USING SVMS AND DISCRIMINATIVE MODELS FOR SPEECH RECOGNITION

N.D. Smith and M.J.F. Gales

Cambridge University Engineering Department
Cambridge, CB2 1PZ, U.K.
{nds1002,mjfg}@eng.cam.ac.uk

ABSTRACT

In speech recognition, standard MAP decoders attribute speech data to the class with the highest posterior probability. This minimises the error rate under assumptions of model correctness. This assumption is invalid for speech recognition with HMMs. Hence, an interesting question is whether extra, useful information about the speech source can be extracted from the HMMs and used to lower error rates in practical systems. In this paper additional features are extracted from HMMs and incorporated into a multi-dimensional score-space. SVMs are then used to implement a decision rule. Preliminary experiments are performed on a small speaker-independent isolated letter task. Score-spaces based on discriminative models are used with previous results based on generative models. Both score-spaces outperform standard schemes.

1. INTRODUCTION

Most state-of-art speech recognition systems use Hidden Markov Models (HMMs) as the basic acoustic models. HMMs are commonly trained by Maximum Likelihood Estimation (MLE) and so may be considered as forms of generative model. Under assumptions of model correctness, these models maximally discriminate between classes. When used in a Maximum A-Posteriori (MAP) decoder, the lowest possible error rate for a problem is attained ('error rate' is used to describe the probability of error, elsewhere 'training error rate' and 'test error rate' are used to describe empirical results). In speech recognition with HMMs, the model correctness assumption is invalid and practical systems based on generative models and MAP decoders yield higher error rates. Error rates can be reduced using discriminative training methods, such as Maximum Mutual Information Estimation (MMIE) [1], since these methods do not require model correctness. During decoding, both generative and discriminative models are similarly used to return simple acoustic scores which are compared in decision rules. Rather than just extracting acoustic scores from the models, it is possible to extract extra data-dependent information. This information may be useful for reducing error rates in practical systems and counteracting poor modelling assumptions. One possible scheme for extracting extra information is to use *Fisher scores* [2]. Given a generative model of the data, Fisher scores originally consisted of the first-order derivatives of the likelihood with respect to each of the model parameters. A multi-dimensional Fisher score was yielded and a classifier, typically a Support Vector Machine (SVM) [3], was constructed in this space. Fisher scores were defined for binary classifiers using a single generative model for both classes. Fisher scores were later extended to more general score-spaces for generative models in [4]. This paper uses and applies this work to score-spaces defined on discriminative HMMs.

2. TRAINING AND DECODING

Consider the set of training utterances $\{\boldsymbol{O}_1 \cdots \boldsymbol{O}_n \cdots \boldsymbol{O}_N\}$ where $\boldsymbol{O}_n = (\boldsymbol{o}_1^n, \ldots \boldsymbol{o}_t^n, \ldots \boldsymbol{o}_T^n)$ and $\boldsymbol{o}_t \in \mathbb{R}^D$. Each utterance belongs to a distinct class and each class is modelled by a probability model called an *acoustic model*. For class ω_i, the acoustic model is $p(\boldsymbol{O}|\omega_i)\}$ ($\boldsymbol{\theta}_i$ is used as the parameter vector for this model). This acoustic model can be trained using various criteria. For example, it may be trained using MLE, in which case it is a form of generative model. Alternatively, it may be trained using MMIE, in which case it is discriminative. These training criteria modify model parameters to respectively maximise $\mathcal{F}_{MLE}(i)$ and $\mathcal{F}_{MMIE}(i)$ given as following, where $P(\omega_i)$ is the prior for class ω_i,

$$\mathcal{F}_{MLE}(i) = \sum_{n=1}^{N} \ln p(\boldsymbol{O}_n|\omega_i)P(\omega_i) \quad (1)$$

$$\mathcal{F}_{MMIE}(i) = \sum_{n=1}^{N} \ln \frac{p(\boldsymbol{O}_n|\omega_i)P(\omega_i)}{\sum_{q=1}^{Q} p(\boldsymbol{O}_n|\omega_q)P(\omega_q)} \quad (2)$$

For each model, MLE maximises the average log likelihood over all the training utterances belonging to the model. MMIE maximises the average log posterior probability over all the training utterances belonging to the model [1].

Although generative and discriminative models have different training criteria, most classifiers use these acoustic models in a similar manner. First, consider two generative models which are used to give a binary decision on class membership for utterance $\boldsymbol{O}$. The two generative models are $p(\boldsymbol{O}|\omega_1)$ and $p(\boldsymbol{O}|\omega_2)$ for classes ω_1 and ω_2 respectively. The two classes can be compared using a log likelihood-ratio $\rho(\boldsymbol{O})$,

$$\rho(\boldsymbol{O}) = \ln \frac{p(\boldsymbol{O}|\omega_1)}{p(\boldsymbol{O}|\omega_2)} \quad (3)$$

Rather than basing a decision on $\rho(\boldsymbol{O})$, a more informative decision can be made by weighting $\rho(\boldsymbol{O})$ by the log ratio of model priors $\ln P(\omega_1)/P(\omega_2)$. This results in a MAP decoder which decides class membership using the log posterior-ratio $\rho_{\mathrm{MAP}}(\boldsymbol{O})$,

$$\rho_{\mathrm{MAP}}(\boldsymbol{O}) = \rho(\boldsymbol{O}) + \ln \frac{P(\omega_1)}{P(\omega_2)} \quad (4)$$

The MAP decoder selects class ω_1 if $\rho_{MAP}(\boldsymbol{O}) > 0$, else it selects class ω_2. This decision minimises the error rate for a given set of models and data, but only attains the lower bound under assumptions of model correctness. MAP decoders differ from likelihood-ratio decoders only in the use of an offset or bias in their decision rule. Under equal class priors, the two decoders are equivalent. In this paper, standard MAP decoders are used which base their decisions on a zero threshold in $\rho(\boldsymbol{O})$ as class priors are assumed equal since each class is equally represented in the training set (some decision schemes are presented later which add a data-dependent offset or bias to the zero threshold on $\rho(\boldsymbol{O})$; these schemes either optimise a threshold to accommodate model incorrectness and lack of data, or to reflect the training algorithm and its regularisation). The MAP decoder is readily extended to multiple classes by selecting the class with the highest posterior probability. If only binary decoders are available, such as in the standard SVM framework, then a method to map binary decisions to a multi-class decision is required. If each binary decoder returns a 'hard' decision on class membership, then a full set of binary comparisons may be performed and the class with most wins selected. Other methods may be used for decoders which return 'soft' probabilistic decisions. Although generative models are usually used to describe MAP decoders, discriminative models can be substituted. However in this case the terms 'likelihood' and 'class prior' should be replaced by the more generic terms 'acoustic score' and 'language model score', since the discriminatively-trained models are no longer generative or representative of data density. For convenience, this paper still retains the term 'MAP decoder' for decoders based on discriminative models, although the decision is strictly no longer based on posterior probabilities.

A binary MAP decoder bases its decision entirely upon the ratio $\rho(\boldsymbol{O})$ and the language model scores (see Equation 4). It is reasonable to assume that a more informative decision can be made by including extra information describing the relationship between the acoustic models and the data they represent. One possible source of useful information is the sensitivity of $\rho(\boldsymbol{O})$ to the parameters of each acoustic model [2]. This sensitivity can be captured by the derivatives of $\rho(\boldsymbol{O})$ with respect to the model parameters. For example, consider the vector of parameters for our two competing models $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T]^T$, where the dimensionalities of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are d_1 and d_2 respectively. The first-order derivatives of $\rho(\boldsymbol{O})$ with respect to these parameters are given by $\nabla_{\boldsymbol{\theta}}\rho(\boldsymbol{O})$ (which is here defined as a column vector). Rather than build a decision rule in the 1-dimensional space given by $\rho(\boldsymbol{O})$, a decision rule can be built in the $(d_1 + d_2 + 1)$-dimensional space given by appending $\nabla_{\boldsymbol{\theta}}\rho(\boldsymbol{O})$ to $\rho(\boldsymbol{O})$. This new space is an example of what is called a *score-space*. A brief presentation of score-spaces follows.

3. SCORE-SPACES AND SVMS

Following the arguments and assumptions in [2], the observation sequence $\boldsymbol{O}$ can be mapped into a vector of fixed dimension,

$$\varphi_{\hat{F}}^{f}(\boldsymbol{O}) = \varphi_{\hat{F}} f\big(\{p(\boldsymbol{O}|\omega_i)\}\big) \tag{5}$$

The models which form the set $\mathcal{P} = \{p(\boldsymbol{O}|\omega_i)\}$ are called *score-models*. A particular function of the score-models, $f(\cdot)$, is called a *score-argument*. The score-argument is mapped via a *score-operator*, $\hat{F}$, to a fixed-dimensional space called the *score-space*, $\varphi_{\hat{F}}^{f}(\boldsymbol{O})$. The different vectors which occupy the score-space are simply called *scores*. The score-space is therefore wholly defined by the score-models, score-argument and score-operator. In this paper, the score-models considered are HMMs with Gaussian Mixture Model (GMM) output distributions. The score-argument is the ratio $\rho(\boldsymbol{O})$, and the score-operator is the zeroth-order and first-order derivative operators (a fuller discussion of score-arguments and score-operators can be found in [4]). The resulting $(d_1 + d_2 + 1)$-dimensional score-space $\varphi(\boldsymbol{O})$ is given by,

$$\varphi(\boldsymbol{O}) = \begin{bmatrix} \rho(\boldsymbol{O}) \\ \nabla_{\boldsymbol{\theta}}\rho(\boldsymbol{O}) \end{bmatrix} = \begin{bmatrix} \rho(\boldsymbol{O}) \\ \nabla_{\boldsymbol{\theta}_1} \ln p(\boldsymbol{O}|\omega_1) \\ -\nabla_{\boldsymbol{\theta}_2} \ln p(\boldsymbol{O}|\omega_2) \end{bmatrix} \tag{6}$$

In this paper, only first-order derivatives with respect to the individual Gaussian means are considered, as these were shown to be important model parameters in [4]. For Gaussian component k belonging to state j of class ω_i, with mean $\boldsymbol{\mu}_{ijk}$ and covariance Σ_{ijk}, the first-order derivative of $\ln p(\boldsymbol{O}|\omega_i)$ with respect to $\boldsymbol{\mu}_{ijk}$ is as follows,

$$\nabla_{\boldsymbol{\mu}_{ijk}} \ln p(\boldsymbol{O}|\omega_i) = \sum_{t=1}^{T} \hat{\gamma}_{ijk}(t)\Sigma_{ijk}^{-1}(\boldsymbol{o}_t - \boldsymbol{\mu}_{ijk}) \tag{7}$$

$\hat{\gamma}_{ijk}(t)$ is a normalised form of the posterior probability for component k of state j of class ω_i at time t (this normalised posterior probability implements *sequence length normalisation* and may be used to ensure that two different utterances, identical apart from their speaking rate, project into the same region of score-space [4]). The vector $\nabla_{\boldsymbol{\theta}_1} \ln p(\boldsymbol{O}|\omega_1)$ is therefore assembled as a super-vector formed by splicing together $\nabla_{\boldsymbol{\mu}_{ijk}} \ln p(\boldsymbol{O}|\omega_1)$ for each Gaussian component in model ω_1. A similar super-vector can be formed for $\nabla_{\boldsymbol{\theta}_2} \ln p(\boldsymbol{O}|\omega_2)$. These two super-vectors are then appended to the ratio $\rho(\boldsymbol{O})$ to yield the $(d_1 + d_2 + 1)$-dimensional score-space defined in Equation 6. If a decision boundary is drawn in this score-space to separate two classes of data, then the decision boundary can be used as a decoder. If the score-space is simply defined using the zeroth-order derivative score-operator, then a single-dimensional score-space is obtained containing only $\rho(\boldsymbol{O})$. If the decision boundary in this 1-dimensional score-space is constrained to lie at zero, then a MAP decoder is yielded.

The decision boundary in score-space can be obtained by different learning algorithms. In this paper, SVMs [3] are used since,

1. SVMs claim to yield decision boundaries which generalise well in the presence of few training samples relative to the size of the vector-space. This is particularly appropriate for score-spaces which may be very large. It does this by placing decision boundaries 'midway' between competing classes. This is known as 'maximising the margin' [3].
2. SVMs can be used to give non-linear decision boundaries in the score-space through the use of non-linear kernels. These are potentially more useful than linear boundaries.
3. An SVM can incorporate the score-mapping directly into its kernel, and so can be viewed as a classifier which operates directly in the input-space of observation sequences [2].

Unfortunately SVMs are inherently binary classifiers and return a hard decision on class membership (though there are some extensions for 'soft' probabilistic outputs [5]). To extend SVMs to a multi-class decision rule, schemes such as majority voting must be used. Such techniques risk sub-optimal performance if each classifier is trained separately.

The approach adopted to train SVMs in the score-space affects the computational cost. For example, if a linear decision boundary is required in the score-space, then training and testing requires the evaluation of inner scalar products in the score-space. Take two observation sequences $\boldsymbol{O}_i$ and $\boldsymbol{O}_j$ which have corresponding scores $\varphi(\boldsymbol{O}_i)$ and $\varphi(\boldsymbol{O}_j)$. As the different dimensions of score-space are non-orthogonal, the inner scalar product between the two scores is strictly defined in terms of $\Sigma_{\rm sc}$, the covariance matrix in the score-space,

$$< \varphi(\boldsymbol{O}_i), \varphi(\boldsymbol{O}_j) > \quad = \quad \varphi(\boldsymbol{O}_i)^T \Sigma_{\rm sc}^{-1} \varphi(\boldsymbol{O}_j) \tag{8}$$

A computationally feasible method of training is as follows: first transform each utterance $\boldsymbol{O}$ into its corresponding score $\varphi(\boldsymbol{O})$, normalise each score by $\Sigma_{\rm sc}^{-1/2}$ (this is called *score-space normalisation*), and then train and test an SVM using a standard linear kernel in the normalised score-space. This is equivalent to training and testing an SVM in the input-space of observation sequences using the kernel $k(\boldsymbol{O}_i, \boldsymbol{O}_j) = \varphi(\boldsymbol{O}_i)^T \Sigma_{\rm sc}^{-1} \varphi(\boldsymbol{O}_j)$.

It is valuable to compare the decision rules for a binary MAP decoder using MMI models, and an SVM classifier using the corresponding score-space. The MMI models are optimised to maximise the average 'posterior margin' over the utterances, where the 'posterior margin' for an utterance is defined as the difference between the log posterior probabilities of correct and incorrect class membership (see Equation 2). For a binary MAP decoder which assumes equal language model scores for an unknown utterance $\boldsymbol{O}$, the decoder uses the models to evaluate $\rho(\boldsymbol{O})$ and sets a zero threshold for a hard decision on class membership. If an SVM is trained using only $\rho(\boldsymbol{O})$, then a zero or non-zero threshold is obtained to jointly minimise errors and 'maximise the margin'. When first-order derivatives are added to form a multi-dimensional score-space, the SVM selects the decision hyperplane which simultaneously minimises errors and 'maximises the margin' across all the dimensions of score-space. The projection of the hyperplane onto the $\rho(\boldsymbol{O})$-axis gives the new threshold in $\rho(\boldsymbol{O})$. The threshold may have shifted. Since each dimension of score-space contains useful information relating the MMI models to the speech data, the new threshold may more accurately reflect differences between classes and counteract some of the modelling deficiencies. Also, when models are used to define score-spaces, it may be possible to use a lower complexity than is normal for corresponding MAP decoders.

The experiments in this paper investigate whether a score-space based on discriminative score-models can be used to extract extra useful information from those models and the speech data. They also illustrate how SVMs can be used to extract this extra information.

4. EXPERIMENTAL RESULTS

The use of SVMs with generative and discriminative models was applied to the classification of isolated letter utterances in the ISOLET database [6]. The data was coded at a 10msec frame rate with a 25.6msec window-size, and parameterised into 39-dimensional feature vectors comprising 12 MFCCs and a log energy term with corresponding delta and acceleration parameters. Training used 240 utterances per letter from `isolet{1,2,3,4}` and testing used 60 utterances per letter from `isolet5`. Experiments were conducted on the harder E-set {`B C D E G P T V Z`}.

The baseline HMMs modelled each letter by a 10-emitting state left-to-right continuous density HMM with no skips. Silence was modelled by a single emitting-state HMM with no skips. Each state output distribution was a GMM with equal numbers of components and diagonal covariances. Three sets of baseline models were used; a set of well-trained ML models, a set of MMI models after 5 MMI training iterations, and a set of MMI models after 20 MMI training iterations [1]. The ML models were used to initialise the MMI models, and to obtain the denominator and numerator lattices for MMI-training. The denominator lattices contained all 9 competing paths for the E-set. MMIE statistics were calculated using hard frame-model alignments because this method was computationally cheaper than using softer frame-model alignments, and was found to give better performance in some preliminary experiments (this was probably due to high confidence in the letter-silence boundaries in our simple task). The acoustic model scaling, a, was difficult to optimise because of noise, but reasonable values were selected for different numbers of components in HMM-MMI20 and applied to both HMM-MMI5 and HMM-MMI20. The performance of the MMI models was very sensitive to a (e.g. for HMM-MMI20 with 2 components per state, a test error rate of 7.0% was obtained for $a = 1$ which is the acoustic model scaling for standard MMI, 4.8% for our baseline at $a = 30$, and 4.1% for $a = 40$). The ML and MMI models were tested using a Viterbi recogniser constrained to a silence-letter-silence network. Multi-class performance was measured as the utterance error rate, obtained by a standard multi-class MAP decoder and also a majority voting scheme among the full set of 1-v-1 binary classifiers (for tied letters, the relevant 1-v-1 classifiers were inspected and then, if necessary, randomly selected; in hindsight, a better scheme for SVM classifiers would replace random selection by a comparison of likelihood-ratios if available). The three sets of baseline HMMs were then used to define score-spaces.

For SVMs, a modified version of SVM^{light} Version 3.02 [7] was used. Following the experiments in [4], scaling approximated full score-space normalisation (i.e. approximate $\Sigma_{\rm sc}^{-1/2}$ by its empirically calculated diagonal form), and sequence length normalisation was used. In these experiments, only linear kernels were used in the normalised score-space (for ML score-models, these were shown to give better performance than GRBF and polynomial kernels constructed in the normalised score-space). The parameter C was fixed at 100, and biased hyperplanes were permitted.

For clarity, HMM-ML, HMM-MMI5 and HMM-MMI20 are used to denote the HMMs trained using MLE, and 5 and 20 iterations of MMIE. Training and testing used the standard multi-class training and MAP decoding framework. SVM-ML, SVM-MMI5 and SVM-MMI20 are used to denote the corresponding SVM systems. Table 1 compares the test error rates as the complexity of the score-models was varied (all results for HMM-ML and SVM-ML in this paper were previously quoted in [4] and are gratefully reproduced for convenience). The training error rates for the HMM systems are bracketed. The results were comparable to reported results on E-set recognition [8]. For this simple task, the majority voting scheme gave the same test error rates as the multi-class MAP decoder for spot-checks on HMM-ML and HMM-MMI20. This indicated that majority voting was an acceptable multi-class scheme for these experiments.

For the standard HMM systems, MMI models gave lower test error rates than ML models, confirming the superior performance of discriminative training. For MMI models, the training errors were almost all zero and illustrated that MMIE was excellent at learning patterns in the training data. The zero error rates also indicated the task was very simple and care should be taken in drawing conclusions. Although the training errors for HMM-MMI5 and HMM-MMI20 were both zero, the two model sets were not identical and HMM-MMI20 attained a higher MMIE criterion value. In general, the results show that the more fully-trained MMI models generalised better to test data than the less-trained MMI models.

		Components per state per class			
		1	2	4	6
H	ML	11.3 (7.2)	8.7 (4.5)	6.7 (2.2)	7.2 (1.7)
M	MMI5	5.6 (0.7)	5.4 (0.0)	5.0 (0.0)	5.4 (0.0)
M	MMI20	5.0 (0.0)	4.8 (0.0)	4.4 (0.0)	4.3 (0.0)
S	ML	6.9	5.0	5.7	6.1
V	MMI5	6.1	4.4	5.2	4.3
M	MMI20	5.9	4.3	5.6	5.4

Table 1. Test (training) error rates for different systems (E-set)

The zero error rates for the MMI models imply perfect representation of the training data. However as the training data is finite, these models do not necessarily generalise well. Although the inclusion of extra information from score-spaces may introduce training errors, the resulting decoder may generalise better. With this aim, the three sets of score-models were used to produce score-spaces and incorporated into SVM systems. SVM-ML outperformed HMM-ML, though the same gains were not obtained for SVM-MMI5 and SVM-MMI20. The lowest test error rate of 4.3% was obtained jointly by HMM-MMI20, SVM-MMI5 and SVM-MMI20. However, the ML results show that the SVM-ML score-space successfully incorporated extra discriminatory information from the generative models which was not available in the log posterior ratio and zero class threshold.

In many applications, feature-selection can be used to discard dimensions of feature-space which are noisy and inhibit classification performance. Experiments were conducted to select the most discriminatory dimensions from the SVM-ML, SVM-MMI5 and SVM-MMI20 score-spaces defined on score-models with 2 components per state (column 2 of Table 1). SVM classifiers were then constructed in these score-subspaces (the computational load in training and testing should consequentially decrease). Dimensions of score-space were selected according to highest Fisher-ratios. Generally, the test error rates for SVM-MMI5 were lower than those for SVM-MMI20 across the range of dimension sizes. The lowest test error rates for SVM-ML, SVM-MMI5 and SVM-MMI20 were respectively 3.3%, 3.2% and 3.9% for dimension sizes of 250, 350 and 500 (the full score-space has 1561 dimensions). Although the results were noisy, the test error rates were generally lower than those obtained using the same models in standard HMM schemes. The use of score-spaces has successfully incorporated extra discriminatory information from the discriminative, as well as the generative, models. However, larger tasks are required to investigate which score-models are best. If the decision thresholds in the $\rho(\boldsymbol{O})$-dimension of score-space are compared with the zero thresholds for a standard MAP decoder, then the thresholds have shifted to yield lower overall test error rates.

The SVMs used a simple majority voting scheme among the full set of binary classifiers. Each SVM was trained in a 1-v-1 manner. However, HMM-MMI5 and HMM-MMI20 were trained in a multi-class manner. It is interesting to produce an MMI system which is analogous to an SVM system in its use of data. For this reason, a pairwise MMI (pMMI) system was trained in which a full set of 1-v-1 MMI classifiers were constructed using HMMs with 2 components per state (comparable to column 2 of Table 1). Each MMI classifier was therefore trained using only two classes of data, and so was optimised for 1-v-1 rather than 1-v-rest discrimination. A single reasonable acoustic model scaling was used for all pairwise models. The same majority voting scheme was used as for the SVMs. A test error rate of 4.8% was obtained, and 5.2% with 0% training error rate. This compared to 4.8% for HMM-MMI20 trained in a multi-class manner. These results are inconclusive, but SVMs may lose performance if each binary classifier is trained separately. It would be useful to investigate multi-class SVM training schemes [9].

5. CONCLUSIONS

A MAP decoder using generative models attains the lowest possible error rate for a problem under conditions of model correctness. These conditions are invalid for speech recognition with HMMs, and discriminative training is sometimes used to reduce error rates in practical systems. This paper describes an alternative scheme using more complex score-spaces which allow extra information from generative or discriminative models to be incorporated into decision rules. These more complex score-spaces were evaluated on a speaker-independent isolated letter task which used SVMs to implement the decision rule. Preliminary experiments showed that useful score-spaces could be defined on discriminative as well as generative models. Larger tasks are required for better evaluation.

6. ACKNOWLEDGEMENTS

N. Smith thanks EPSRC and his CASE sponsor, the Speech Group at IBM U.K. Laboratories, and Thorsten Joachims, University of Dortmund, for his software SVM^{light}. He also thanks Dan Povey and KK Chin for software, and Phil Woodland for helpful discussions on MMI-training. Part of this investigation involved use of equipment kindly donated by IBM under an SUR award.

7. REFERENCES

[1] P.C. Woodland and D. Povey, “Large Scale Discriminative Training for Speech Recognition,” in *Proceedings*. ISCA ITRW ASR2000, September 2000, pp. 7–16.

[2] T. Jaakkola and D. Haussler, “Exploiting Generative Models in Discriminative Classifiers,” in *Advances in Neural Information Processing Systems 11*, M.S. Kearns, S.A. Solla, and D.A. Cohn, Eds. MIT Press, 1999.

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

[4] N. Smith and M. Gales, “Speech Recognition using SVMs,” in *Advances in Neural Information Processing Systems*, T.G. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. MIT Press, 2002.

[5] J. Platt, “Probabilistic outputs for Support Vector Machines and comparisons to Regularized Likelihood Methods,” in *Advances in Large Margin Classifiers*. MIT Press, 1999.

[6] M. Fanty and R. Cole, “Spoken Letter Recognition,” in *Neural Information Processing Systems 3*, R.P. Lippmann, J.E. Moody, and D.S. Touretzky, Eds., pp. 220–226. Morgan Kaufmann Publishers, 1991.

[7] T. Joachims, “Making Large-Scale SVM Learning Practical,” in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. MIT-Press, 1999.

[8] P.C. Loizou and A.S. Spanias, “High-Performance Alphabet Recognition,” *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 6, pp. 430–445, Nov. 1996.

[9] J. Weston and C. Watkins, “Multi-class Support Vector Machines,” Tech. Rep. CSD-TR-98-04, Royal Holloway, University of London, Department of Computer Science, May 1998.