one. If necessary, the dimension of the score-space is reduced using the goodness criterion (Equation 10) with appropriate thresholds. SVMs for each confusion pair are then trained in our normalized score-space using the appropriate acoustic segments identified as above.

6.3. SMBR decoding with SVMs

Initial test set lattices are generated using the baseline HMM system. The MAP hypothesis is obtained from this decoding pass and the lattice is aligned against it. Period-1 lattice pinching is performed on the test set lattices. Instances of confusion pairs for which SVMs were trained are identified and retained; other confusion pairs are pruned back to the MAP word hypothesis.

The appropriate SVM is applied to the acoustic segment associated with each confusion pair in the lattice. The HMM outputs in the regions of high confidence are concatenated with the outputs of the SVMs in the regions of low confidence. This is the final hypothesis of the SMBR-SVM system.

6.4. Rationale

The most ambitious formulation of acoustic code-breaking is to first identify all acoustic confusion in the test set, and then return to the training set to find any data that can be used to train models to remove the confusion. To present these techniques and show that they can be effective, we have chosen for simplicity, to focus on modeling the most frequent errors found in training. Earlier work [6] has verified that training set errors found in this way are good predictors of errors that will be encountered in unseen data.

7. EXPERIMENTS AND RESULTS

We evaluate our proposed method on the OGI-Alphadigits corpus [19]. This is a small vocabulary task that is fairly challenging. The baseline Word Error Rates (WERs) for ML models are around 10%; this ensures that there are enough number of errors to allow for analysis. The corpus has a vocabulary of 36 words: 26 letters and 10 digits. The corpus has 46,730 training and 3,112 test utterances. We first describe the training procedure for the various baseline models. A more detailed description can be found in Doumpiotis *et al.* [7].

Word based HMMs were trained for each of the 36 words. The word models were left-to-right with approximately 20 states each, 12 mixtures per state. The data are parametrized as 13 dimensional MFCC vectors with first and second order differences. The baseline ML models were trained HTK-style [20]. The AT&T decoder [21] was used to generate lattices on both the training and the test set. Since the corpus has no language model (each utterance is a random six word string), an unweighted free loop grammar was used during decoding. MMI training was performed [22] [23] at the word level using word time boundaries taken from the lattices. A new set of lattices for both the training and the test sets was then generated using the MMI models. The Lattice Oracle Error Rate for these lattices was 1.27%. Period-1 lattice cutting is then performed on these lattices; the number of confusable words in each segment is further restricted to two. This increased the Lattice Oracle Error Rate to 3.11%. At this point there are two sets of confusion pairs from the pinched lattices, one set comes from the training data, and the other from the test data. We keep the 50 confusion pairs that are observed most frequently in the test data. All other confusion pairs in training and test data are pruned back. We emphasize that this is a 'fair' process; the truth is not used in identifying confusion. Pinched Lattice MMI (PLMMI) [7] is then performed on the MMI models with these lattices.

Table 1 presents the results for the baseline HMM systems. Even though the pinched lattices have a higher oracle error rate,

we see the PLMMI models have substantial gains over the MMI models (7.98% vs. 9.07%).

SVMs were then trained for the 50 dominant confusion pairs using the $Gini$SVM toolkit [24]. Log-likelihood ratio scores were generated from the 12 mixture MMI system. The time boundaries were estimated by the same HMM system. The scores are then normalized as described in section 5.

We initially investigated score spaces constructed from both Gaussian mean and variance parameters. However, training SVMs in this complete score space is impractical since the dimension of the score space is prohibitively large; the complete dimension is approximately 40,000. Filtering these dimensions based on Equation 10 made training feasible, however performance was not much improved. We hypothesize that there is significant dependence between the model means and variances so that the underlying assumptions of the goodness criterion are violated.

We then used only the filtered mean sub-space scores for training SVMs (training on the unfiltered mean sub-space is still impractical because of the prohibitively high number of dimensions). The best performing SVMs used around 2,000 of the most informative dimensions, which is approximately 10% of the complete mean space. As shown in Table 1, applying SVMs to the MMI system yields a significant 9.5% relative reduction in WER from 9.07% to 8.20%. This demonstrates that the SMBR-SVM system can be used to improve performance of MMI trained HMM continuous speech recognition systems.

In comparing the MMI and SMBR-SVM hypotheses, we observed that they differ by more than 4%; this has been observed in some but not all previous work [10, 25, 4]. We therefore performed a simple system combination: for each acoustic segment in a confusion set, if the posterior of the HMM output is greater than a threshold, we accept the HMM output; else, we choose the SVM output. This was done because in these experiments the SVM posteriors were not found to be reliable indicators of word correctness. This combined system ('Voting' in Table 1) gives comparable performance to the PLMMI system (8.04% vs.7.98%).

SVMs were also trained on the filtered mean only sub-space of the 12 mixture PLMMI models. The best performing SVMs in this case also used 10% of the most informative dimensions. While the performance was comparable to the PLMMI HMM system, we still do not improve upon it (8.01% vs. 7.98%) However, the same system combination scheme outlined above does produce significant gains over the PLMMI HMM system (7.73% vs. 7.98%).

Finally, the effect of the SVM trade-off parameter (C in Equation 4) was studied. Figure 4 presents the WER results from training the SVMs for the confusion pairs at different values of C. We find some sensitivity to C, however optimal performance was found over a fair broad range of values (0.3 to 1.0).

All experiments reported thus far employ a global trade-off parameter value for the SVMs trained for the confusion pairs. We now investigate tuning the trade-off parameter for each SVM. The results in Table 2 show that further gains can be obtained by finding the optimal value of this parameter for each SVM. The oracle result is obtained by 'cheating' and choosing the parameter for each SVM that yields the lowest class error rate. An alternative systematic rule for choosing the parameter based on the number of training examples is presented in Table 3 where C decreases with the amount of training data. WER results using SVMs trained with the trade-off parameter set by this rule are presented in Table 2. By this tuning we find that the SVMS have the potential to improve over the PLMMI HMMs.

	HMM	SMBR-SVM	Voting
ML	10.14	-	-
MMI	9.07	8.20	8.04
PLMMI	7.98	8.01	7.73

Table 1. WERs for HMM and SMBR-SVM systems.

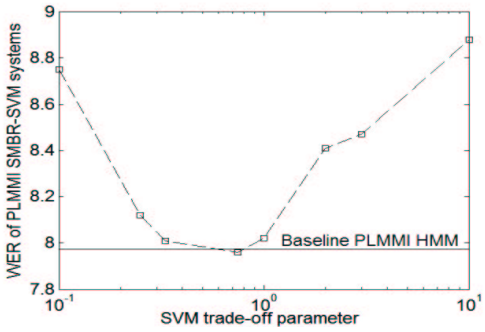

Fig. 4. WERs for different PLMMI SMBR-SVM systems as the global SVM trade-off parameter (C) is varied.

8. CONCLUSIONS AND FUTURE WORK

We have introduced and developed a new approach for the application of SVMs in ASR. The idea is to first perform an initial recognition pass with the best possible HMMs; then isolate and characterize regions of acoustic confusion; and then use specially trained SVMs to resolve these confusions. On a small vocabulary task, we showed significant improvements over MMI trained HMMs. While we find significant improvements over MMI training, we are still investigating the best way to incorporate the recently developed PLMMI training procedure into the SMBR-SVM framework. However, we find that system combination yields improvement over both these types of discriminative training.

We have also investigated the use of $Gini$SVMs, a variant of the basic SVMs, for their use in ASR. We found significant improvements over basic SVMs which we believe is due to the ability of $Gini$SVMs to incorporate non-positive-definite kernels in training.

We also see considerable improvement in the performance of SVMs through selection of the most informative score-space dimensions, as has been noted [17]. We suspect this to be an artifact of the approximation to the scaling matrix Σ_{sc}. If improved normalization of the score-space is found either through better numerical methods or an improved modeling formulation, the SMBR-SVM formulation should yield improvements over pure HMM formulations [5].

Previous work [17] suggests that the best performing HMMs are not necessarily the best HMMs to seed the SVMs. In our case use of any system other than that used to generate lattices leads to complications in implementing SMBR-SVM systems. This requires further work.

We have so far studied a simple task so that we could develop this modeling framework and present it without complications.

	HMM	SMBR-SVM	Voting
PLMMI	7.98	8.01	7.73
Oracle	-	7.77	7.59
Piecewise C	-	7.88	7.67

Table 2. WERs for SMBR-SVM systems with trade-off parameter tuning.

N	$N > 10{,}000$	$N > 10{,}000$ $N < 5{,}000$	$N > 5{,}000$ $N < 500$	$N < 500$
C	0.33	0.75	1.0	2.0

Table 3. Piecewise Rule for choosing trade-off parameter (C) through the number of training observations (N).

Our ultimate goal is however to apply this framework to large vocabulary continuous speech recognition, where we expect to face data sparsity and prohibitively large score-space dimensions.

We have not made use of the ability of the $Gini$SVMs to generate conditional probability estimates over hypotheses. We expect to be able to see further improvements in system combination by deriving these posteriors directly from the SVM.

Acknowledgments. We would like to thank Prof. Gert Cauwenberghs for helpful ideas and discussions. All baseline MMI and PLMMI models were trained by Vlasios Doumpiotis; the ML models were trained by Teresa M. Kamm.

9. REFERENCES

[1] V. Vapnik, *The Nature of Statistical Learning Theory*, chapter 5, Springer-Verlag, New York, Inc., 1995.

[2] C. J. Burges and B. Schölkopf, "Improving the accuracy and speed of support vector learning machines," in *Advances in Neural Information Processing Systems 9*, M. Mozer, M. Jordan, and T. Petsche, Eds. 1997, pp. 375–381, Cambridge: MIT Press.

[3] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in neural Information Processing Systems 9*. 1997, pp. 155–161, Cambridge: MIT Press.

[4] N. D. Smith, M. J. F. Gales, and M. Niranjan, "Data-dependent kernels in SVM classification of speech patterns," Tech. Rep. CUED/F-INFENG/TR387, Cambridge University Eng. Dept., April 2001.

[5] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Advances in Neural Information Processing System*, S. A. Solla M. S. Kearns and D. A. Cohn, Eds. 1998, MIT Press.

[6] V. Doumpiotis, S. Tsakalidis, and W. Byrne, "Discriminative training for segmental minimum Bayes risk decoding," in *ICASSP*, Hong Kong, 2003.

[7] V. Doumpiotis, S. Tsakalidis, and W. Byrne, "Lattice segmentation and minimum Bayes risk discriminative training," in *Eurospeech*, 2003.

[8] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001.

[9] A. Ganapathiraju, J. Hamaker, and J. Picone, "Advances in hybrid SVM/HMM speech recognition," in *GSPx / International Signal Processing Conference*, Dallas, Texas, USA, 2003.

[10] S. E. Golowich and D. X. Sun, "A support vector/hidden Markov model approach to phoneme recognition," in *ASA Proceedings of the Statistical Computing Section*, 1998, pp. 125–130.

[11] S. Chakrabartty and G. Cauwenberghs, "Forward decoding kernel machines: A hybrid HMM/SVM approach to sequence recognition," in *Proc. SVM'2002, Lecture Notes in Computer Science*. 2002, vol. 2388, pp. 278–292, Cambridge: MIT Press.

[12] V. Goel and W. Byrne, "Minimum Bayes-risk automatic speech recognition," in *Computer Speech & Language*, 2000, vol. 14(2).

[13] V. Goel, S. Kumar, and W. Byrne, "Confidence based lattice segmentation and minimum Bayes-risk decoding," in *Eurospeech*, Aalborg, Denmark, 2001.

[14] V. Goel, S. Kumar, and W. Byrne, "Segmental minimum Bayes-risk decoding for automatic speech recognition," *IEEE Transactions on Speech and Audio Processing*, 2003, *to appear*.

[15] V. Goel and W. Byrne, "Minimum Bayes-risk automatic speech recognition," in *Pattern Recognition in Speech and Language Processing*, W. Chou and B.-H. Juang, Eds. CRC Press, 2003.

[16] S. Kumar and W. Byrne, "Risk based lattice cutting for segmental minimum Bayes-risk decoding," in *ICSLP*, Denver, Colorado, USA, 2002.

[17] N. D. Smith and M. J. F. Gales, "Using SVMs to classify variable length speech patterns," Tech. Rep. CUED/F-INFENG/TR412, Cambridge University Eng. Dept., April 2002.

[18] A. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997.

[19] M. Noel, *Alphadigits*, CSLU, OGI, Available: http://www.cse.ogi.edu/CSLU/corpora/alphadigit, 1997.

[20] S. Young et. al., *The HTK Book, Version 3.0*, July 2000.

[21] M. Mohri, F. Pereira, and M. Riley, *AT&T General-purpose Finite-State Machine Software Tools*, Available: http://www.research.att.com/sw/tools/fsm/, 2001.

[22] Y. Normandin, "Maximum mutual information estimation of hidden Markov models," in *Automatic Speech and Speaker Recognition*, S. A. Solla M. S. Kearns and D. A. Cohn, Eds. 2002, vol. 15, Cambridge: MIT Press.

[23] P. C. Woodland and D. Povey, "Large scale discriminative training for speech recognition," in *Proc. ITW ASR, ISCA*, 2000.

[24] S. Chakrabartty, *The giniSVM toolkit, Version 1.2*, Available: http://bach.ece.jhu.edu/svm/ginisvm/, 2003.

[25] S. Fine, J. Navrátil, and R. Gopinath, "A hybrid GMM/SVM approach to speaker identification," in *ICASSP*, Utah, USA, 2001.