*Gini*Support Vector Machines for Segmental Minimum Bayes Risk Decoding of Continuous Speech [1]

Veera Venkataramani

Fair Isaac Corporation, San Diego, CA 92130, USA

Shantanu Chakrabartty

Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA.

William Byrne [*]

Department of Engineering, Cambridge University, Cambridge CB2 1PZ, U.K.

Abstract

We describe the use of Support Vector Machines (SVMs) for continuous speech recognition by incorporating them in Segmental Minimum Bayes Risk decoding. Lattice cutting is used to convert the Automatic Speech Recognition search space into sequences of smaller recognition problems. SVMs are then trained as discriminative models over each of these problems and used in a rescoring framework. We pose the estimation of a posterior distribution over hypotheses in these regions of acoustic confusion as a logistic regression problem. We also show that *Gini*SVMs can be used as an approximation technique to estimate the parameters of the logistic regression problem. On a small vocabulary recognition task we show that the use of *Gini*SVMs can improve the performance of a well trained Hidden Markov Model system trained under the Maximum Mutual Information criterion. We also find that it is possible to derive reliable confidence scores over the *Gini*SVM hypotheses and that these can be used to good effect in hypothesis combination. We discuss the problems that we expect to encounter in extending this approach to large vocabulary continuous speech recognition and describe initial investigation of constrained estimation techniques to derive feature spaces for SVMs.

Key words: Support Vector Machines, Segmental Minimum Bayes Risk decoding, discriminative training, continuous speech recognition

---

1 Introduction

In their basic formulation Support Vector Machines (SVMs) (Vapnik, 1995) are binary classifiers of fixed dimension feature vectors. An SVM is defined by a hyperplane in the feature space that serves as a decision boundary between two classes. This hyperplane is usually determined by a small number of training samples located at the class boundary, so that SVMs generalize well from limited training data. These data vectors can also be transformed into higher dimensional feature spaces so that they can be more easily separated by a linear classifier. These properties, together with an elegant and powerful formalism, have motivated the successful application of SVMs to many pattern recognition problems (Burges, 1998).

The difficulties involved in applying SVMs to automatic speech recognition (ASR) are apparent. Speaking rate fluctuations, pauses, disfluencies, and other spontaneous speech effects prevent a simple mapping of the acoustic signal to a fixed dimension representation. Moreover, the recognition decision space is defined by the ASR task grammar, and in only the simplest of tasks is this a binary decision. Even with techniques that extend SVMs to multiclass problems (Weston and Watkins, 1998), it is unlikely that a single classifier will be powerful enough to distinguish all permissible sentences in a natural language application. For SVMs to be employed in continuous ASR their formulation as isolated-pattern classifiers of fixed dimension observations must be either overcome, or the ASR problem itself must be redefined. In this work we take the latter approach.

We transform the continuous speech recognition problem into sequential, independent, classification tasks. Each of these sub-tasks is an isolated recognition problem in which the objective is to decide which of several words or phrases were spoken. Binary problems in this collection are extracted, and specialized SVMs are trained and applied to each problem. In this way we transform the continuous speech recognition problem into tasks suitable for SVMs.

We refer to this divide-and-conquer recognition strategy as *acoustic code-*

* Address for Correspondence: Department of Engineering, Cambridge University, Trumpington Street, Cambridge, CB2 1PZ, U.K.

Email addresses: veera@gmail.com (Veera Venkataramani), shantanu@egr.msu.edu (Shantanu Chakrabartty), wjb31@cam.ac.uk (William Byrne).

[1] This work was performed at the Center for Language and Speech Processing, The Johns Hopkins University, Baltimore, MD, USA. V. Venkataramani and W. Byrne were supported by the NSF (U.S.A) under the Information Technology Research (ITR) program, NSF IIS Award No. 0122466. S. Chakrabartty was supported by a grant from the Catalyst Foundation, New York.

breaking (Jelinek, 1996). The idea is first to perform an initial recognition pass with the best system available, which we take as based on Hidden Markov Models (HMMs); then to isolate and characterize regions of acoustic confusion encountered in the first-pass; and finally to apply models that are specially trained for these confusion problems. This provides a framework for using models that might not otherwise be appropriate for continuous speech recognition. It is also fundamentally an ASR rescoring approach. The goal is to apply SVMs to resolve the uncertainty that remains after the first-pass of the HMM-based recognizer.

We will build on prior work in the application of SVMs to continuous speech recognition. Smith et al. (2001) have developed *score-spaces* (Jaakkola and Haussler, 1998) 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 function 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. Smith and Gales (2002) demonstrate that these score-spaces can be used to obtain extra discriminatory information even though the scores are generated by the HMMs themselves; thus the SVMs trained on these score-spaces can improve upon the performance of the HMMs. However, the SVM is still essentially an isolated pattern classifier, so that this approach is still limited to the classification of variable length sequences as isolated binary classes.

To extend SVMs to continuous speech recognition, we set as the SVM training criterion the maximization of the posterior distribution over binary confusion sets found in the training set; in other words, we construct the SVM to lower the probability of error in training over continuous utterances. We will employ the *Gini*SVM (Chakrabartty and Cauwenberghs, 2002) which is an SVM variant that can be directly constructed to provide a posterior distribution over competing hypotheses with the goal of minimizing classification error. We note that a crucial step in the code-breaking procedure is the extraction of the training set used to train the SVMs, and we will show that some of the performance improvement obtained by code-breaking is directly attributable to this refinement of the training set.

In addition to selecting a hypothesis from each region of acoustic confusion, we use the SVMs to provide a posterior distribution over all the hypotheses in each confusion set. This will allow us to associate a measure of confidence with each SVM hypothesis. This is a valuable modeling tool and it allows us to perform hypothesis combinations (Fiscus, 1997) to produce results that improve over those of the individual HMM and SVM systems themselves.

To place our work in context, there have been previous applications of SVMs to speech recognition and we review some of the relevant prior work. Ganapathiraju et al. (2003) obtain a fixed dimension classification problem by using a heuristic method to normalize the durations of each variable length utterance. The distances to the decision boundary in feature space are then transformed into phone posteriors using sigmoidal non-linearities. Smith et al. (2001) use score-spaces to train SVMs followed by a majority voting scheme among binary SVMs to recognize isolated letters. Golowich and Sun (1998) 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 (Chakrabartty and Cauwenberghs, 2002) 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. Salomon et al. (2002) use a frame-by-frame classification approach and explore the use of the Kernel Fisher Discriminant for the application of SVMs for ASR.

In ASR, hypothesis combination is now well-established as a lattice analysis and processing technique (Fiscus, 1997). Mangu et al. (2000) developed methods to transform lattices into confusion networks which can be analyzed and rescored, for instance using rules based on word posteriors derived from the lattices (Mangu and Padmanabhan, 2001). Our approach differs from previous work in several respects. We use segment sets, an analogue of confusion networks, obtained by lattice-to-string alignment procedures (Goel et al., 2004; Kumar and Byrne, 2002) designed to identify regions of confusion in the original lattices while retaining the paths in the original lattice that form complete word sequences. We also apply our models in a hypothesis combination scheme, although we do so with models specially trained to resolve the confusions identified in the lattices and do not restrict ourselves to statistics derived from the underlying lattices. 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 (Lafferty et al., 2001).

Acoustic code-breaking was developed by Venkataramani and Byrne (2003) for small vocabulary tasks and subsequently applied to large vocabulary recognition tasks (Venkataramani and Byrne, 2005). That work forms the Ph.D. dissertation of Venkataramani (2005). Several other recent Ph.D. dissertations contribute directly to the modeling approach presented here. Lattice segmentation procedures described in the next section were developed by Kumar (2004) and subsequently used by Doumpiotis (2005) to develop the novel discriminative training procedures used in the baseline experiments of Section 6. *Gini*SVMs were developed by Chakrabartty (2004) and the use of SVMs with score-spaces derived from HMMs was studied originally by Smith (2003).

The rest of the paper is organized as follows: we first give a brief introduction to ASR and formulate it as a sequential classification problem. Next we discuss the application of SVMs for variable length observations and use the *Gini*SVMs to approximate a posterior distribution over hypotheses via logistic regression. We will then list the steps involved in implementing the new framework; this framework is evaluated in the experiments section. Following this we explore approaches to extend our work to large vocabulary tasks and conclude with final remarks.

2 Continuous Speech Recognition as a Sequence of Independent Classification Problems

The goal of a speech recognizer is to determine what word string W was spoken given an input acoustic signal O. The acoustic signal is represented as a T-length string of spectral measurements $O = o_1, o_2, \cdots, o_T$ and W by a string of N words given by $W = w_1, w_2, \cdots, w_N$.

In the usual manner, the hypothesis $\hat{W}$ is found by the *maximum a posteriori* (MAP) recognizer as

$$\hat{W} = \operatorname*{argmax}_{W \in \mathcal{W}} P(O|W)P(W) \tag{1}$$

where $\mathcal{W}$ represents the space of all possible word strings. To compute $P(O|W)$, we employ an *acoustic model*, usually an HMM. An HMM is defined by a finite state space $\{1, 2, \cdots, S\}$; an output space $\mathcal{O}$, usually R^d; transition probabilities between states $P(s_t = s'|s_{t-1} = s)$; and output distributions for states $P(o|s)$. For continuous output spaces, the output distribution of each HMM state is modeled as a multiple Gaussian mixture model

$$P(\mathbf{o}_t = o|\mathbf{s}_t = s) = \sum_{j=1}^{K} \frac{w_{i,s,j}}{(2\pi)^{D/2}|\Sigma_{i,s,j}|^{1/2}} \exp\left\{(o - \mu_{i,s,j})^\top \Sigma_{i,s,j}^{-1}(o - \mu_{i,s,j})\right\} \tag{2}$$

where K is the number of Gaussian components, $w_{i,s,j}$, $\mu_{i,s,j}$ and $\Sigma_{i,s,j}$ are the mixture weight, mean and co-variance matrix of the j^{th} component of the observation distribution of state s of the i^{th} word. The language model probability $P(W)$ appears in its usual role and assigns probability to word sequences $W = w_1, \ldots, w_N$.

In addition to producing the MAP hypothesis $\hat{W}$, the speech recognizer can also produce a set of likely hypotheses compactly represented by a lattice (see Fig. 1, a). Each link in the lattice represents a word hypothesis. Associated

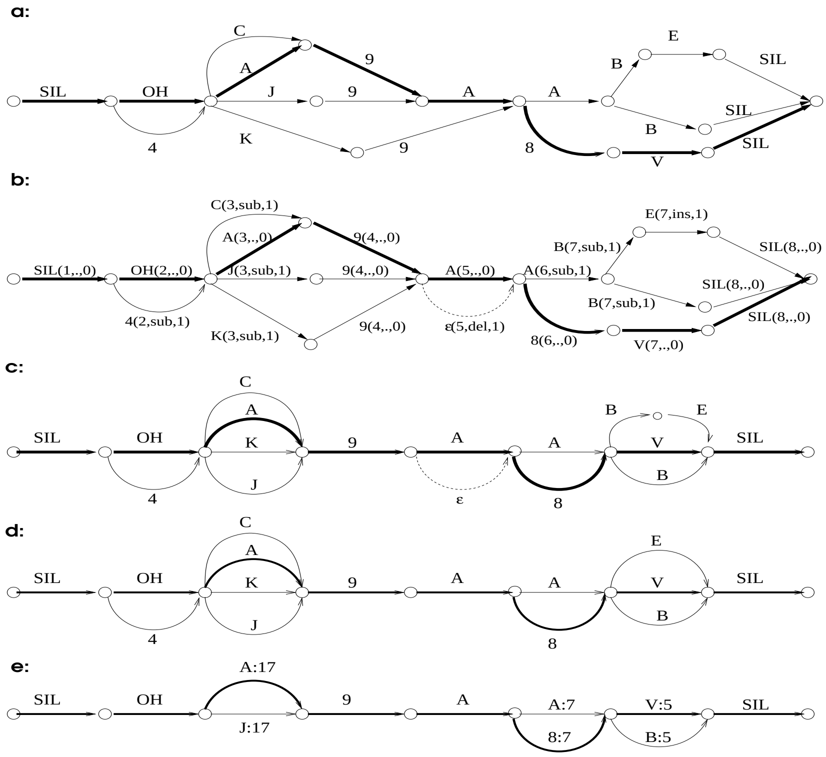

Fig. 1. Lattices and their segmentation. *a*: First-pass lattice of likely sentence hypotheses with a reference path (in bold); *b*: Alignment of lattice paths to the reference path with link labels indicating a word hypothesis, an alignment index, an edit operation and its cost; *c*: Alternate hypotheses for words in the reference hypotheses; *d*: Pruned segment sets; *e*: Search space consisting of binary segment sets with word hypotheses tagged to indicate membership in specific segment sets.

with each link are also the start and end times of the word hypothesis and the posterior probability of that word hypothesis relative to all the hypotheses in the lattice (Wessel et al., 1998). The N most likely hypotheses can also be generated from a lattice; such a list is called an N-best list.

2.1 The Sequential Problem Formulation

The MAP decoder as given in Eq. (1) assumes all word strings are of equal importance. The Minimum Bayes Risk (MBR) decoder (Goel and Byrne, 2000, 2003) attempts to address this issue by associating an empirical risk $E(W)$ with each candidate hypothesis W. Given a loss function $l(W, W')$ between two word strings W and W', *e.g.* the string-edit distance, $E(W)$ can be found as

$$E(W) = \sum_{W' \in \mathcal{W}} l(W, W') P(W'|O). \tag{3}$$

The goal of the MBR decoder is then to find the hypothesis with the minimum empirical risk as

$$\hat{W} = \underset{W \in \mathcal{W}}{\operatorname{argmin}} E(W). \tag{4}$$

It is not feasible to consider all possible hypotheses while computing $E(W)$. A possible solution is to approximate $\mathcal{W}$ by an N-best list. However for coverage and computational reasons we use lattices as our hypothesis space. Thus we find

$$E(W) = \sum_{W' \in \mathcal{L}} l(W, W') P(W'|O), \tag{5}$$

where $\mathcal{L}$ is a lattice for the utterance under consideration.

Given a string W, computation of Eq. (5) requires the alignment of every path in the lattice against W. Given the vast number of paths in a lattice, this cannot be done by enumeration. However, we have an efficient algorithm (Kumar and Byrne, 2002; Goel et al., 2004) that transforms the original lattice into a form (Fig. 1, b) that contains the information needed to find the best alignment of every word string to the reference string W.

Using the alignment we can then transform the original lattice into a form in which all paths in the lattice are represented as alternatives to the words in the reference string W (Fig. 1, c). This alignment identifies high confidence regions corresponding to the reference hypothesis as well as low confidence regions within which the lattice contains many alternatives. At this point we note that no paths have been removed; any path that was in the original lattice remains in the aligned lattice. Therefore we can use these segmented or *pinched* lattices for rescoring. This segmentation also leads to an *induced loss function* L_I between any two lattice paths, i.e. the alignment between the strings is constrained by the pinched lattice (Goel et al., 2004).

The particular form of lattice cutting shown in Fig. 1, c is referred to as period-1 lattice cutting (Goel et al., 2004); each word in the pinched lattice appears as an alternative for a single word in the reference hypothesis. In this cutting procedure we first discard alternatives that contain more than one word in succession; this gives groups of single word hypothesis (Fig. 1, d). We then apply likelihood-based pruning to reduce the number of alternatives to produce pairs of confusable words (Fig. 1, e). Each of these remaining word pairs is called a confusion pair.

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 start and end times. This pruning does reduce the search space; however alternatives to the reference hypothesis are available so that improvement is still possible.

2.2 MBR over Segmented Lattices

Let the original lattice be segmented into N sub-lattices, $\mathcal{W}_1, \mathcal{W}_2, \cdots, \mathcal{W}_N$. We can perform MBR decoding using the induced loss

$$\hat{W} = \underset{W' \in \mathcal{L}}{\operatorname{argmin}} \sum_{W \in \mathcal{L}} L_I(W, W') P(W|O), \tag{6}$$

which reduces (Goel et al., 2001; Goel and Byrne, 2003) to

$$\hat{W}_i = \underset{W' \epsilon \mathcal{W}_i}{\operatorname{argmin}} \sum_{W \epsilon \mathcal{W}_i} l(W, W') P_i(W|O) \tag{7}$$

where $\hat{W}_i$ is the minimum risk path in the ith sub-lattice and $\mathcal{W}_i$ represents all possible strings in the ith sub-lattice. The sentence-level MBR hypothesis is obtained as $\hat{W} = \hat{W}_1 \cdot \hat{W}_2 \cdots \hat{W}_M$ (Goel et al., 2004). Note that this formulation allows for the use of specially trained probability models $P_i(W|O)$ for each sub-lattice $\mathcal{W}_i$. We emphasize that while the hypothesis space $\mathcal{L}$ has been segmented, the observed acoustics $\mathbf{O}$ remain unsegmented. In the case of binary decision problems, each $\mathcal{W}_i$ that contains alternatives is reduced to a confusion pair $G_i = \{w_1, w_2\}$, where the subscripts indicate their classes. If $l(\cdot, \cdot)$ is taken to be the string-edit distance and $\delta(W, w)$ is the Kronecker delta function, Eq. (7) reduces to

$$\begin{aligned} \hat{W}_i &= \underset{W \epsilon G_i}{\operatorname{argmin}} \{P_i(w_1|O)\delta(W, w_2),\ P_i(w_2|O)\delta(W, w_1)\} & (8) \\ &= \underset{W \epsilon G_i}{\operatorname{argmax}} P_i(W|O), & (9) \end{aligned}$$

i.e., the sub-lattice $\mathcal{W}_i$ specific decoder chooses the word with the higher posterior probability. Note that in Eq. (8) the loss associated with a hypothesis is the posterior probability of its alternative. As can be seen in Fig. 1, e it often happens that in many cases the $\mathcal{W}_i$ contain only a single word. In these cases the word from the reference string is selected as the segment hypothesis.

In summary, lattice cutting converts ASR into a sequence of smaller, independent regions of acoustic confusion. Specialized decoders can then be trained for these decision problems and their individual outputs can be concatenated to obtain a new system output. We will next discuss Support Vector Machines and a formulation which allows them to be applied in this way.

3 Support Vector Machines for Variable Length Observations

We now briefly review the basic SVM (Vapnik, 1995). Let $\{\mathbf{x}^i\}_{i=1}^l$ be the training data and $\{y^i\}_{i=1}^l$ be the corresponding labels, where $\mathbf{x}^i \in \mathbf{R}^d$ and $y^i \in \{-1, +1\}$. Training an SVM involves maximizing a measure of the margin between the two classes or, equivalently, minimizing the following cost function

$$\frac{1}{2}\|\phi\|^2 - C\left[\sum_i 1 - y^i(\phi \cdot \zeta(\mathbf{x}^i) + \mathbf{b})\right]_+ \tag{10}$$

where $\|\phi\|^{-1}$ is the margin, C is the SVM trade-off parameter that determines how well the SVM fits the training data, ζ is the mapping from the input space ($\mathbf{R}^d$) to a higher dimensional feature space, $\mathbf{b}$ is the bias of the hyperplane separating the two classes, and $[\cdot]_+$ gives the positive part of the argument. This minimization is carried out using the technique of Lagrangian multipliers (Boser et al., 1992) which results in minimizing

$$\frac{1}{2}\sum_{i,j}\alpha_i \mathbf{K}(\mathbf{x}^i, \mathbf{x}^j)\alpha_j - \sum_i \alpha_i \tag{11}$$

subject to

$$\sum_i y^i\alpha_i = 0, \quad \text{and} \quad 0 \leq \alpha_i \leq C, \tag{12}$$

where α_i are the Lagrange multipliers and $\mathbf{K}(\cdot,\cdot)$ is the kernel function that computes an inner product in the higher dimensional feature space $\zeta(\cdot)$ (Cortes and Vapnik, 1995). New observations $\mathbf{x}$ are classified using the decision rule

$$\hat{y} = \operatorname{sgn}\left(\sum_i y^i\alpha_i \mathbf{K}(\mathbf{x}, \mathbf{x}^i) + \mathbf{b}\right). \tag{13}$$

3.1 Feature Spaces

SVMs are static classifiers; a data sample to be classified must belong to the input space ($\mathbf{R}^d$). However, speech utterances vary in length. To be able to use SVMs for speech recognition we need some method to transform variable length sequences into vectors of fixed dimension. Towards this end we would also like to use the HMMs that we have trained so that some of the advantages of the generative models can be used along with the discriminatively trained models.

Fisher scores (Jaakkola and Haussler, 1998) 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 (Smith et al., 2001). Each component of the Fisher score is defined as the sensitivity of the likelihood of the observed sequence to each parameter of an HMM. Since the HMMs have a fixed number of parameters, this yields a fixed dimension feature even for variable length observations. Smith et al. (2001) have extended Fisher scores to score-spaces in the case when there are two competing HMMs. This formulation has the added benefit that the features provided to the SVM can be derived from a well-trained HMM recognizer. For a complete treatment of score-spaces, see the work of Smith and Gales (2002).

For discriminative binary classification problems the log likelihood-ratio score-space has been found to perform best among a variety of possible score-spaces. If we have two HMMs with parameters θ_1 and θ_2 and corresponding likelihoods $p_1(\mathbf{O};\theta_1)$ and $p_2(\mathbf{O};\theta_2)$, the projection of an observation sequence ($\mathbf{O}$) into the log likelihood-ratio score-space is given by

$$\varphi(\mathbf{O};\theta) = \begin{bmatrix} \varphi_0(\mathbf{O};\theta) \\ \varphi_1(\mathbf{O};\theta_1) \\ -\varphi_2(\mathbf{O};\theta_2) \end{bmatrix} = \begin{bmatrix} \log \frac{p_1(\mathbf{O};\theta_1)}{p_2(\mathbf{O};\theta_2)} \\ \nabla_{\theta_1} \log p_1(\mathbf{O};\theta_1) \\ -\nabla_{\theta_2} \log p_2(\mathbf{O};\theta_2) \end{bmatrix} \tag{14}$$

where $\theta = [\theta_1 \;\; \theta_2]$.

In our experiments we derive the score space solely from the means of the multiple-mixture Gaussian HMM state observation distributions, denoted via the shorthand $\theta_i[s,j,k] = \mu_{i,s,j}[k]$, where k denotes a component of a vector; the omission of the Gaussian variance parameters will be discussed in Section 6. We first define the parameters of the j^{th} Gaussian observation distribution associated with state s in HMM i as $(\mu_{i,s,j}, \Sigma_{i,s,j})$. The gradient with respect to these parameters (Smith et al., 2001) is

$$\nabla_{\mu_{i,s,j}} \log\, p_i(\mathbf{O};\theta_i) = \sum_{t=1}^{T} \gamma_{i,s,j}(t) \Big[(o_t - \mu_{i,s,j})^\top \Sigma_{i,s,j}^{-1} \Big]^\top, \tag{15}$$

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. As these scores are accumulated over the individual observations, they must be normalized for the sequence length (T). We mention two such schemes in Section 4.1.

SMBR decoding over binary classes requires estimation of the posterior distribution $P(W|\mathbf{O})$ (Eq. (9)) over binary segment sets $G = \{w_1, w_2\}$. To interpret the application of SVMs to classification within the segment sets, we will first recast this posterior calculation as a problem in logistic regression. Our approach follows the general approach of Jaakkola and Haussler (1998).

If we have binary problems with HMMs as described in the previous section, the posterior can be found by first computing the quantities $p_1(\mathbf{O};\theta_1)$ and $p_2(\mathbf{O};\theta_2)$ so that

$$P(w_j|\mathbf{O};\theta) = \frac{p_j(\mathbf{O};\theta_j)P(w_j)}{p_1(\mathbf{O};\theta_1)P(w_1) + p_2(\mathbf{O};\theta_2)P(w_2)} \quad j = 1, 2\ . \tag{16}$$

This distribution over the binary hypotheses can be rewritten as

$$P(w|\mathbf{O};\theta) = \frac{1}{1 + \exp[k(w)\log\frac{p_1(\mathbf{O};\theta_1)}{p_2(\mathbf{O};\theta_2)} + k(w)\log\frac{P(w_1)}{P(w_2)}\ ]} \tag{17}$$

$$\text{where} \quad k(w) = \begin{cases} -1 & w = w_1 \\ +1 & w = w_2 \end{cases}\ .$$

If a set of HMM parameters $\bar{\theta}$ is available, the posterior distribution can be found by first evaluating the likelihood ratio $\log\frac{p_1(\mathbf{O};\bar{\theta}_1)}{p_2(\mathbf{O};\bar{\theta}_2)}$ and inserting the result into Eq. (17). If a new set of parameter values becomes available, the same approach could be used to reestimate the posterior. Alternatively, the likelihood ratio could be considered simply as a continuous function in θ whose value could be found by a Taylor Series expansion around $\bar{\theta}$

$$\log\frac{p_1(\mathbf{O};\theta_1)}{p_2(\mathbf{O};\theta_2)} = \log\frac{p_1(\mathbf{O};\bar{\theta}_1)}{p_2(\mathbf{O};\bar{\theta}_2)} + (\theta - \bar{\theta})\ \nabla_\theta \log\frac{p_1(\mathbf{O};\bar{\theta}_1)}{p_2(\mathbf{O};\bar{\theta}_2)} + \ \cdots \tag{18}$$

which of course is only valid for $\theta \approx \bar{\theta}$.

If we ignore the higher order terms in this expansion and gather the statistics into a vector

$$\Psi(\mathbf{O};\bar{\theta}) = \begin{bmatrix} \varphi_0(\mathbf{O};\bar{\theta}) \\ \varphi_1(\mathbf{O};\bar{\theta}_1) \\ -\varphi_2(\mathbf{O};\bar{\theta}_2) \\ 1 \end{bmatrix} \tag{19}$$

we obtain the following approximation for the posterior at θ

$$P(w|\mathbf{O};\theta) \approx \frac{1}{1+\exp[\ k(w)\ [1 \quad (\theta-\bar{\theta}) \quad \log\frac{P(w_1)}{P(w_2)}]\ \ \Psi(\mathbf{O};\bar{\theta})\ ]}\ . \tag{20}$$

We will realize this quantity by the logistic regression function

$$P_a(w|\mathbf{O};\phi) = \frac{1}{1+\exp[\ k(w)\ \phi^\top\ \Psi(\mathbf{O};\bar{\theta})\ ]} \tag{21}$$

and Eq. (20) is realized exactly if we set

$$\phi \quad = \quad \begin{bmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} \quad = \quad \begin{bmatrix} 1 \\ \theta_1 - \bar{\theta}_1 \\ \theta_2 - \bar{\theta}_2 \\ \log\frac{P(w_1)}{P(w_2)} \end{bmatrix} \ . \tag{22}$$

Our goal is to use estimation procedures developed for large margin classifiers to estimate the parameters of Eq. (21) and in this we will allow ϕ to vary freely. This has various implications for our modeling assumptions. If we allow ϕ_3 to vary, this is equivalent to computing P_a under a different prior distribution than initially specified. If ϕ_1 or ϕ_2 vary, we allow the parameters of the HMMs to vary from their nominal values $\bar{\theta}_1$ and $\bar{\theta}_2$. This might produce parameter values that lead to invalid models, although we restrict ourselves here to the means of the Gaussian observation distributions which can be varied freely. Variations in ϕ_0 are harder to interpret in terms of the original posterior distribution derived from the HMMs; despite that, we still allow this parameter to vary.

Taking the form of Eq. (21), we assume that we have a labeled training set $\{\bar{\mathbf{O}}^j, \bar{w}^j\}_j$ and that we wish to refine the distribution P_a over the data according to the following objective function

$$\min_{\phi} \ \frac{1}{2}\|\phi\|^2 - C\sum_j \log P_a(\bar{w}^j|\bar{\mathbf{O}}^j;\phi) \ , \tag{23}$$

where C is a trade-off parameter that determines how well P_a fits the training data. The role of the regularization term $\|\phi\|^2$ is to penalize HMM parameter estimates that vary too far from their initial values $\bar{\theta}$. Similarly, it allows reestimation of the prior over the hypotheses, but prefers estimates that assign comparable likelihood to hypotheses, i.e. estimates for which $P(w_1) \approx P(w_2)$. This could be easily modified to incorporate prior information about which choice is more likely.

If we define a binary valued indicator function over the training data

$$y^j = \begin{cases} +1 & w^j = w_1 \\ -1 & w^j = w_2 \end{cases}$$

we can use the approximation techniques of Chakrabartty and Cauwenberghs (2002) to minimize Eq. (23) where the dual is given by

$$\frac{1}{2}\sum_{i,j}\alpha_i \ [\ \mathbf{K}(\ \Psi(\mathbf{O}^i;\bar{\theta}),\ \Psi(\mathbf{O}^j;\bar{\theta})\ ) + \frac{2\gamma}{C}\delta_{ij}\ ]\ \alpha_j - 2\gamma\sum_i \alpha_i \tag{24}$$

subject to

$$\sum_i y^i\alpha_i = 0, \quad 0 \le \alpha_i \le C, \tag{25}$$

where γ is the rate distortion factor chosen as $2\log 2$ in the case of binary classes and δ_{ij} is the Kronecker delta function. The optimization can be carried out using the *Gini*SVM Toolkit which is available online (Chakrabartty, 2003).

After the optimal parameters α are found, the posterior distribution of an observation is found as

$$P_a(w|\mathbf{O};\phi) = \frac{1}{1+\exp[\ k(w)\ \phi^\top\ \zeta(\Psi(\mathbf{O};\bar{\theta}))]} \tag{26}$$

$$= \frac{1}{1 + \exp[\ k(w) \sum_i y^i\ \alpha_i\ \mathbf{K}(\ \Psi(\mathbf{O}^i; \bar{\theta}),\ \Psi(\mathbf{O}; \bar{\theta})\ )]}\ , \qquad (27)$$

and ϕ can be written as $\phi = \sum_i \alpha_i\ y^i\ \zeta(\Psi(\mathbf{O}^i; \bar{\theta}))$.

Using *Gini*SVM in this way allows us to estimate the posterior distribution under penalized likelihood criterion of Eq. (23). The distribution that results can be used directly in the classification of new observations with the added benefit that the form of the distribution in Eq. (27) makes it easy to assign 'confidence scores' to hypotheses. This will be useful in the weighted hypothesis combination rescoring procedures that will be described subsequently.

We note in practice for purposes of classification, the score space requires some form of normalization, the use of a non-linear kernel and the estimation of ϕ_0; the regression SVM will then not realize Eq. 21 exactly. Some of these modeling issues are discussed in Sections 4 and 6. Therefore under certain conditions the SVM regression can be viewed as posterior probability estimation under the HMM likelihood model.

4 Modeling Issues

4.1 Estimation of sufficient statistics

We wish to apply SVMs to word hypotheses in continuous speech recognition where the start and end times of word hypotheses are uncertain. One possibility is to take the timing information from the first pass ASR output. An alternative approach can be seen in the example in Fig. 1, e. Consider the confusion pair A:17 *vs.* J:17. We can compute the statistics for this pair by performing two forward-backward calculations with respect to the transcriptions

SIL OH A:17 NINE A EIGHT B SIL

SIL OH J:17 NINE A EIGHT V SIL

where A:17 and J:17 are cloned versions of models A and J respectively.

When we perform forward-backward calculations over the entire utterance to calculate statistics for a particular confusion pair, it is also possible to consider alternative paths that arise due to other confusion segments. For instance, for the confusion pair B:5 *vs.* V:5 in Fig. 1, e, considering the neighbouring segments would imply gathering statistics over the following four hypotheses:

SIL OH A NINE A EIGHT B:5 SIL

SIL OH A NINE A EIGHT V:5 SIL

SIL OH A NINE A A B:5 SIL

SIL OH A NINE A A V:5 SIL

We mention this scheme of using the alternatives in neighbouring segments as an option; in our experiments we used the simpler case.

Either the Viterbi or the Baum-Welch algorithm can be used to compute the mixture-level posteriors of Eq. (15). As discussed by Smith and Gales (2002), these scores must be normalized to account for individual variations in sequence length. If time segmentations of the utterance at the word level are available, one possibility is simply to normalize each score by the length of its word (T). Alternatively, the sum of the state occupancy over the entire utterance may be used, *i.e.*, $\sum_{t=1}^{T} \gamma_s(t)$, where s is the state index.

4.2 Normalization

While a linear classifier can subsume a bias in the training, the parameter search (α_i in Eq. 24) can be made more effective by ensuring that the training data is normalized. We first adjust the scores for each acoustic segment via mean and variance normalization. The normalized scores are given by

$$\varphi^N(\mathbf{O}) = \hat{\Sigma}_{sc}^{-1/2}[\varphi(\mathbf{O}) - \hat{\mu}_{sc}], \tag{28}$$

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 incorporate the $\hat{\mu}_{sc}$ bias and the variance normalization would be performed by the scaling matrix $\hat{\Sigma}_{sc}$ as

$$\varphi^N(\mathbf{O}) = \hat{\Sigma}_{sc}^{-1/2}\varphi(\mathbf{O}) \tag{29}$$

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

$$\hat{\Sigma}_{sc} = \frac{1}{M-1}\sum(\varphi(\mathbf{O}) - \hat{\mu}_{sc})^\top(\varphi(\mathbf{O}) - \hat{\mu}_{sc}) \tag{30}$$

where $\hat{\mu}_{sc} = \frac{1}{M}\sum\varphi(\mathbf{O})$, and M 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 segment are normalized by the segment length T.

4.3 Dimensionality Reduction

For efficiency and modeling robustness there may be value in reducing the dimensionality of the score-space. There has been research (Blum and Langley, 1997; Smith and Gales, 2002) 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 (Smith and Gales, 2002)

$$g[d] = \frac{|\hat{\mu}_{sc[1]}[d] - \hat{\mu}_{sc[2]}[d]|}{\hat{\Sigma}_{sc[1]}[d] + \hat{\Sigma}_{sc[2]}[d]} \tag{31}$$

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 $g[d]$. We note that the dimensionality of the feature space is large enough that the computation of decorrelating transformations would be numerically difficult. Dimensionality reduction by pruning is a practical approach this modeling problem.

4.4 GiniSVM and its Kernels

*Gini*SVMs have the advantage that, unlike regular SVMs, they can employ non positive-definite kernels. 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 (Smith and Gales, 2002). 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, $G(1)$ and $G(2)$. Ideally if $\mathbf{x}, \mathbf{y} \in G(1)$ and $\mathbf{z} \in G(2)$, 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.

We also found that the *Gini*SVM classification performance was sensitive to the SVM trade-off parameter C; this is in contrast to earlier work on other tasks (Smith et al., 2001). 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.

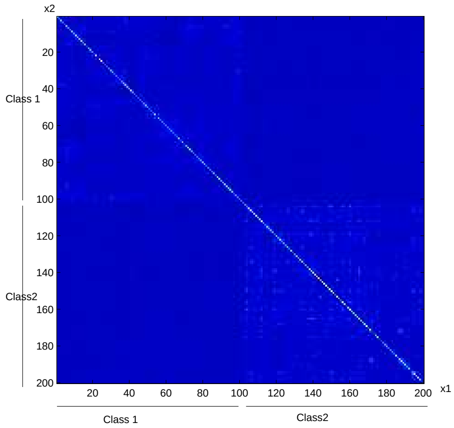

Fig. 2. Kernel Map $\mathbf{K}(\ \Psi(\mathbf{O}^i;\bar{\theta}),\ \Psi(\mathbf{O}^j;\bar{\theta})\ )$ for the linear kernel over two class data.

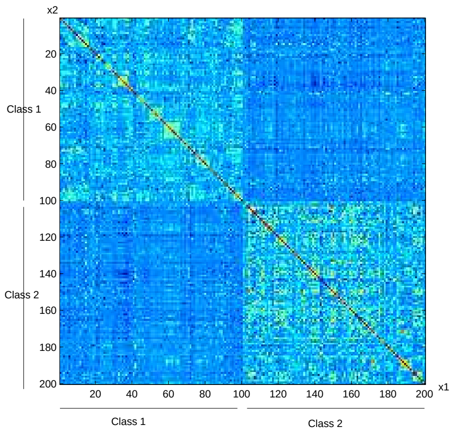

Fig. 3. Kernel Map $\mathbf{K}(\ \Psi(\mathbf{O}^i;\bar{\theta}),\ \Psi(\mathbf{O}^j;\bar{\theta})\ )$ for tanh kernel over two class data.

5 The SMBR-SVM framework

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

5.1 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 paths in the lattices are then aligned against the reference transcriptions (Goel et al., 2004). 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.

5.2 Training SVMs for each confusion pair

For each acoustic segment in every sub-lattice, likelihood-ratio scores as given by Eq. (14) 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 (Eq. (31)) with appropriate thresholds. SVMs for each confusion pair are then trained in our normalized score-space using the appropriate acoustic segments identified as above.

5.3 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 (found by Eq. (27)) in the regions of low confidence. This is the final hypothesis of the SMBR-SVM system.

5.4 Posterior-based System Combination

We now have the HMM and the SMBR-SVM system hypotheses along with their posterior estimates. If these posterior estimates serve as reliable confidence measures, we can combine the system hypotheses to yield better performance. We use two simple schemes, either

$$\hat{p}_+(w_i) = \frac{p_h(w_i) + p_s(w_i)}{2}, \tag{32}$$

or

$$\hat{p}_\times(w_i) = \frac{p_h(w_i)p_s(w_i)}{p_h(w_1)p_s(w_1) + p_h(w_2)p_s(w_2)}. \tag{33}$$

where $p_h(w_1)$ and $p_h(w_2)$ are the posterior estimates of the two competing words in a segment as estimated by the HMM system and $p_s(w_1)$ and $p_s(w_2)$ are those of the SMBR-SVM system. These schemes then pick the word with the higher value. In our experiments, we used the p_+ combination scheme. For these simple binary problems, many voting procedures yield identical results and the actual form is not crucial.

5.5 Rationale

The most ambitious formulation of acoustic code-breaking is first to 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 (Doumpiotis et al., 2003a) has verified that training set errors selected in this way are good predictors of the errors that will be encountered in unseen test data.

6 Acoustic Code-Breaking with HMMs and SMBR SVMs

We evaluated this modeling approach on the OGI-Alphadigits corpus (Noel, 1997). This is a small vocabulary task that is fairly challenging. The baseline Word Error Rates (WERs) for Maximum Likelihood (ML) models are approximately 10%; at this error rate there are there are enough errors to support detailed analysis. The task has a vocabulary of 36 words (26 letters and 10 digits), and 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. (2003b).

Whole-word HMMs were trained for each of the 36 words. The models had left-to-right topology with approximately 20 states each and 12 mixtures per state. The data were parametrized as 13 dimensional MFCC vectors with first and second order differences. The baseline ML models were trained following the HTK-book (Young et al., 2000). The AT&T decoder (Mohri et al., 2001) 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. The ML baseline WER was 10.73% (Table 1 System A). MMI training was then performed (Normandin, 2002; Woodland and Povey, 2000) 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 resulting WER was 9.07% (Table 1 System D) and the lattice oracle error rate for these lattices was 1.70%. Period-1 lattice cutting was then performed on these lattices, and the number of confusable words in each segment was further restricted to two. This increased the lattice oracle error rate to 4.07%; the number of hypothesized words in the lattices decreased from ∼768000 to ∼11500.

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 kept the 50 confusion pairs observed most frequently in the training data. All other confusion pairs in training and test data were pruned back to the truth and the MAP hypothesis respectively. We emphasize that this is a fair process; the truth is not used in identifying confusion in the test data.

Doumpiotis et al. (2003b) have also found that performing further MMI training of the baseline MMI models on the pinched lattices yields additional improvements. The performance of this Pinched Lattice MMI (PLMMI) system is listed in Table 1 as System E. We see a reduction in WER over the MMI models from 9.07% to 7.98%.

6.1 The Role of Training Set Refinement in Code-Breaking

We have proposed a technique that first identifies errors, then selects training data associated with each type of error, and finally applies models trained to fix those errors. We will show that using SVMs in this way improves over recognition with HMMs; however some of this improvement maybe due simply to training on these selected subsets.

We investigated the effect of retraining on the confusable data in the training set. Specifically, we performed supervised Baum-Welch re-estimation of the whole-word HMMs over the time bounded segments of the training data associated with all the error classes. The confusion sets and their time boundaries from the ML system were available for both training and test data; therefore

Table 1
HMM and SMBR-SVM System Performance. A: Baseline HMMs trained under the ML criterion; B: System A HMMs with further Baum-Welch estimation performed over confusable segments; C: HMMs from System A cloned and tagged as in Fig. 1, e with Baum-Welch estimation performed over confusable segments; D: System A HMMs refined by MMI; E: System B HMMs refined by MMI over pinched lattices (PLMMI). Three different search procedures are evaluated: MAP (Eq. 1); SMBR-SVM segment rescoring; and MAP and SMBR-SVM hypothesis combination ('Voting'). Performance is measured in Word Error Rate (%).

	HMM Training	Segmented	Cloned	Decoding Procedure		
System	Criterion	Data	HMMs	MAP	SMBR-SVM	Voting
A	ML	N	N	10.73	8.63	8.24
B	ML	Y	N	10.00	-	-
C	ML	Y	Y	10.30	-	-
D	MMI	N	N	9.07	8.10	7.76
E	PLMMI	N	N	7.98	8.13	7.16

these results are directly comparable to the ML baseline (Table 1, System A). Simply by refining the training set in this way we found a reduction in WER from 10.73% to 10.00% (Table 1, System B). We conclude that significant gains can be obtained simply by retraining the ML system on the confusable segments identified in the training set.

We next considered ML training of a set of HMMs for each of the error classes. This is the most basic approach to Code-Breaking: we clone the ML-baseline models and retrain them over the time-bounded segments of the training data associated with each error class. Since there are 50 binary error classes, this adds 100 tagged models to the baseline model set. The results of rescoring with these models are given in Table 1, System C. We see a reduction in WER from the 10.73% baseline to 10.30%, however these error-specific models do not perform as well as a single set of models trained over the refined training set (10.00% WER). Given that the single set of models can be trained to good effect, there is clearly a risk of data fragmentation in this type of training set refinement. Moreover, as the WER of the baseline system decreases, the number of confusion sets naturally decreases, as well: there were ∼120000 confusion pairs identified in the training set by the MMI System D, and that number drops to ∼80000 under the PLMMI System E. This effect has been observed before and robust discriminative estimation techniques are available to improve HMMs cloned in this way (Doumpiotis et al., 2003a,b). This experiment demonstrates that effective use of refined training sets requires both novel model architecture and novel estimation procedures.

6.2 SMBR-SVM Systems

The *Gini*SVM Toolkit (Chakrabartty, 2003) was used to train SVMs for the 50 dominant confusion pairs extracted from the lattices generated by the MMI system. The word time boundaries of the training samples were extracted from the lattices. The statistics needed for the SVM computation were found using the forward-backward procedure over these segments; in particular the mixture posteriors of the HMM observation distributions were found in this way. Log-likelihood ratio scores were generated from the 12 mixture MMI models and normalized by the segment length as described in Section 4.1.

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 Eq. (31) made training feasible, however performance was not much improved. One possible explanation is that there is significant dependence between the model means and variances which violates the underlying assumptions of the goodness criterion used in filtering. We then used only the filtered mean sub-space scores for training SVMs (training on the unfiltered mean sub-space remained impractical because of the prohibitively high number of dimensions). The best performing SVMs used around 2,000 of the most informative dimensions, which was approximately 10% of the complete mean space.

As shown in Table 1, applying SMBR-SVM yielded improvements relative to MAP decoding for both the ML-trained system (System A) and the MMI-trained system (System D). For System A, SMBR-SVM reduced the WER from 10.73% to 8.63%, while for System D the reduction was from 9.07% to 8.10%. Building an SMBR-SVM from the MMI-trained system is a significant improvement relative to the ML-trained system (8.63% vs. 8.10%). However, in System E the SVM system does not yield improved performance relative to the PLMMI HMM baseline and in fact performance degrades slightly when used in straightforward SMBR-SVM decoding (8.13% vs. 7.98%).

6.3 Posterior-based system combination

In comparing the MMI and SMBR-SVM hypotheses to each other, we observed that they differ by more than 4% in WER; this has been observed in some but not all previous work (Fine et al., 2001; Golowich and Sun, 1998; Smith et al., 2001). This suggests that hypothesis selection can produce an output better than each of the individual outputs. Ideally the voting schemes will be based on posterior estimates provided by each system. Transforming HMM acoustic likelihoods into posteriors is well established (Wessel et al., 1998). In various experiments not reported here, the quality of the posteriors under the SMBR-SVM system was found to be comparable to that of the HMM system as measured by Normalized Cross-Entropy (Fiscus, 1997).

The recognition performance of hypothesis combination schemes (Section 5.4) using the SMBR-SVM posterior and the HMM-based posterior are presented in the 'Voting' column of Table 1. In all systems the Voting procedure gives substantial improvement relative to the MAP and to the SMBR-SVM performance alone. Notably in System E, Voting between the PLMMI system and the SVM system reduces the MAP hypothesis WER from 7.98% to 7.16% even though the SMBR-SVM result alone was slightly worse than the MAP result. The code-breaking modeling procedure clearly produces complimentary systems suitable for hypothesis combination. It is also interesting to note that both the sum and the product voting schemes yielded the same output even to the level of individual word hypotheses.

6.4 PLMMI SMBR-SVM Tuning

All SMBR-SVM experiments reported thus far employ a fixed global trade-off parameter value for the SVMs trained for the confusion pairs. This is a fair baseline for developing novel techniques, but may not be optimal since the confusion sets will vary in difficulty, number of samples, and other factors which might affect the optimal value of C. Therefore the effect of the SVM trade-off parameter (C in Eq. (25)) on a SMBR-SVM system was studied. The specific system studied was a PLMMI SMBR-SVM system (Venkataramani and Byrne, 2003) that used word time boundaries from MMI lattices. Note that while this is a different system from Table 1, System E, the performance of the PLMMI HMM baseline (7.98% WER) remains unchanged. WER results from training the SVMs for the confusion pairs at different values of C are presented in Fig. 3. We find some sensitivity to C, however optimal performance was found over a fairly broad range of values (0.3 to 1.0).

We also investigated tuning of individual trade-off parameter values for each SVM with results presented in Table 3. The oracle result is obtained by 'cheating' through choosing the parameter for each SVM that yielded the lowest class error rate. Choosing C by this oracle reduced the WER from 8.01% to 7.77% suggesting that variations in the trade-off parameter are worth exploring. A fair systematic rule for choosing the parameter based on the number of training examples is presented in Table 2. By following this rule we almost matched the oracle performance (7.88% vs. 7.77%). We note also that this unsupervised tuning procedure matches the best PLMMI HMM system of 7.98% (Table 1

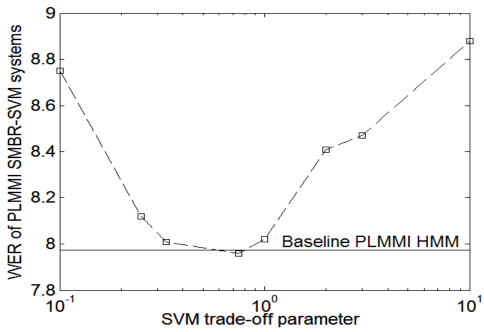

Fig. 4. PLMMI SMBR-SVM performance as a function of the SVM trade-off parameter C.

Table 2
Piecewise Rule for choosing the trade-off parameter (C) through the number of training observations (N).

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
PLMMI SMBR-SVM performance with tuning of the SVM trade-off parameter C.

	SMBR-SVM
$C = 1$	8.01
Oracle C	7.77
Piecewise C	7.88

System E). Although C was originally introduced to control the sensitivity of the model to the data, we believe there are other factors, such as task complexity or redundancy in the training material, that explain why the mapping given in Table 2 is effective for this task. For instance, an SVM training set containing many HMM score samples with consistent discriminatory information would require a lowering of the value of C in Equation 23; the kernel map of Fig. 3 suggests that such behavior may indeed be a factor.

7 SVM Score-Spaces Through Constrained Parameter Estimation

We have studied a simple ASR task so that we could develop the SMBR-SVM modeling framework and describe it without complication. Our ultimate goal is to apply this framework to large vocabulary speech recognition systems which are usually built on sub-word acoustic models shared across words. Large vocabulary systems typically consist of sub-word models that are shared across words. We could apply the approach we have described thus far in a brute force manner by cloning the models in a large vocabulary HMM system, and retraining them over confusion sets, and deriving SVM statistics from the models and the confusion sets.

Apart from the unwieldy size of a cloned system, the main problem would be data sparsity in calculating statistics for SVM training. This observation suggests the use of models obtained via constrained estimation. We can use linear transforms (LTs) such as Maximum Likelihood Linear Regression (MLLR) (Legetter and Woodland, 1995) to estimate model parameters. Following the approach we have developed, these transforms are estimated over segments in the acoustic training set that were confused by the baseline system. We emphasize that these LTs are not used as a method of adaptation to test set data.

Consider the case of distinguishing between two words w_1 and w_2 in a large vocabulary system. We need to construct word models θ_1 and θ_2 from sub-word acoustic models and we will use the two word models to find the statistics needed to train an SVM. We identify all instances of this confusion pair in the training set and use this data to estimate two transforms L_1 and L_2 relative to the baseline HMM system. These are trained via supervised adaptation, *e.g.* by MLLR over the refined training set. One approach to deriving an LT score-space is to rely directly on the parameters of the transform

$$\varphi(\mathbf{O}) = \begin{bmatrix} 1 \\ \nabla_{L_1} \\ \nabla_{L_2} \end{bmatrix} \log\left(\frac{p_1(\mathbf{O}|L_1 \cdot \theta_1)}{p_2(\mathbf{O}|L_2 \cdot \theta_2)}\right). \tag{34}$$

However in experiments not reported here the score-space of Eq. (34) proved unsuitable for classification. When we inspected the kernel maps, we saw no evidence of the block diagonal structure characteristic of features useful for pattern classification. Since the linear transforms only provide a direction in the HMM parameter manifold, it is possible that they do not provide enough information for the SMBR-SVM system to build effective decision boundaries in the LT score-space.

An alternative is to create a constrained score-space by applying MLLR transforms to a set of original models to derive a new set of models. The score-space is the original mean score-space but is derived from the adapted HMMs. If $\theta_1' = L_1 \cdot \theta_1$ and $\theta_2' = L_2 \cdot \theta_2$ the constrained score space is

$$\varphi(\mathbf{O}) = \begin{bmatrix} 1 \\ \nabla_{\theta} \end{bmatrix} \log\left(\frac{p_1(\mathbf{O}|\theta_1')}{p_2(\mathbf{O}|\theta_2')}\right). \tag{35}$$

Table 4
HMM and SMBR-SVM System Performance. SMBR-SVM systems were trained in the score-space of the transformed models

System	HMM Training Criterion	Decoding Procedure	
		MAP	SMBR-SVM
D	MMI	9.07	8.10
F	MMI+MLLR	9.35	8.00

Although intended for large vocabulary recognition tasks, we investigated the feasibility of the approach in our small vocabulary experiments. The results are tabulated in Table 4. We estimated MLLR transforms with respect to the MMI models over the confusion sets. A single transform was estimated for each word hypothesis in each confidence set. We then applied the transforms to the MMI models to estimate statistics as described in Eq. (35). The performance is shown in Table 4, System F. We see a reduction in WER with respect to the MMI baseline from 9.07% to 8.00%. In comparing this result to that of the SVMs derived from the MMI models (8.00% vs. 8.10%), we conclude that this severely constrained estimation is able to generate score-spaces that perform similarly to those score spaces derived by unconstrained estimation. For completeness, we rescored the confusions sets using the ML-transformed MMI models. As can be expected performance degrades slightly from 9.07% to 9.35%, suggesting that performing ML estimation subsequent to MMI estimation undoes the effects of discriminative training, as has been previously reported (Normandin, 1995).

8 Conclusions

We have developed a Code-Breaking framework that applies Support Vector Machines in continuous speech recognition. We use available baseline HMM models for the identification of confusable regions, train error specific SVMs, and attempt to resolve the remaining confusion in the test data using the error specific models.

Our framework uses lattice cutting techniques to convert the continuous ASR problem into a sequence of independent but coupled classification problems. We used the previously proposed technique of score-spaces to convert the variable length acoustic sequences associated with the problems into fixed dimensional vectors which can then be classified by SVMs.

We posed the estimation of a posterior distribution over hypothesis in the confusable regions as a logistic regression problem. We showed that *Gini*SVMs can be used as an approximation technique to estimate the parameters of the logistic regression problem. We also found significant improvements by using tanh kernels over other kernels that have been studied for ASR. We conjecture that this is due to the ability of *Gini*SVMs to incorporate non-positive-definite kernels in its training.

We investigated several methods to compute the sufficient statistics required to generate scores. While the approaches performed similarly on the problems we study, we noted different implementation aspects of their implementation that may make them more appropriate choices for large vocabulary recognition tasks. We see considerable improvement in the performance of SVMs through selection of the most informative score-space dimensions, as has been noted (Smith and Gales, 2002). We suspect this to be an artifact of the approximation to the scaling matrix. If improved normalization of the score-space is found either through better numerical methods or an improved modeling formulation, the SMBR-SVM formulation should be expected to yield further improvements.

We find that confidence measures over hypotheses can be robustly produced by *Gini*SVMs. This allows for hypothesis selection from the baseline and the SVM system using a weighted voting scheme. We further found that SMBR-SVM rescoring performed significantly better than MMI and using the voting schemes we obtained significant improvements over another form of discriminative training, namely PLMMI.

We have identified two components to the gains that we find in this use of SVMs. The first contribution comes from the refinement of the training data. The baseline models themselves can be improved by training over confusable data identified by lattice cutting. The second contribution comes from the use of SVMs themselves.

Our ultimate goal is to apply our new framework to large vocabulary continuous speech recognition. We have discussed some of the problems we expect to encounter and have proposed and investigated constrained estimation techniques that will allow us to derive features for SVMs when training data is scarce. Initial experiments in applying this modeling approach to a large vocabulary speech recognition task have been performed (Venkataramani and Byrne, 2005). We have found that the techniques described in this paper are very effective at resolving binary confusions in large vocabulary recognition, however the overall impact on word error rate is necessarily limited. Byrne (2006) discusses these issues in detail.

In this work, we were able to ignore the effects of the language model due to the nature of the small vocabulary task studied. Since only acoustic models were involved in decoding, the only deficiencies that were present in the decoder derived from the acoustic model. However, in large vocabulary continuous speech recognition, the role of the language model must also be considered. Possible approaches have been discussed in the disseration of Venkataramani (2005).

We have introduced a new framework that incorporates the benefits of HMMs and improves upon their performance. The promise of this framework is that it allows us to explore the application of new modeling techniques to continuous speech recognition without having to address all aspects of that large and complex problem.

Acknowledgments We would like to thank Prof. G. Cauwenberghs for helpful suggestions. Baseline MMI and PLMMI models were trained by V. Doumpiotis and the ML models were trained by T. Kamm. We thank AT&T for use of the AT&T Large Vocabulary Decoder. V. Venkataramani thanks S. Kumar, P. Xu, and Y. Deng for helpful discussions. The authors also thank the anonymous reviewers for their comments.

References

Blum, A., Langley, P., 1997. Selection of relevant features and examples in machine learning. Artificial Intelligence 97 (1-2), 245–271.

Boser, B., Guyon, I., Vapnik, V., 1992. A training algorithm for optimal margin classifier. In: Proc. 16th Conf. Computational Learning Theory. pp. 144–152.

Burges, C. J. C., 1998. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2 (2), 121–167.

Byrne, W., 2006. Minimum Bayes risk estimation and decoding in large vocabulary continuous speech recognition. In: Minimum Bayes risk estimation and decoding in large vocabulary continuous speech recognition. Japan.

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

Chakrabartty, S., 2004. Design and Implementation of Ultra-Low Power Pattern and Sequence Decoders. Ph.D. thesis, The Johns Hopkins University.

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

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learning 20 (3), 273–297.

Doumpiotis, V., 2005. Discriminative Training for Speaker Adaptation and Minimum Bayes Risk Estimation in Large Vocabulary Speech Recognition. Ph.D. thesis, The Johns Hopkins University.

Doumpiotis, V., Tsakalidis, S., Byrne, W., 2003a. Discriminative training for segmental minimum Bayes risk decoding. In: Proc. ICASSP. Hong Kong.

Doumpiotis, V., Tsakalidis, S., Byrne, W., 2003b. Lattice segmentation and minimum Bayes risk discriminative training. In: Proc. Eurospeech. Geneva.

Fine, S., Navrátil, J., Gopinath, R., 2001. A hybrid GMM/SVM approach to speaker identification. In: ICASSP. Utah, USA.

Fiscus, J., 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In: IEEE Wkshp Spch Recogn & Und.

Ganapathiraju, A., Hamaker, J., Picone, J., 2003. Advances in hybrid SVM/HMM speech recognition. In: GSPx / International Signal Processing Conference. Dallas, Texas, USA.

Goel, V., Byrne, W., 2000. Minimum Bayes-risk automatic speech recognition. In: Computer Speech & Language. Vol. 14(2).

Goel, V., Byrne, W., 2003. Minimum Bayes-risk automatic speech recognition. In: Chou, W., Juang, B.-H. (Eds.), Pattern Recognition in Speech and Language Processing. CRC Press.

Goel, V., Kumar, S., Byrne, W., 2001. Confidence based lattice segmentation and minimum Bayes-risk decoding. In: Proc. Eurospeech. Aalborg, Denmark.

Goel, V., Kumar, S., Byrne, W., 2004. Segmental minimum Bayes-risk decoding for automatic speech recognition. IEEE Transactions on Speech and Audio Processing12 (3):287-333.

Golowich, S. E., Sun, D. X., 1998. A support vector/hidden Markov model approach to phoneme recognition. In: ASA Proceedings of the Statistical Computing Section. pp. 125–130.

Jaakkola, T., Haussler, D., 1998. Exploiting generative models in discriminative classifiers. In: M. S. Kearns, S. A. S., Cohn, D. A. (Eds.), Advances in Neural Information Processing System. MIT Press.

Jelinek, F., February 1996. Speech recognition as code-breaking. Tech. Rep. Tech Report No. 5, CLSP, JHU.

Kumar, S., 2004. Minimum Bayes-Risk Techniques in Automatic Speech Recognition and Statistical Machine Translation. Ph.D. thesis, The Johns Hopkins University.

Kumar, S., Byrne, W., 2002. Risk based lattice cutting for segmental minimum Bayes-risk decoding. In: ICSLP. Denver, Colorado, USA.

Lafferty, J., McCallum, A., Pereira, F., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML.

Legetter, C. J., Woodland, P. C., 1995. Maximum likelihood linear regression for speaker adaptation of continuous density hmms. In: Computer, Speech and Language. Vol. 9. pp. 171–186.

Mangu, L., Brill, E., Stolcke, A., 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. Computer Speech and Language 14 (4), 373–400.

Mangu, L., Padmanabhan, M., 2001. Error corrective mechanisms for speech recognition. In: ICASSP. Utah, USA.

Mohri, M., Pereira, F., Riley, M., 2001. AT&T Generalpurpose Finite-State Machine Software Tools. Available: http://www.research.att.com/sw/tools/fsm/.

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

Normandin, Y., 1995. Optimal splitting of HMM gaussian mixture components with mmie training. In: Proc. ICASSP. Vol. 15. pp. 449–452.

Normandin, Y., 2002. Maximum mutual information estimation of hidden Markov models. In: M. S. Kearns, S. A. S., Cohn, D. A. (Eds.), Automatic Speech and Speaker Recognition. Vol. 15. Cambridge: MIT Press.

Salomon, J., King, S., Osburne, M., 2002. Framewise phone classification using support vector machines. In: Proc. ICSLP.

Smith, N. D., 2003. Using Augmented Statistical Models and Score Spaces for Classification. Ph.D. thesis, Cambridge University Engineering Department.

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

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

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, New York, Inc., Ch. 5.

Venkataramani, V., 2005. Code breaking for automatic speech recognition. Ph.D. thesis, The Johns Hopkins University.

Venkataramani, V., Byrne, W., 2003. Support vector machines for segmental minimum Bayes risk decoding of continuous speech. In: Proc. ASRU.

Venkataramani, V., Byrne, W., 2005. Lattice segmentation and support vector machines for large vocabulary continuous speech recognition. In: Proc. ICASSP.

Wessel, F., Macherey, K., Schlueter, R., 1998. Using word probabilities as confidence measures. In: Proc. ICASSP. Seattle, WA, USA, pp. 225–228.

Weston, J., Watkins, C., May 1998. Multi-class support vector machines. Tech. Rep. CSD-TR-9800-04, Department of Computer Science, University of London.

Woodland, P. C., Povey, D., 2000. Large scale discriminative training for speech recognition. In: Proc. ITW ASR, ISCA.

Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P., July 2000. The HTK Book, Version 3.0.