

# An Approach to Discriminative Classification Using Generative Models in Speaker Recognition: EE227A Project

Kofi Boakye

December 16, 2004

## 1 Introduction

For the task of data classification, two main approaches exist and are commonly employed: generative and discriminative. In the generative case, we assume that the data was generated according to some underlying distribution and seek to estimate it. More specifically, for a class  $Y$  and data  $X$  we estimate  $P(Y)$  and  $P(X|Y)$  to obtain a surrogate to  $P(Y|X)$ . In the discriminative case, no direct attempts are made to model the underlying distribution of the data. Rather  $P(Y|X)$  is estimated or in some cases simply a separating hyperplane is obtained and no probabilistic models are developed at all. Naturally, there exist tradeoffs between these two methods [1] and the specific task may dictate which is more beneficial.

In the area of speaker recognition, the dominant techniques rely on generative modeling approaches. Specifically, the state of the art involves the use of a Gaussian Mixture Model (GMM) [2] or possibly a Hidden Markov Model (HMM) [3] for speech acoustic features. Recently, however, discriminative modeling, specifically the use of Support Vector Machines (SVMs), has been shown to have good performance as well on this task [4]. A natural question, then, is whether both approaches can simultaneously be incorporated into a speaker recognition system

This project sought to illustrate that the answer to the above is indeed “yes” and also examined the performance of such a hybrid system. The system uses HMM model parameters as features for an SVM classifier. HMMs of specific keywords and phrases are adapted to each speaker from a Universal Background Model (UBM) and the means of the Gaussian mixtures which model each HMM state are used as the SVM features.

This paper is organized as follows: section 2 describes the basic speaker recognition task; sections 3 and 4 describe a keyword HMM system and a hybrid HMM/SVM system, respectively, that will be compared; section 5 explains the theory of the Support Vector Machine; section 6 outlines the basic procedure used; details on the evaluation process are given in section 7; results are presented in section 8; and conclusions and future work are discussed in section 9.

## 2 The Speaker Recognition Task

The dominant framework for speaker recognition is that of a detection task. The objective is to determine whether a putative target is the speaker in a given test utterance. As previously mentioned the Gaussian Mixture Model approach yields high performance. The

approach is as follows: Let  $X = \{x_i : i = 1, \dots, N\}$  be the sequence of speech feature vectors for a test utterance. For a given test/target pair we look at the log-likelihood ratio of the data given the target model and the data given what is referred to as a Universal Background Model (UBM):

$$LLR(X) = \log P(X|S) - \log P(X|UBM) \quad (1)$$

where  $S$  refers to the speaker model. The UBM is obtained by training a Gaussian Mixture Model with speech from a large collection of speakers that are not within the test population. The speaker-specific model is generated by adaptation of the UBM with the speaker's training data, in the form of spoken conversation sides (details provided in section 7). The benefit of this approach is that fuller speaker models can be obtained than if a model were to be generated from the speaker's training data alone.

If the log-likelihood ratio score exceeds a specified threshold, the test utterance is accepted as having been produced by the putative target speaker. If not, the speaker assignment is rejected. For a collection of trials one can sweep through a series of thresholds, each of them representing an operating point for the system. For each point, a false alarm (FA) and missed detection (MD) rate can be computed. The overall system performance is then represented graphically by a Detection Error Tradeoff (DET) curve which plots false alarm probability versus miss probability on a normal deviate scale. This is done to have the curves appear linear (as opposed to ROC curves which could also be used) which assists in comparison of systems. A summary statistic, the Equal Error Rate (EER), is often used and represents the operating point at which the probability of false alarm equals the probability of missed detection.

### 3 Keyword HMM System

Though the GMM approach works well, recent work [3] involving Hidden Markov Models of specific keywords has yielded systems with good performance. This approach is motivated by the observations that 1) GMMs (incorrectly) assume that speech feature vectors are independent and thus fail to take advantage of some sequential information in speech; and 2) Certain words/phrases possess high speaker discriminative characteristics than others.

The system works as follows: the keywords for which the models exist are identified and extracted from the speech stream based on alignment information provided by an Automatic Speech Recognizer (ASR). For each keyword a UBM is trained using speakers not in the test population, similar to the GMM case. Also similarly, a speaker-specific keyword HMM is obtained by adapting the UBM version with speaker training data. The adaptation procedure used is Maximum A Posteriori (MAP) adaptation of the means. The output distribution for each state in the HMM is a mixture of 4 Gaussians and the MAP adaptation means are given by:

$$\hat{\mu}_{jm} = \frac{N_{jm}}{N_{jm} + \tau} \bar{\mu}_{jm} + \frac{\tau}{N_{jm} + \tau} \mu_{jm} \quad (2)$$

where  $\tau$  is a weighting of the a priori knowledge to the adaptation speech,  $N$  is the occupation likelihood of the adaptation data,  $\mu_{jm}$  is the speaker independent (i.e., UBM) mean and  $\bar{\mu}_{jm}$  is the mean of the observed adaptation data. When performing a test, all instances of each keyword are found and extracted, again using ASR. Each speech segment is scored by its appropriate target speaker and UBM HMM. These scores represent the logprobabilities

of the sequences having been generated by the respective HMMs. A log-likelihood ratio score is obtained by summing the target scores, subtracting the sum of the UBM scores, and normalizing by the total number of speech feature vectors for the sequences. Figure 1 shows a schematic of the system.

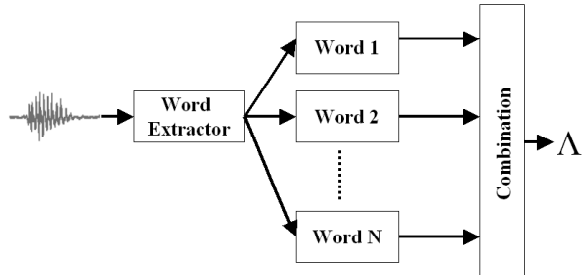


Figure 1: Keyword system.

This technique has been shown to work quite well, with the advantage that it looks at only a small subset of the total data, as only feature vectors corresponding to the keywords are utilized. Presently the system uses only about 10% of the total data available.

## 4 Hybrid HMM/SVM System

From equation (1), we can view the log-likelihood ratio score as a kind of measure between models (the target and UBM) with respect to a fixed utterance  $X$ . Models which are “farther” from the UBM tend to be more reliably detected. Adopting this framework, one might also find it reasonable to utilize other measures between models. One alternative is distances in the model parameter space. From this we arrive at the hybrid HMM/SVM system.

For this system, each speaker model is represented as a vector of parameters; specifically, the concatenation of the mean vectors of the Gaussian mixtures in the appropriate HMM. The speaker model is here, too, obtained by adaptation with the speaker training data. The difference in this case is that multiple models are generated for each speaker, one for each conversation in the training data, for a total of 8. The derived vectors serve as positive examples for an SVM classifier. The negative examples are vectors obtained from adapting to the conversations used for training the UBM. This is done for each keyword HMM.

For testing, the keyword HMMs are adapted using the test data to produce a data point for each SVM classifier for the target speaker model. Each classifier then generates a score and the scores are then combined to yield a final score.

## 5 Support Vector Machines

The Support Vector Machine is a discriminative classifier of increasing utilization in speaker recognition and other machine learning fields. This binary classifier seeks to find the separating hyperplane that maximizes the margin between the two classes in a (potentially) higher-dimension space than that of the original features. The margin is defined to be the minimal distance of a sample to the decision surface. The optimization problem is formally

stated as [6]:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \tag{3}$$

Here  $w$  is the weight vector which appears in the decision function  $f(\mathbf{x}) = (w \cdot \Phi(\mathbf{x})) + b$  for the classifier. This quadratic program cannot be solved directly, as  $w$  lies in the higher dimensional space and this space is only accessible via inner products obtained by a kernel function  $k(\mathbf{x}_i, \mathbf{x}_j)$ . As in many cases, a solution is obtained by looking at the dual problem:

$$\begin{aligned} \max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ \text{subject to } \alpha_i \geq 0, \quad i = 1, \dots, n \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned} \tag{4}$$

which is a quadratic optimization problem that is readily solvable. The above formulation presupposes separability between the two classes, which is often not obtainable for real data owing to factors such as noise. This can be addressed by the addition of “slack” variables  $\xi_i$  which allow for points to lie inside the margin area. The new problem is:

$$\begin{aligned} \max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ \text{subject to } 0 \leq \alpha_i \leq C, \quad i = 1, \dots, n \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned} \tag{5}$$

One of the appealing aspects of the SVM is that the solution of the optimization problem is sparse in  $\alpha$ . Specifically, the Karush-Kuhn-Tucker conditions for the problem indicate that only  $\alpha_i$  corresponding to  $\mathbf{x}_i$  that are on or inside the margin area are nonzero.

## 6 Procedure

The initial part of the process involves training of the background HMMs. The keywords for which models were trained were as follows:  $\{actually, anyway, i\_know, i\_mean, i\_think, i\_see, like, now, okay, right, see, uh, uhuh, um, well, yeah, yep, you\_know, you\_see\}$ . These words/phrases were selected because of their high frequency in conversational speech as well as because they may possess strong-speaker distinctive characteristics owing to their habitual, spontaneous behavior. The HMMs were trained using Hidden Markov Model Toolkit (HTK), a common toolkit for speech and speaker recognition.

After training the background HMMs, they are then adapted and their parameters are “vectorized” as described in section 4. Scoring also occurs as previously described. A complicating issue exists in that for each trial, 19 scores from the 19 classifiers are generated and must be combined into a single score. This is especially a problem because the scores do not represent probabilities or likelihoods as in other cases, so the principled combination techniques developed for other speaker recognition systems cannot be applied. Ultimately three different techniques were explored for this project: simple linear combination (normalized by the number of classifiers), maximum entropy classification for the scores, and another SVM which served as a score combiner. A maximum entropy classifier is yet another discriminative classifier, which seeks to find  $P(Y|X)$  which maximizes the conditional entropy:

$$H(P) = - \sum_{x,y} \tilde{P}(X) P(Y|X) \log P(Y|X) \tag{6}$$

for features  $f_i$  which are typically binary in nature. The solution to this optimization problem is:

$$P^*(Y|X) = Z(x) \exp\left(\sum_i \lambda_i f_i(x, y)\right) \quad (7)$$

with  $Z(x) = \sum_y \exp(\sum_i \lambda_i f_i(x, y))$ . Training of the classifier amounts to determining the  $\lambda_i$  and is done using techniques such as Generalized Iterative Scaling (GIS). The binary features used in this case were the decisions by each SVM classifier as well as the absence of a classifier decision (which occurs when the keyword does not appear in the test utterances).

The training and the scoring of the SVM classifiers was done using SVM<sup>light</sup>, a publicly available implementation of Support Vector Machines. The package allows for the modification of various parameters, the most pertinent being the kernel function  $k(\mathbf{x}_i, \mathbf{x}_j)$ , the cost-factor (i.e., the factor by which positive misclassifications outweigh negative ones), and the regularization constant  $C$ . From experimentation it was shown that a linear kernel ( $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$ ) along with a cost-factor of one performed best. Extensive analysis was not performed on the regularization constant, which was set to one as well.

## 7 Evaluation

The hybrid HMM/SVM system was evaluated within the framework of the Extended Data Task of the 2001 NIST Speaker Recognition Evaluation [5]. In the NIST evaluation speaker models are trained using 1,2,4,8, and 16 complete conversation sides and are subsequently tested on a complete conversation side. The conversation sides are approximately 2.5 minutes in length. The Extended Data Task uses the Switchboard I conversational telephone corpus in a cross-validation process whereby the data is partitioned into 6 sections of approximately equal size and each partition is tested independently; when one partition is being tested, data from the others can only be used for background models or normalization. For this project, testing was performed on splits 1-3 and the three others were reserved for providing background model data. In addition, models were trained using 8 conversation sides as previously mentioned.

## 8 Results

The Equal Error Rate results for the experiments can be found in table 1 and the overall Detection Error Tradeoff curves are pictured in figures 2 and 3. The first item of note is the comparable performance of the linear and SVM combination techniques along with the poor performance of the maximum entropy combiner. The maximum entropy combiner's lag in performance is in some ways to be expected: the binarization of the features amounts to removing information which could be used to aid classification.

The comparison of the best hybrid system (i.e., the one with the SVM combiner) to the keyword system also yields interesting observations. The keyword system significantly outperforms the hybrid one. Note, though, that there is a crossover point in the low false alarm region where the hybrid system begins to have the better performance.

## 9 Conclusions and Future Work

The objective of the project was to demonstrate that both a generative and discriminative approach could simultaneously be applied to a classification task, specifically that of speaker

Combination scheme	EER (%)
Maxent	4.15
Linear	1.40
SVM	1.29
System	EER (%)
Keyword	1.08
HMM/SVM	1.29

Table 1: System performance. The first three entries give results for the various combination methods and the last two entries are a comparison of the keyword system and the best HMM/SVM system..

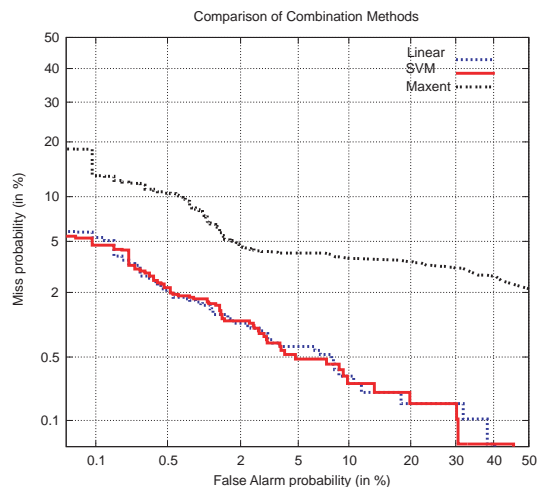


Figure 2: Comparison of different score combination techniques

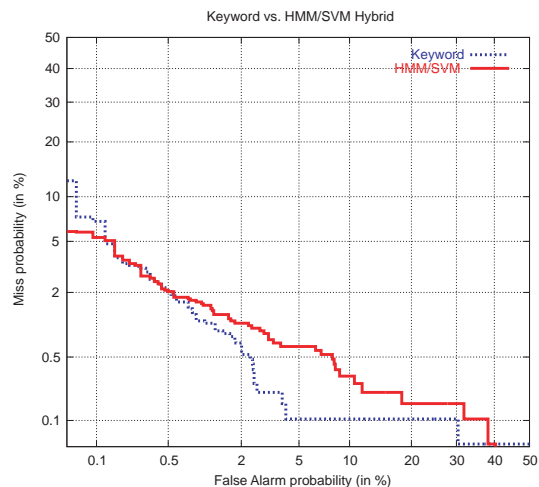


Figure 3: Keyword system versus hybrid HMM/SVM system

recognition. From the results presented, this indeed appears to be the case. Further still, the results suggest that the hybrid approach can be applied rather successfully, obtaining performance somewhat competitive with the state-of-the-art generative classification systems.

It still remains to be seen whether the hybrid method could be more effective than either of the two in isolation. For the present task and framework, there exist possible future avenues of exploration. One issue in this work is that the SVMs are not necessarily optimized to minimize the EER. Optimization of the cost-factor and regularization constant using either held-out data or cross-validation could change that. Another possible alteration lies in the MAP adaptation procedure. Modification of the weighting  $\tau$  in equation (2) affects the amount of shifting of parameters, which may influence the SVM training and classification.

## References

- [1] G. Bouchard, B. Triggs, “The Trade-off Between Generative and Discriminative Classifiers,” *Proc. IASC International Symposium on Computational Statistics*, 2004.
- [2] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, “Speaker Verification using Adapted Gaussian Mixture Models,” *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19-41, 2000.
- [3] K. Boakye and B. Peskin, “Text-Constrained Speaker Recognition on a Text-Independent Task,” *Proc. Odyssey 2004 - The Speaker and Language Recognition Workshop*, 2004.
- [4] W.M. Campbell, J.P. Campbell, D.A. Reynolds, D.A. Jones, and T.R. Leek, “Phonetic Speaker Recognition with Support Vector Machines” *Proc. Neural Information Processing Systems*, 2003.
- [5] NIST 2001 Speaker Recognition website:  
<http://www.nist.gov/speech/tests/spk/2001>.
- [6] K. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, “An Introduction to Kernel-Based Algorithms,” *IEEE Trans. on Neural Networks*, vol. 12, no. 2, pp. 181-202, 2001.