Speaker Verification Using Support Vector Machines

S. Raghavan, G. Lazarou and J. Picone
Center for Advanced Vehicular Systems
Mississippi State University
{raghavan, glaz, picone}@cavs.msstate.edu

Abstract

Support vector machines (SVM) have become a very popular pattern recognition algorithm for speech processing. In this paper we describe an application of SVMs to speaker verification. Traditionally speaker verification systems have used hidden Markov models (HMM) and Gaussian mixture models (GMM). These classifiers are based on generative models and are prone to overfitting. They do not directly optimize discrimination. SVMs, which are based on the principle of structural risk minimization, consist of binary classifiers that maximize the margin between two classes. The power of SVMs lie in their ability to transform data to a higher dimensional space and to construct a linear binary classifier in this space. Experiments were conducted on the NIST 2003 speaker recognition evaluation dataset. The SVM training was made computationally feasible by selecting only a small subset of vectors for building the out-of-class data. The results obtained using the SVMs showed a 9% absolute improvement in equal error rate and a 33% relative improvement in minimum detection cost function when compared to a comparable HMM baseline system.

1. Introduction

Speaker verification is a simple pattern recognition task that involves assessing the distance between a speaker claiming an identity and a stored model representing that identify [1]. A speaker verification system has to perform two main tasks: enrollment and verification. Enrollment is the task of constructing a speaker model that captures the spectral and temporal variations of the speaker. This enrollment data is used to build a model that will be used to authenticate the speaker during the verification phase. Speaker verification places a great emphasis on the acoustic modeling part of the speech processing task.

During verification the input speech from the test subject is matched against the acoustic model. A likelihood representing the distance between the model and the measured observations is used to make a decision about the speaker's identity. The test speaker is accepted or rejected based on a threshold set for the acoustic likelihood. A likelihood threshold is empirically determined such that a required trade off between false alarms and miss detections is obtained. Conventional speaker verification systems used an hidden Markov model (HMM) based classifier with Gaussian mixture models representing the emission probabilities of the HMM [2]. These systems use a generative model for the acoustic model and are prone to overfitting. Such models do not directly maximize discrimination [3].

Factors such as channel conditions, ambient noise, type of microphone etc. affect the performance of any pattern recognition system. Noise robustness can be achieved either by making the features robust to noise or by building a model that is robust to noisy features. SVMs take the later approach.

SVMs are a popular approach to classification that are based on the principles of structural risk minimization. SVMs have recently been used for building large vocabulary continuous speech recognition (LVCSR) systems [3], and have shown good promise as a basis for discriminative training. SVMs have also been used for language and speaker identification applications recently [4]. SVMs are binary classifiers and are naturally suited for tasks such as speaker verification. In this paper we describe a basic speaker verification system using SVMs. The NIST 2003 Eval data was used for the experiments. The results obtained using SVMs showed a 9% absolute improvement in equal error rate and a 33% relative improvement in minimum detection cost function when compared to an HMM-based baseline system.

2. Support Vector Machines

SVMs are trained in a similar fashion as neural networks. Neural networks have the ability to learn complex nonlinear decision boundaries from the data, but lack generalization because they are prone to overfitting [3]. SVMs are based on the principles of structural risk minimization, and hence have good generalization ability when compared to HMM and neural network based classifiers. SVMs in their simplest form are hyperplane classifiers [3]. The optimal hyperplane is the one that maximizes the margin between in-class and out-of-class data and at the same time reduce the empirical risk [3]. The power of SVMs lies in their ability to transform data to a higher dimensional space and construct a linear binary classifier in the higher dimensional space [3]. A linear hyperplane in the higher dimensional space transforms to a complex nonlinear decision region in the input feature space.

Let x be a set of input feature vectors, and y be the class labels for the feature vectors, this can be represented as tuples $\{x_i, y_i\}$ where $i = \{1,...,l\}$ and $y = \pm 1$. The points lying on the decision surface satisfy Equation 1.

$$w \bullet x + b = 0\ . \qquad (1)$$

where w is the normal to the decision region, and b is the distance of the hyperplane from the origin. The generalization power of SVMs lies in its ability to work as a soft decision classifier [3]. This is accommodated by the addition of a slack variable as shown below:

$$x_i \bullet w + b \geq 1 - \xi_i \;\; for \;\; y_i = +1\ . \qquad (2)$$

$$x_i \bullet w + b \geq 1 + \xi_i \;\; for \;\; y_i = -1\ . \qquad (3)$$

$$\xi_i \geq 0 \quad \forall i, . \qquad (4)$$

where ξ_i is a slack variable. Once the hyperplane using soft margin is obtained, the training examples must satisfy Equations 2 and 3. All points satisfying Equation 2 lie along the hyperplane H1 while those satisfying Equation 2 lie along H2 as shown in Figure 1. Training and optimization of the classifier for speech signals is covered extensively in [3].

Thus far we have described an approach which can build a linear classifier on either overlapping or non-overlapping data, but in real world the data is not linearly separable. This is where a kernel function is employed to transform the training data to a higher dimensional space. Radial basis functions (RBF) are the most popular choice for a kernel function for speech processing [3]. An RBF kernel is defined as:

$$K(x, y) = \exp\{-\gamma \mid x - y \mid^2\}\ . \qquad (5)$$

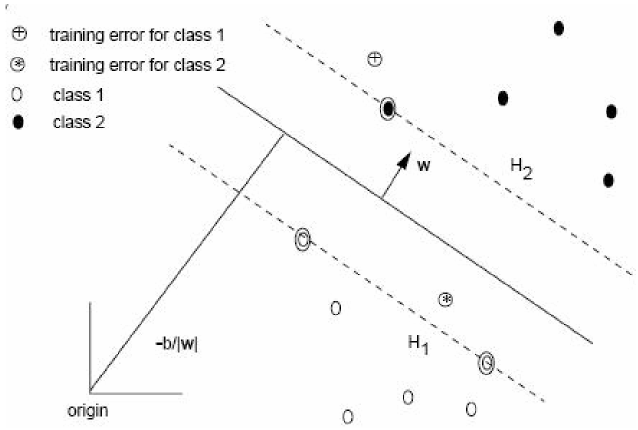

Figure 1. Examples of a soft margin classifier which is robust to training errors

where γ is the kernel parameter that influences the extent of non-linearity of the decision surface in the feature space. While testing, the SVM classifier returns a distance from the hyperplane. The functional form of the kernel based SVM classifier is defined as:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b\ , \qquad (6)$$

where α holds the weights corresponding to every sample point in the feature space, these weights are the lagrangian multipliers [3]. Training examples with non-zero multipliers are referred to as support vectors because they define the classifier's decision surface. The complexity of the classifier is loosely related to the number of support vectors used in the classifier.

3. SVM Based Speaker Verification System

A block diagram of a basic speaker verification system is shown in Figure 2. By using a discriminative classifier such as SVM we can combine the task of building impostor model and speaker model into a single training process. The feature extraction block extracts salient features from the raw speech data. The features must capture the temporal and spectral variations of the speech signal. The most popular features are the mel frequency cepstral coefficients (MFCCs) [5]. The MFCCs serve as inputs to the classifier.

Training of the SVM classifier requires information about in-class data and out-of-class data. The training data should be split in a manner such that every speaker is associated with a pair of data sets, one for in-class data and the other representing the out-of-class data. The in-class data is obtained from the speaker's MFCC vectors while the out of class data comprises of the MFCC vectors of all the remaining speakers in the training data set. This is equivalent to building an impostor model for HMM based systems.

The SVM algorithm finds a suitable decision boundary that separates the in-class and out of class features. The functional form in Equation 6 is obtained after training, and during verification the test MFCC vectors are plugged into the model corresponding to the claimed identity and a distance measure is computed as output. A suitable threshold is used to decide the acceptance and rejection of a

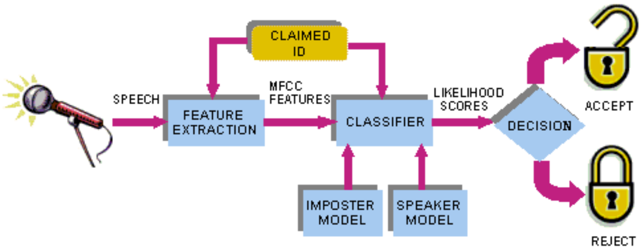

Figure 2. Basic blocks of a speaker verification system

Table 1. Minimum DCF as a function of γ

Gamma(C=50)	Min DCF
0.010	0.2125
0.015	0.2168
0.019	0.1406
0.030	0.2305

speaker. The main parameters that have to be tuned are the kernel parameter and the penalty value which is a critical factor in preventing training errors [3].

The speaker verification system described above is a very basic implementation using SVMs. However, as shown in the next section, this technique is promising since its performance is comparable to a baseline HMM system and uses only a small subset of the original training data. It is possible to further improve performance using a hybrid system that combines SVMs and HMMs. The final distance measures can be converted to a posterior probability using the sigmoid function and used as emission probabilities for an HMM. Hybrid systems have proven to give reasonable improvements in performance for speech recognition tasks [3].

4. Experimental Setup and Analysis

NIST 2003 speaker recognition evaluation data was used for all the experiments described in this section [6]. All utterances in the development data set were approximately 2 minutes in length. The development set contained 60 utterances for training and 78 utterances for testing. These utterances were taken from the Switchboard corpus. A standard 39-dimension MFCC feature vector was used.

As described in Section 3, the SVM classifier requires information about in-class and out-of class data for every speaker in the training set. Suppose a model ‘x’ has to be trained for utterance ‘x’, in which case the in-class data for training will contain all the 39 dimensional MFCC feature set for the utterance ‘x’, and the out-of-class data is obtained by randomly picking “n” feature vectors from all the remaining utterances in the training data set. The size of “n” was determined in such a way that the out-of-class data had twice the number of MFCC vectors when compared to the in-class data. This is an approximation and hence will not contain all the information required to represent the true out-of-class distribution, but this sort of approximation was necessary to make the SVM training computationally feasible. Hence, it has to be kept in mind that the performance of this system is based on classifiers that were exposed to only a small subset of data during training.

During testing, the test MFCC vectors are used as input to compute the distance using the functional form of the model shown in Equation 6. A distance is computed for every single test vector, and finally an average distance for the entire feature vector set is computed. The average distance is used for final decision making. An ideal decision threshold is zero for SVM classifiers, but for speaker verification tasks we can determine a threshold where the detection cost function is minimum (DCF) [6].

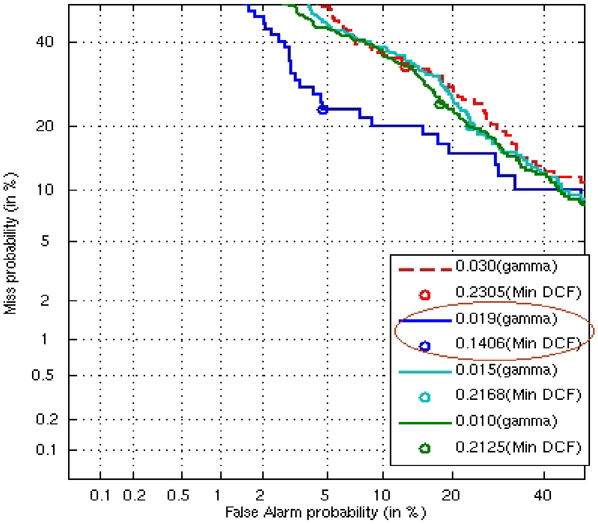

Figure 3. DET curves for various values of the RBF kernel parameter γ

The first set of experiments was conducted to determine the optimum value of γ for the RBF kernel. It was observed that for γ values between 2.5 to 0.02 there was very little variation in the distance scores for the test utterances. Performance was stable between 0.03 and 0.01 as shown in the DET [7] curves of Figure 3.

The minimum DCF points were obtained for each of these curves and it was observed that for γ=0.019 we obtained the lowest minimum DCF. The minimum DCF for various values of γ are shown in Table 1. The Equal Error Rate was 16% with a γ of 0.019 and the penalty parameter set to 50. It can be observed from the DET plot that there is very marginal change in performance for changes in the γ values in the selected range. The most significant improvement in performance was observed only with a γ value of 0.019 and the effect of this improvement also reflected in an improvement in minimum DCF value as shown in Table 1.

Another parameter that had to be tuned was the penalty value that accounts for the training errors [3]. The penalty value was varied from 10 to 100 and no significant change in the in Min DCF value was observed. Hence a mid range value of 50 was chosen.

We compared the results obtained on the SVM based speaker verification system with the baseline HMM system. The baseline system used 16-mixture Gaussians as the underlying classifier. An impostor model was trained on all the utterances in the development train set while the speaker models were built using the corresponding speaker utterance and constructing 16-mixture Gaussians. During testing, a likelihood ratio was computed between the speaker model and the impostor model. The likelihood ratio was defined as:

$$LR = \log P(x \mid sp\_\text{mod}) \\ - \log P(x \mid imp\_\text{mod}) \; . \qquad (7)$$

where LR is the likelihood ratio, “x” is the input test vector, “sp_mod” and “imp_mod” are the speaker and impostor models respectively. The equal error rate obtained on the HMM baseline system was close to 25% and the Min DCF was 0.2124. A comparative DET plot between SVM and baseline HMM system is shown in Figure 4 and their comparative performances are listed in Table 2.

5. Conclusions

In this paper we applied SVMs to the task of speaker verification. The SVM-based speaker verification system was compared with a baseline HMM system and a reasonable improvement in performance was noted with the SVM system. The EER improved by 9% absolute and the min DCF value improved by 33% relative. These improvements suggest that SVMs are better suited for tasks such as speaker verification which requires a simple nonlinear binary classifier. The effect of RBF kernel parameter gamma and the training error penalty was also analyzed. Overall system performance was not extremely sensitive to changes in these parameters.

Future work will include combining both the HMM and SVM systems into a single architecture. In order to make SVM training computationally feasible, an experimental approach was designed that leveraged a subsampling strategy. A more comprehensive strategy based on principles of active learning is under development and is expected to significantly improve performance over the subsampling strategy.

6. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0414450. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

7. References

[1] J.P. Campbell, “Speaker Recognition: A Tutorial,” *Proceedings of IEEE*, pp. 1437-1462, September 1997.

[2] D. A. Reynolds, “Speaker identification and verification using Gaussian mixture speaker models,” *Speech Commun*, pp. 91–108, 1995.

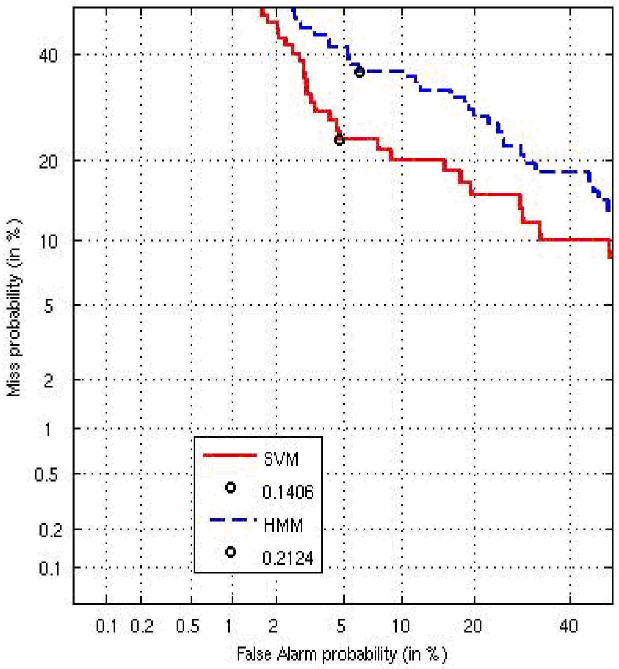

Figure 4. A comparison of HMM and SVM performance

Table 2. Comparision of SVM based speaker verification system with the baseline HMM system

HMM	SVM
EER 25%	EER 16%
Min DCF 0.2124	Min DCF 0.1406

[3] A. Ganapathiraju, “Support Vector Machines for Speech Recognition,” *Ph.D. Dissertation*, Department of Electrical and Computer Engineering, Mississippi State University, January 2002.

[4] W. M. Campbell, E. Singer, P. A. Torres-Carrasquillo, and D. A. Reynolds, *“Language Recognition with Support Vector Machines,” Proc. Odyssey: The Speaker and Language Recognition Workshop,* Toledo, Spain, ISCA, pp. 41-44, 31 June 2004.

[5] J. Picone, “Signal Modeling Techniques in Speech Recognition,” *IEEE Proceedings*, vol. 81, no. 9, pp. 1215-1247, September 1993.

[6] “NIST 2003 Speaker Recognition Evaluation Plan,” *http://www.nist.gov/speech/tests/spk/2003/doc/2003-spkrec-evalplan-v2.2.pdf.*

[7] A. Martin, G. Doddington, M. Ordowski, and M. Przybocki, “The DET curve in assessment of detection task performance,”. *In Proceedings of EuroSpeech*, volume 4, pages 1895--1898, 1997.