A SVM/HMM SYSTEM FOR SPEAKER RECOGNITION

W. M. Campbell

Motorola Human Interface Lab
Tempe, AZ 85284
E-mail: wmcampbell@ieee.org

ABSTRACT

A framework for combining support vector machines with hidden Markov models (HMM's) is given. A HMM is used with a Viterbi alignment to generate a set of subsequences of feature vectors. Each subsequence is then scored using a support vector machine sequence kernel. Experiments are performed for both text-independent and text-prompted speaker recognition tasks. Results show that the method can dramatically reduce error rates over a support vector machine (SVM) only system.

1. INTRODUCTION

Hidden Markov models (HMM's) and Gaussian mixture models (GMM's) have been popular methods for implementing speaker verification, see [1] and [2] respectively. In both cases, a log likelihood score between a speaker model and a background is used for verification. For hidden Markov models, a speaker model is constructed by concatenating speaker-specific models of subwords; the background is constructed using a speaker independent ASR system. For Gaussian mixture models, a typical strategy is to use MAP adaptation on a background model to create a speaker model [2]; the background model is created by training a GMM on a large population of speakers.

Alternatives to HMM's and GMM's have emerged in the research literature because of several needs. First, methods which minimize speaker model size and memory footprint are required in many applications (e.g., cell phones). Discriminatively trained classifiers produce systems which maximize the utility of trainable parameters. Second, computational scalability is desirable for large server applications or large speaker identification problems; low computation minimizes hardware and/or decreases transaction time. Discriminatively trained classifiers do not require background scoring and with certain structures can be made computationally scalable [3, 4].

An exciting area of recent focus has been the application of support vector machines (SVM's) in many different fields. In several applications, SVM's have excited much fervor because of their superior performance. SVM's have been applied to speaker recognition in several instances [5, 6, 7]. Because of discriminative training methods, SVM's satisfy many of the alternative needs stated.

We propose a new method for combining HMM's with SVM's. The idea is based upon the probabilistic framework for scoring derived in [3]. The same computational advantages are preserved by the new approach; i.e., both computational scalability (with the number of speakers) and low computational complexity are maintained. In this paper, we focus on the problem of speaker verification, but the methods are applicable to speaker identification also.

The outline of the paper is as follows. In Section 2, we review support vector machines and issues for speech processing. In Section 3, we review a sequence kernel for speaker recognition. Section 4 proposes a new framework for scoring with a generalized linear discriminant/HMM architecture. Section 5 describes our new training method. In Section 6, we show experiments comparing a SVM only system with the new SVM/HMM system. Based upon whether a text-prompted or text-independent scenario is selected, we obtain quite different results.

2. SUPPORT VECTOR MACHINES FOR SPEECH PROCESSING

A support vector machine is typically constructed as a two-class classifier. The classifier is constructed from sums of a kernel function $K(\cdot,\cdot)$,

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i t_i K(\mathbf{x}, \mathbf{x}_i) + b; \tag{1}$$

where the t_i are targets, and $\sum_{i=1}^{N} \alpha_i t_i = 0$. The vectors $\mathbf{x}_i$ are support vectors and obtained from the training set by an optimization process [8]. The target values are either 1 or -1 depending upon whether the corresponding support vector is in class 1 or class 2. For classification, one looks at the sign of $f(\mathbf{x})$ and makes a class decision based upon whether the value is positive or negative.

The kernel $K(\cdot,\cdot)$ is constrained to have certain properties (the Mercer condition), so that $K(\cdot,\cdot)$ can be expressed

as

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{b}(\mathbf{x})^t \mathbf{b}(\mathbf{y}) \quad (2)$$

where $\mathbf{b}(\mathbf{x})$ is a mapping from the input space (where $\mathbf{x}$ lives) to a possibly infinite dimensional space. If the explicit form of $\mathbf{b}(\mathbf{x})$ is available (which is not usually the case), then $f(\mathbf{x})$ can be written as

$$f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^t \left[\sum_{i=1}^{N} \alpha_i t_i \mathbf{b}(\mathbf{x}_i) \right] + b = \begin{bmatrix} \mathbf{b}(\mathbf{x}) & 1 \end{bmatrix}^t \mathbf{w} \quad (3)$$

where $\mathbf{w}$ is the sum in brackets in (3) with the scalar b appended. The advantage of the form (3) over (1) is that it involves only storage and computation with the vector $\mathbf{w}$ as opposed to the support vectors. If the dimension of $\mathbf{b}(\mathbf{x})$ is "small" and there are a significant number of support vectors, using (3) can significantly reduce storage space and computation.

Generalizing support vector machines to a classification task involving speech presents several difficulties. First, speech feature vectors (e.g., cepstral coefficients) tend to produce classification problems with complex overlapping classification regions. This along with other factors produces many support vectors. This problem can be mitigated by using either the alternate form (3) (our approach) or methods which reduce the number of support vectors, e.g. [9]. Second, speech classification problems involve sequences of feature vectors rather than the typical usage of a SVM as a single input classifier. Since the ("soft") SVM output $f(\mathbf{x})$ is not a probability, this presents a challenge. To solve this problem, one can use an ad hoc method where the SVM output $f(\mathbf{x})$ is used to approximate an emission probability for a HMM [6]. Alternatively a kernel which compares sequences of speech feature vectors directly can be constructed [5, 7].

3. A SEQUENCE KERNEL

Following the approach in [7], we review a kernel based upon comparing *sequences* of speech feature vectors. Suppose we have two utterances from two speakers (possibly the same speaker), and they are represented by sequences of speech vectors $\{\mathbf{x}_i\}_{i=1}^{N_x}$ and $\{\mathbf{y}_i\}_{i=1}^{N_y}$ (e.g., each vector is the cepstral coefficients for a frame of speech). We construct a kernel by first training on one utterance using a generalized linear discriminant. Then, the resulting model is scored on the remaining utterance using the standard independence of observations assumption.

In detail, suppose we are using a generalized linear discriminant of the form $\mathbf{d}^t \mathbf{b}(\mathbf{x})$ where

$$\mathbf{b}(\mathbf{x}) = \begin{bmatrix} b_1(\mathbf{x}) & b_2(\mathbf{x}) & \dots & b_k(\mathbf{x}) \end{bmatrix}^t, \quad (4)$$

b_i is a mapping from $\mathbb{R}^m$ to $\mathbb{R}$, and $\mathbf{d}$ is the vector of discriminant model parameters. One can then show that mean-squared error training for the speaker verification problem gives

$$\mathbf{d} = \bar{\mathbf{R}}^{-1} \bar{\mathbf{b}}_x. \quad (5)$$

The vector $\bar{\mathbf{b}}_x$ is defined as

$$\bar{\mathbf{b}}_x = \frac{1}{N_x} \sum_{i=1}^{N_x} \mathbf{b}(\mathbf{x}_i), \quad (6)$$

and $\bar{\mathbf{R}}$ is a correlation matrix derived from a large background population. Scoring on the remaining utterance produces the Generalized Linear Discriminant Sequence (GLDS) kernel,

$$K_{\mathrm{GLDS}}(\{\mathbf{x}_i\}, \{\mathbf{y}_i\}) = \bar{\mathbf{b}}_x^t \bar{\mathbf{R}}^{-1} \bar{\mathbf{b}}_y. \quad (7)$$

where $\bar{\mathbf{b}}_y$ is defined in an analogous manner to (6).

Two items should be noted about the kernel in (7). First, since we are using an explicit expansion $\mathbf{b}(\cdot)$, we can simplify to a single model as in (3). Second, optimization using the kernel produces support *sequences*. As a consequence, we find sequences from the speaker and from the background set of speakers (in a verification problem) that construct the decision surface.

4. SCORING

The GLDS kernel can be combined in a straightforward manner with a HMM. For the case of text-prompted speaker recognition, we assume we have a left to right HMM with the states corresponding to subwords in the utterance. For the case of text-independent speaker recognition, we transform a single GMM to a multi-state HMM where each individual Gaussian is a state.

Now suppose we have a sequence of feature vectors, $\{\mathbf{x}_i\}_{i=1}^{N_x}$. If we perform a Viterbi alignment using an HMM model, we obtain a series of states $\{s_i\}$. Suppose $1 \leq s_i \leq N_s$ and $S_i = \{j | s_j = i\}$. Now define

$$\bar{\mathbf{b}}_{x,i} = \frac{1}{|S_i|} \sum_{j \in S_i} \mathbf{b}(\mathbf{x}_j). \quad (8)$$

If the set S_i is empty, we let $\bar{\mathbf{b}}_{x,i} = \mathbf{0}$. We can then construct a HMM/SVM kernel (HGLDS) as follows ($\lambda_j > 0$):

$$K_{\mathrm{HGLDS}}(\{\mathbf{x}_i\}, \{\mathbf{y}_i\}) = \sum_{j=1}^{N_s} \lambda_j \bar{\mathbf{b}}_{x,j}^t \bar{\mathbf{R}}_j^{-1} \bar{\mathbf{b}}_{y,j}. \quad (9)$$

Here, $\bar{\mathbf{R}}_j$ is calculated by finding correlations in the subsequences of feature vectors that decode into state j for utterances from the background set. The function (9) is also a kernel since it is the sum of GLDS kernels to applied to subsequences labeled by the states [8]. We choose $\lambda_j = |S_j|/N_x$.

5. TRAINING

Training with the new HGLDS kernel entails some difficulty. For any two utterances from two speakers, the same sequence of subwords may not be spoken (and not necessarily in the same order). One could, of course, just assign a zero average expansion, $\bar{\mathbf{b}}_{x,i} = \mathbf{0}$, to the particular subsequence and then evaluate the kernel. Alternatively, one might expand a speaker's set of utterances (for the purposes of training) to include all possible concatenations of different subsequences. This is the approach we use. This process allows one to take an instance of the speaker saying "4" and "7" in different utterances and combine them into a new utterance with "4" and "7" as subsequences (if states correspond to digits). Note that for impostors in the background, we pool all impostors together so a valid sequence may contain subsequences from distinct speakers.

A second source of difficulty with the new kernel arises during training. Expanding the set of utterances by allowing arbitrary concatenations creates a huge data set. Thus, to simplify optimization, we assume that the problem is separable. That is, the support sequences can be found by finding support *subsequences* for each state individually.

The algorithm for training after these two assumption is straightforward. For each state, we construct a kernel based only upon the feature vectors that decode into that particular state for both a particular speaker and all impostors in the background. SVM training using the original GLDS kernel is then applied producing a model of the speaker in that state.

When scoring with the HGLDS kernel under this framework, one ignores transitions and orderings of states (another way to view this is that states are tied). That is, if the test utterance is "4-7-5", then apply a speaker model for "4" to the subsequence of the utterance corresponding to "4", etc., even if this particular ordering of subwords was never seen in training (as is typical in HMM speaker modelling [1]).

6. EXPERIMENTS

We performed two distinct tasks with the HGLDS kernel. One task applied HGLDS in text-prompted mode on the YOHO database using decades and digits as subwords. A second task involved using HGLDS with an 8 mixture GMM applied to the NIST 1998 speaker recognition evaluation.

For speech processing, a frame size of 30 ms was used with 20 ms overlap. Mean removal, preemphasis, and a Hamming window were applied. LP cepstral coefficients and the corresponding deltas were found. Energy-based endpointing eliminated non-speech frames.

6.1. Text-Prompted Experiments

YOHO uses combination lock phrases such as "26-81-57." Subwords (and states) corresponding to decades (20, 30, 40, etc.) and digits (1-9) were used. Both the GLDS kernel and the HGLDS kernel with full covariance $\bar{\mathbf{R}}$ were applied to the database. Input features vectors were 12 cepstral coefficients and 12 delta-cepstral coefficients. A vector of monomials up to degree 2 was used as the expansion $\mathbf{b}(\mathbf{x})$. The database was split into two distinct parts for enrollment and verification to avoid "seen" impostors (see [10] for details). Verification was done using 1-phrase tests. This resulted in $138 * 40 = 5520$ valid claims. Impostor attempts numbered $380,880$. We used a tradeoff between margin and error of $c = 1$ with SVMTorch [11].

Results for the text-prompted case are shown in the DET plot in Figure 1. From this figure, we see that the HGLDS kernel is clearly superior to the GLDS kernel. The equal error rate (EER) drops in this case from 1.64% for the SVM to 0.54% for the SVM/HMM. This result compares favorably with methods in the literature for HMM's, where a typical 1-phrase EER is, for example, 0.62% [1].

6.2. Text-Independent Experiments

We applied the SVM/HMM classifier system to the 1998 NIST one-speaker detection task. In this case, since the task is text-independent speaker recognition, a GMM split into Gaussian states was used for decoding. An impostor background was constructed from the 1997 NIST speaker recognition database. Four separate HGLDS kernels and

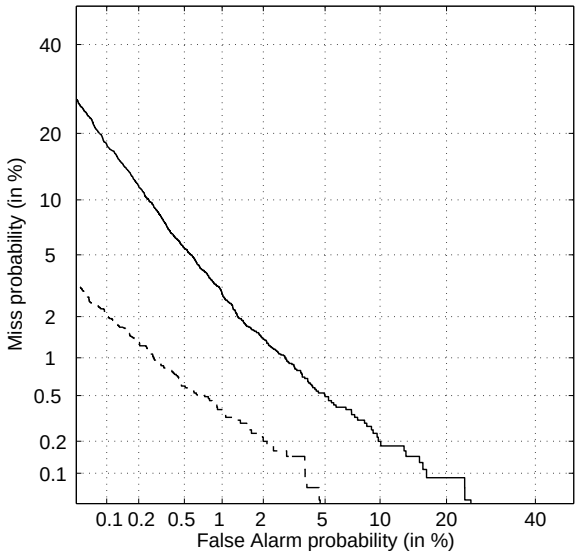

Fig. 1. DET plot of the SVM approach (solid line) versus the SVM/HMM approach (dashed line) on YOHO.

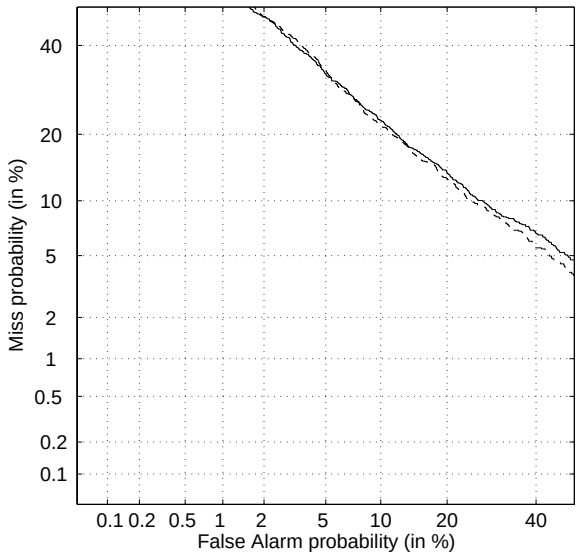

Fig. 2. DET plot of the SVM approach (dotted line) versus the SVM/GMM approach (dashed line) on NIST '98 SRE.

backgrounds for training were constructed corresponding to the labeling of each utterance as male or female and electret or carbon button. In addition, four separate GMM's were constructed with 8 mixture components. We trained the (female,electret) and (male,electret) using the (female,carbon button) and (male,carbon button) models, respectively, as starting points; this insured that Viterbi alignment was not significantly affected by channel issues. For enrollment, we chose the 2-session enrollment scenario. For verification, we concentrated on the 30 second test.

A vector of monomial terms up to degree three was used as the expansion $\mathbf{b}(\mathbf{x})$. For features, 12 cepstral coefficients were used. The matrix $\bar{\mathbf{R}}_j$ in (9) was approximated using only diagonal elements; this significantly reduced training time. We used a tradeoff between margin and error of $c = 1$ with SVMTorch. Both an SVM only system and a HMM/SVM were applied to the data.

The EER for different poolings is: same-number same-type (SNST) 4.0%, different-number same-type (DNST) 10.7%, and different-number different-type (DNDT) 15.6% (DNDT) systems. Figure 2 shows the DET for DNDT pooling. Both systems compare favorably to results in the literature [12] where typical SNST EER's are 5% and DNDT EER's are $15 - 20\%$. Note that the EER's are for a simple SVM/HMM system in this case, and we have not tested higher degree polynomials. Also, note that the SVM/HMM system performs only slightly better than the SVM-only system. This circumstance may have to do with the nature of the data set. Since the NIST trials have considerable variation in content and the GMM is initialized randomly in training, subsequences may not be independent enough to create a separable training problem (see Section 5). One possible way to avoid this difficulty is to initialize the GMM model based upon, e.g., phonemes.

7. CONCLUSIONS

A new method was presented for combining HMM's and SVM's for speaker verification. Results showed that dramatic improvements over SVM only systems could be achieved. Future work includes exploring alternate expansions $\mathbf{b}(\mathbf{x})$ and improving the performance of text-independent speaker recognition.

8. REFERENCES

[1] C. Che and Q. Lin, "Speaker recognition using HMM with experiments on the YOHO database," in *Proceedings of Eurospeech*, 1995, pp. 625–628.

[2] D.A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, pp. 19–41, Jan. 2000.

[3] William M. Campbell and C. C. Broun, "A computationally scalable speaker recognition system," in *Proceedings of EUSIPCO*, 2000, pp. 457–460.

[4] Ran D. Zilca, "Text-independent speaker verification using utterance level scoring and covariance modeling," *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 6, pp. 363–370, Sept. 2002.

[5] Shai Fine, Jiří Navrátil, and Ramesh A. Gopinath, "A hybrid GMM/SVM approach to speaker recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2001, pp. I–417–I–420.

[6] Vincent Wan and William M. Campbell, "Support vector machines for verification and identification," in *Neural Networks for Signal Processing X, Proceedings of the 2000 IEEE Signal Processing Workshop*, 2000, pp. 775–784.

[7] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing*, 2002, pp. 161–164.

[8] Nello Cristianini and John Shawe-Taylor, *Support Vector Machines*, Cambridge University Press, Cambridge, 2000.

[9] C. J. C. Burges, "Simplified support vector decision rules," in *13 th International Conference on Machine Learning*, 1996, pp. 71–77.

[10] William M. Campbell and Khaled T. Assaleh, "Polynomial classifier techniques for speaker verification," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 321–324.

[11] Ronan Collobert and Samy Bengio, "Support vector machines for large-scale regression problems," Tech. Rep. IDIAP-RR 00-17, IDIAP, 2000.

[12] George R. Doddington, Mark A. Przybocki, Alvin F. Martin, and Douglas A. Reynolds, "The NIST speaker recognition evaluation–overview, methodology, systems, results, perspective," *Speech Communication*, vol. 31, no. 2-3, pp. 225–254, 2000.