Speaker Verification with SVM and DGMM

Tales Imbiriba, Rafael Marinho, Adalbery Castro and Aldebaro Klautau

***Abstract*— The Gaussian mixture model (GMM) is the main technique used in speaker recognition systems. However, in tasks other than speaker recognition, GMM is often outperformed by modern classifiers, such as support vector machines (SVM). This work seeks a better understanding of the reasons for discriminative classifiers not being as successful in speaker recognition as in other applications. This is done by comparing GMM and a novel technique called discriminative GMM, which is similar to SVM in many aspects. Simulation results using the IME corpus show that both SVM and DGMM can improve the performance compared to GMM, and indicate that a proper model selection is essential to make them competitive in speaker verification.**

I. Introduction

The Gaussian mixture model (GMM) is the main technique used in speaker recognition systems. The GMM is trained through *generative* learning, which is often outperformed by modern *discriminative* learning techniques [1], [2]. However, applying discriminative learning to speaker recognition has proven to be a tricky task [3]. Powerful techniques, such as support vector machines (SVM) sometimes perform poorly in speaker recognition, when compared to GMM. Such results puzzle researchers that work in machine learning problems other than speaker recognition, where SVM often often outperform GMM by a comfortable margin (see, e.g., [4]).

This work tries to achieve a better understanding of this issue. Instead of seeking the best results for a specific task, it addresses, for example, the problems that led us to obtain poor results for SVM in [5]. The approach we take is to compare generative and discriminative learning, by contrasting GMM and a similar classifier, called *discriminative GMM* [4]. Besides being a novel technique, applying DGMM to speaker verification sheds some light on SVM because both are discriminative learning techniques.

Another contribution of this work is to continue promoting the adoption of the IME 2002 corpus[1], which is a Brazilian Portuguese corpus for speaker recognition. It has been made available free of charge to several research groups by the Signal Processing Group at IME (http://www.ime.eb.br/∼labvoz/), and is a very useful resource for researchers working in speaker recognition. Since now, most of the research in speaker recognition in Brazil is conducted with proprietary (and relatively small) datasets. The IME corpus provides an opportunity to change this situation, and promote the comparison of results obtained by different groups given that, besides the corpus, there are good open source softwares for speaker recognition [5].

This paper is organized as follows. In Section 2 we discuss the frame-based architecture for speaker verification, a formalism that helps to understand the role of classifiers in this application. Section 3 discusses classifiers, with emphasis on contrasting GMM and DGMM, two Bayes classifiers that differ in the training procedure. Experimental results are presented in Section 4, which is followed by the conclusions.

The authors are with the Signal Processing Laboratory (LaPS), DEEC, UFPA, 66075-110, Belém, Pará, Brazil, http://www.laps.ufpa.br. Emails: {tales,rafael,adalbery,aldebaro}@deec.ufpa.br. A preliminary version of this work appeared at the workshop TIL/SBC'2005, São Leopoldo, Brazil.

[1]Work supported by FAPERJ, Brazil, under grant number E-26/171.307/2001.

II. The Frame-Based Architecture for Speaker Verification

Speaker recognition is the process of automatically recognizing who is speaking, and can be split into speaker identification and speaker verification. Speaker identification determines which registered speaker provides a given utterance from amongst a set of known speakers. Speaker verification is a binary problem, in which the system accepts or rejects the identity claim of a speaker. This work deals exclusively with verification.

The speaker recognition problem is closely related to the conventional supervised classification. Hence, we start by providing few related definitions. In such framework, one is given a *training set* $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ containing N *examples*, which are independently and identically distributed (iid) samples from an unknown but fixed distribution $P(\mathbf{x}, y)$. Each example $(\mathbf{x}, y)$ consists of a vector $\mathbf{x} \in \mathcal{X}^L$ of dimension L, called *instance*, and a *label* $y \in \{1, \ldots, Y\}$. A *classifier* is a mapping $F : \mathcal{X}^L \rightarrow \{1, \ldots, Y\}$. Of special interest are binary classifiers, for which $Y = 2$, and for mathematical convenience, sometimes the labels are $y \in \{-1, 1\}$. Some classifiers are able to provide *confidence-valued scores* $f_i(\mathbf{x})$ for each class $i = 1, \ldots, Y$. Commonly, these classifiers use the max-wins rule $F(\mathbf{x}) = \arg\max_i f_i(\mathbf{x})$. When these are binary classifiers, only a single score $f(\mathbf{x}) \in \mathbb{R}$ is needed. For example, if $y \in \{-1, 1\}$, the final decision can be simply the sign of the score, i.e., $F(\mathbf{x}) = \text{sign}(f(\mathbf{x}))$.

Contrasting to classifiers, in speaker recognition systems, the input is a matrix $\mathbf{X} = \{\mathbf{x}_t\}$, $\mathbf{X} \in \mathcal{X}^{T \times Q}$, which corresponds to a segment of speech parameterized by the *front end* stage [6]. The number T of rows is the number of *frames* (or segments) of speech, and Q (columns) represents the number of parameters of each frame. If T is fixed (say, $T = 1000$ frames), $\mathbf{X}$ could be turned into a vector of dimension $L = T \times Q$, and one would end up with a conventional classification problem. However, in text-independent speaker verification, any comparison between elements of two such vectors could fail, because they would eventually represent different sounds. Hence, verification systems often adopt a

frame-based architecture[2](see, e.g., [5]), which is similar to, but does not exactly match a conventional classifier F.

The frame-based verification system is a mapping $G : \mathcal{X}^{T\times Q} \rightarrow \{-1, 1\}$, where -1 and 1 correspond to speaker rejection and acceptance, respectively. More specifically,

$$G(\mathbf{X}) = \text{sign}(g(\mathbf{X}) - \lambda),$$

where $g(\mathbf{X})$ is a score, typically provided by a *model* corresponding to the claimed identity, and λ is a threshold that allows to tradeoff the false rejection and false acceptance rates. In this architecture, $g(\mathbf{X})$ is calculated by invoking repeatedly a conventional classifier that returns a confidence-valued score $f(\mathbf{x}_t)$, i.e., $g(\mathbf{X}) = \sum_{t=1}^{T} f(\mathbf{x}_t)$ or, eventually

$$g(\mathbf{X}) = \sum_{t=1}^{T} \log(f(\mathbf{x}_t)),$$

as in the case of adopting GMMs for computing $f(\mathbf{x})$.

There are many learning algorithms for training classifiers (see, e.g., [8]). Roughly speaking, all of them can be used in frame-based speaker verification. The next sections discuss some of the most prominent classifiers, and pros and cons of their adoption in this application.

III. Classifiers for Frame-Based Verification

GMM, which is a special case of a Bayes classifier, is the most popular classifier for speaker verification. However, in many other tasks, GMM is outperformed by other classifiers. Among these competitors, of special interest are the ones based on *kernel learning*, such as SVM [9]. Notice that a Bayes classifier is called by some authors a "kernel" classifier (see, e.g., page 188 in [8]). However, by kernel classifier we mean the ones obtained through kernel learning, as defined, e.g., in [10].

In spite of the good performance achieved by kernel methods (and other discriminative techniques) in several tasks [10], adopting it in speaker verification remains a challenge. For example, GMM outperformed SVM in some of our preliminary experiments [5]. Such conclusion puzzles machine learning experts, but speech verification has idiosyncrasies that require better understanding for the successful adoption of discriminative learning. This work is a small step towards this goal. To make the simulations manageable, it deals exclusively with SVM, which is the most popular kernel classifier, and two Bayes classifiers: GMM and DGMM. We start by discussing SVM and afterwards we conduct a thorough review of Bayes classifiers.

A. SVM

SVM (and other kernel methods) can be related to regularized function estimation in a reproducing kernel Hilbert space (RKHS) [11]. One wants to find the function f that minimizes

$$\frac{1}{N}\sum_{n=1}^{N} L(f(\mathbf{x}_n), y_n) + \lambda||f||^2_{\mathcal{H}_\mathcal{K}}, \quad (1)$$

where $\mathcal{H}_\mathcal{K}$ is the RKHS generated by the kernel $\mathcal{K}$, $f = h + b$, $h \in \mathcal{H}_\mathcal{K}$, $b \in \mathbb{R}$ and $L(f(\mathbf{x}_n), y_n)$ is a loss function.

The solution to the optimization problem described in Equation 1, as given by the *representer* theorem [12], is

$$f(\mathbf{x}) = \sum_{n=1}^{N} \omega_n \mathcal{K}(\mathbf{x}, \mathbf{x}_n) + b. \quad (2)$$

This expression indicates that SVM and related classifiers are *example-based* [10], i.e., f is given in terms of the training examples $\mathbf{x}_n$. In other words, assuming a Gaussian kernel $\mathcal{K}(\mathbf{x}, \mathbf{x}') = e^{-\gamma||\mathbf{x}-\mathbf{x}'||^2}$, the mean of a Gaussian is restricted to be a training example $\mathbf{x}_n$.

Some examples $\mathbf{x}_n$ may not be used in the final solution (e.g., the learning procedure may have assigned $\omega_n = 0$). We call *support vectors* the examples that are actually used in the final solution. For saving memory and computations in the test stage, it is convenient to learn a sparse f, with few support vectors.

In speaker verification, the number of support vectors can be as high as 90% of the training set. There are several algorithms for SVM training, but most of them provide a parameter to influence the number of support vectors. In this work, the "complexity" parameter C was adopted [10].

The next subsection discusses Bayes classifiers, for which the number of Gaussians (equivalent to the number of support vectors when SVM uses a Gaussian kernel) is specified before training the classifier.

B. Generative and discriminative Bayes classifiers

Bayes classifiers are ideal to contrast generative and discriminative learning applied to speaker verification. Throughout this work, the nomenclature follows the one used in [13], where[3] $P(y|\mathbf{x})$, $P(\mathbf{x}|y)$, $P(y)$ and $P(\mathbf{x})$ are called *posterior*, *likelihood*, *prior* and *evidence*, respectively, and are related through Bayes' rule

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}. \quad (3)$$

Bayes classifiers attempt to select the label $\arg\max_{y=1,\ldots,Y} P(\mathbf{x}|y)P(y)$, which maximizes the posterior probability. However, neither $P(y)$, nor $P(\mathbf{x}|y)$ is known, hence the classifiers use estimates $\hat{P}(y)$ and $\hat{P}(\mathbf{x}|y)$ and maximize

$$F(\mathbf{x}) = \arg\max_{y=1,\ldots,Y} \hat{P}(\mathbf{x}|y)\hat{P}(y). \quad (4)$$

In most cases, the prior $P(y)$ can be reliably estimated by counting the labels in the training set, and we assume here that $\hat{P}(y) = P(y)$. Estimating $\hat{P}(\mathbf{x}|y)$ is more difficult. Hence, classifiers typically assume a parametric distribution $\hat{P}(\mathbf{x}|y) = \hat{P}_{\Theta_y}(\mathbf{x}|y)$ called likelihood model, where Θ_y describes the distribution's parameters to be determined (e.g., mean and covariance matrix if the likelihood model is a Gaussian).

If $\hat{P}(\mathbf{x}, y) = P(\mathbf{x}, y)$, this classifier achieves the optimal (Bayes) error [13]. However, with limited data, one has to

[2]An alternative architecture is discussed in [3], [7].

[3]We use P to denote both probability mass functions and densities.

carefully choose the model assumed for the likelihoods and the algorithm for their estimation.

Different likelihood models have been adopted for Bayes classifiers (see, e.g., [14]). Assuming individual diagonal covariance matrices $\mathbf{\Sigma}_{yg}$ for each Gaussian leads to the model adopted for both GMM and DGMM classifiers:

$$\hat{P}(\mathbf{x}|y) = \sum_{g=1}^{G_y} w_{yg} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{yg}, \mathbf{\Sigma}_{yg}). \quad (5)$$

The distinction between GMM and DGMM is their training algorithms.

Training a Bayes classifier consists in estimating the parameters Θ of all its likelihood functions $\hat{P}(\mathbf{x}|y)$. The conventional way of estimating Θ for all Bayes classifiers but DGMM is through maximum likelihood estimation (MLE). Assuming N iid training examples, MLE classifiers seek

$$\Theta^g = \arg\max_{\Theta} R^g(\Theta),$$

where

$$R^g(\Theta) = \prod_{n=1}^{N} \hat{P}(\mathbf{x}_n|y_n).$$

The Bayes classifiers trained with MLE are called *generative* [2] or *informative* [1]. The term generative is used because if the estimated $\hat{P}(\mathbf{x}, y)$ is "close" to the true distribution $P(\mathbf{x}, y)$, we could use $\hat{P}(\mathbf{x}, y)$ to generate samples with statistics similar to the ones of our original training set. However, for the sake of classification, we do not need to keep Θ. For example, one cannot generate samples out of a LDA classifier after simplifying the expressions [1] that define F. In such cases, the term informative seems more appropriate.

By contrast, discriminative Bayes classifiers (and other probabilistic classifiers, such as the relevance vector machine [10]) seek

$$\Theta^d = \arg\max_{\Theta} R^d(\Theta),$$

where

$$R^d(\Theta) = \prod_{n=1}^{N} \hat{P}(y_n|\mathbf{x}_n).$$

Note that

$$R^d(\Theta) = \prod_{n=1}^{N} \frac{\hat{P}(\mathbf{x}_n|y_n)\hat{P}(y_n)}{\hat{P}(\mathbf{x}_n)} \quad (6)$$

$$= \prod_{n=1}^{N} \left(1 + \frac{\sum_{j \neq y_n} \hat{P}(\mathbf{x}_n|j)\hat{P}(j)}{\hat{P}(\mathbf{x}_n|y_n)\hat{P}(y_n)}\right)^{-1}. \quad (7)$$

It follows that discriminative procedures try not only to maximize the likelihood of examples $(\mathbf{x}, y)$, but, at the same time, minimize the likelihood of competing classes $j \neq y$.

Conventionally, the *expectation-maximization* (EM) algorithm [15] is used for training GMMs through MLE. As for others generative-discriminative pairs of classifiers, training a discriminative Bayes classifier is harder than a generative. There are no closed-form solutions and iterative optimization algorithms are needed. In this work, DGMMs are trained with the algorithm proposed in [14], which is called here *fast extended EM* (FEEM) algorithm.

Roughly speaking, if the modeling assumptions are correct, adopting a generative classifier is more appropriate [16], [1], [2]. In fact, if training data is scarce, generative classifiers can achieve better performance than their discriminative counterparts [2]. On the other hand, there is empirical evidence showing that discriminative outperform generative classifiers if the likelihood model is not correct (see, e.g., [1]) or the estimated prior probabilities do not match the statistics of the test set [16].

C. Comparing the Classifiers

A SVM with a linear kernel can be converted to a perceptron, which avoids storing the support vectors and saves computations during the test stage. However, for speaker verification, the task posed to the classifier is very hard: to disambiguate a speaker from the others based only on a short segment (typically 20 to 40 milliseconds of speech) and one often needs to adopt a non-linear kernel. Besides, the space dimension is relatively low (typically Q=39) and sometimes the SVM training algorithm does not properly converge with the linear kernel. In this subsection, we assume Gaussian kernels. The Gaussian kernel allows for a direct comparison of SVM with GMM and DGMM, given that in all three cases the training procedure seeks a linear combination of Gaussians.

In speaker verification, the priors $\hat{P}(y)$ for GMM and DGMM are assumed *non-informative*, i.e., neglected in Eq. (4). Hence, the score of the t-th frame, provided by the model representing the speaker y, would be $f(\mathbf{x}_t) = \log \hat{P}(\mathbf{x}_t|y)$. However, it is well-know that it is beneficial to use a *universal background model* (UBM) [17], [3], and typically the score

$$f(\mathbf{x}_t) = \log \hat{P}(\mathbf{x}_t|y) - \log \hat{P}(\mathbf{x}_t|\text{ubm})$$

is the subtraction of the log-likelihoods obtained through two *convex* linear combinations (mixtures) of Gaussians, one for the target speaker and the other for the UBM. SVMs using a Gaussian kernel, as in Eq. (2), also output a score based on a linear combination of Gaussians

$$f(\mathbf{x}_t) = \sum_{n=1}^{N} \omega_n e^{-\gamma||\mathbf{x}_t - \mathbf{x}_n||^2} + b.$$

but the weights ω do not need to obey probabilistic constraints. On the other hand, the covariance matrix is restricted to be $\sigma^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix, which requires normalizing the front end parameters [3].

Concerning the computational cost for training the classifiers, GMM is the best option because the EM algorithm is fast and its memory requirement is very small. In the E-step, EM goes over the whole training set just collecting *sufficient statistics* for the M-step (see, e.g., [14]). DGMM training can also exploit sufficient statistics, but requires more computations. The FEEM algorithm incorporates some speedup techniques [14] and leads to a training time around 2 to 3 times longer than GMM. SVM requires a much longer training time, as it scales approximately with $\mathcal{O}(N^2)$.

Table I presents a summary of the most important features for the three classifiers. The next section presents experimental results achieved by them.

IV. Experimental Results

In this section we discuss experimental results comparing GMM, DGMM and SVM. We start by describing the IME corpus, adopted for the simulations.

A. IME Corpus

The IME corpus is composed by 468 files[4], corresponding to 21.9 hours of recorded signal. For the sake of comparison, the popular NIST-2001 corpus (http://www.ldc.upenn.edu) is composed by 2350 (shorter) files, which correspond to 26.4 hours of speech. The utterances in the IME corpus were collected from cellular and wired phone calls made by 75 speakers. The amount of files in each group is: 111 - cellular / test, 118 - cellular / train, 120 - wired / test and 123 - wired / train.

In order to better organize the simulations, we converted the original 11-digit file names (e.g., 12151110051.wav) into names such as id001.cel.train.man.RJ.cn.42.wav, where a dot separates the information fields. These fields represent a unique speaker ID, cellular or wired phone, train or test, gender, speaker geographical origin, recording conditions, speaker's age and file extension.

The D41ESC Dialogic board was used to collect all utterances. According to its documentation, this board supports 8-bit PCM μ and A-laws. However, the speech files are stored in the Microsoft RIFF format as 8-bit PCM linear[5]. One would expect 12 or more bits per sample when expanding from the logarithmic to a linear scale [18]. Besides this problem, silence represents a relatively high percentage of the total amount of data. Figure 1 compares the histograms of speech samples from all utterances in the NIST 2001 and IME corpora. One can see that silence is much more frequent in the IME than in the NIST corpus.

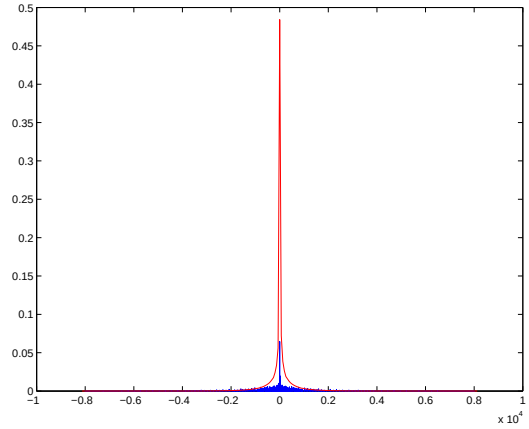
Hence, we tried to eliminate silence from the utterances using a simple voice activity detector (VAD) that is based on the signal energy. The VAD routine generates a label file, indicating where silence occurs. Then, to avoid problems when calculating derivatives of the parameters, we run the front end using the whole utterance, and cut off the frames corresponding to silence based on the VAD label file.

B. Performance Results

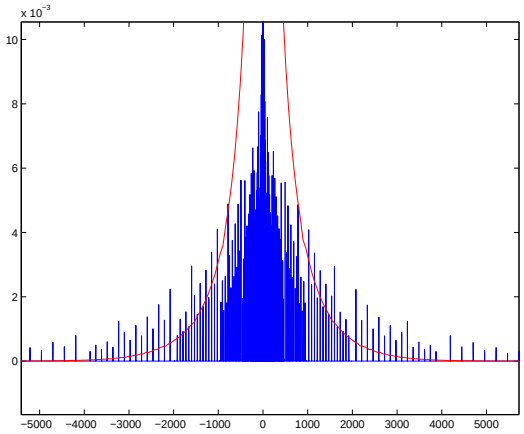
In [5] we presented results using the IME corpus for several front ends. Here we adopt the same experimental setup, but use exclusively 12 perceptual linear prediction (PLP) parameters, plus the energy and two first derivatives (the so-called PLPEDA39). We restrict the simulations even more by using only the utterances for the "wired" phone calls (discarding the "cellular" utterances). Even this restricted scenario is enough for stressing the pitfalls of applying discriminative learning to speaker verification.

[4]In fact, the IME corpus originally has 472 files, but 4 are corrupted.

[5]The 20-th byte of a Microsoft RIFF file (WAV) indicates the kind of PCM: 6 means A-law, 7 is μ-law and 1 is linear PCM. The IME corpus uses 1.



(a) Normalized histogram.



(b) Zoom.

Fig. 1. NIST (discrete representation using vertical lines) and IME (continuous curve) normalized histograms of speech samples. NIST has a peak of 0.1 around zero, while IME almost reaches 0.5.

The first point to consider is that the training algorithm tries to find the best classifier F, while the overall goal is to find the best system G. The two are obviously related, as indicated in Figures 2 and 3, which show the error rate per frame and the *equal error rate* (EER) [5], respectively. In these figures, the abscissa is the number of frames in the training set. The results were obtained adopting 20 Gaussians for both GMM and DGMM, based on the conclusions in [5]. One can see that, as discussed in [2], generative can outperform discriminative classifiers when the training data is scarce. Our results indicate that this behavior also happens for SVM.

As mentioned, the task of learning F is very hard: disambiguate a speaker from the others based only on a short segment. Besides, the space dimension is relatively low, i.e., there are relatively few parameters and a strong overlap of the classes in the input space $\mathcal{X}^L$. These two facts impact specially the SVM classifier, which performed poorly with an average EER of 3% when the training set had 1500 frames, which is higher than the GMM and DGMM as shown in Figure 3. The next subsection discusses some issues related to this situation.

TABLE I
COMPARISON OF GMM, DGMM AND SVM.

	GMM	DGMM	SVM
Dependency on N (training examples)	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$
Support multiclass problems	yes	yes	no
Optimization criterion	$R^g(\Theta)$	$R^d(\Theta)$	Eq. (1)
Low memory footprint through sufficient statistics	yes	yes	no
Is the number G of Gaussians pre-specified?	yes	yes	no
Gaussian means restricted to be training instances?	no	no	yes
Same (pre-specified) variance for all Gaussians ?	no	no	yes

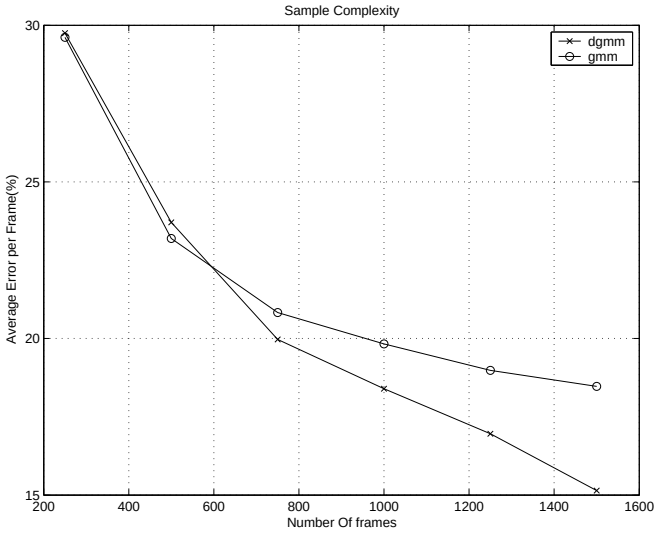


Fig. 2. The error (%) per frame for GMM and DGMM.

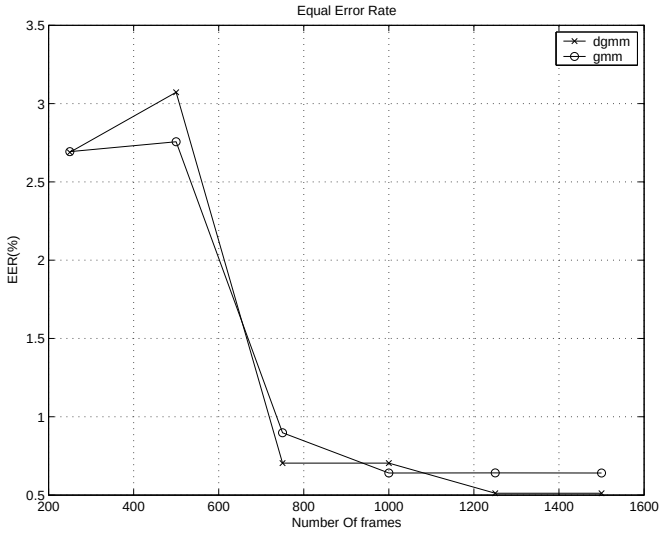


Fig. 3. Equal error rate (%) for GMM and DGMM.

C. The Importance of Model Selection for SVM

A classical way of performing model selection is through cross-validation (CV), typically with 10 folds. The folds are disjoints, that is, each vector $\mathbf{x}$ belongs to only one fold. In many situations, the error using such validation sets provide a good indication of generalization capability [8]. Unfortunately, this is not true for a typical CV applied speech processing scenarios. For example, training a verification system with CV over the frames x_t of the training set, would lead to overly optimistic error rates for the validation set, because the impostors in this set are the same used in the training. This can be circumvented by a careful partition of the training, test and validation sets, such that impostors (negative examples) in the validation do not coincide with the ones in the training set.

Another problem is that, typically, the training set should use only frames x_t from an unique utterance or conversation (for example, recorded over a single phone call). On the other hand, for testing, one has to use frames obtained in a different recording situation (e.g., channel mismatch). Ideally, the validation set, for performing model selection when training a classifier, should have frames from the target speaker (positive examples) with conditions (mismatch, etc.) similar to the one found during test. When that is not the case, GMM and DGMM present a higher degree of robustness, while SVM often fails, overfitting the training data and leading to relatively high error rates in the test set (this can be interpreted under the light of the structural risk theory [10].

In order to study this situation, we conducted an experiment where the validation set was made the same as the test set. Note that this is not the same as testing with the training set. The validation was simply used to choose the number of Gaussians (for GMM and DGMM), C and γ for SVM. The results showed that SVM was able to outperform both GMM and DGMM. It should be noticed that this test was conducted only for the sake of interpreting the results, but neither test or validation data can be used during training when one is trying to estimate the generalization capability.

D. The Importance of Input Space Dimension for SVM

It is well-know that SVM achieves good performance when the input space has a high dimension (L is large). Therefore, a simple experiment was conducted, in which the number L of features was increased from 39 to 87, by concatenating a stream of 48 mel-frequency cepstrum coefficients (MFCC) [6]. The results are shown in Figure 4, which indicates the EER when varying the number of frames that compose the training sequence. It can be seen that SVM catches up GMM when

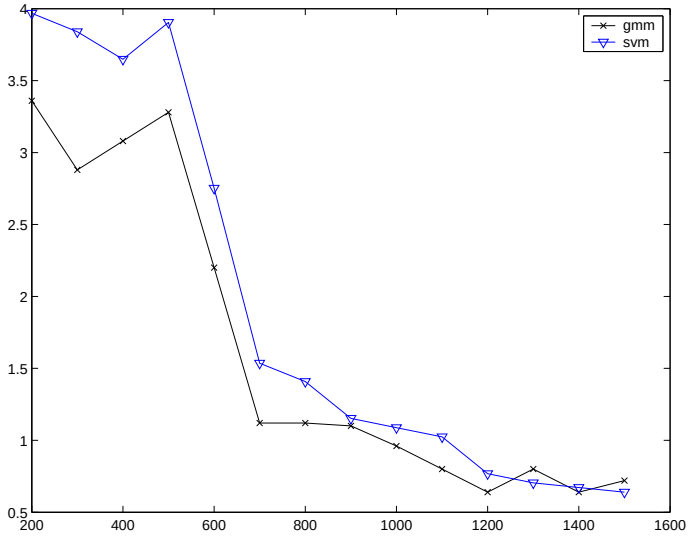


Fig. 4. The EER (%) for GMM and SVM when using 87 features (39 PLPs and 48 MFCCs).

1300 frames are used, and ends up with a slightly better result.

In spite of its simplicity, the experiment reinforces the idea that SVM and other discriminative classifiers need larger training sets when compared to generative ones. Besides, Figure 4 shows that SVM benefits more than GMM of a larger input space dimension in this particular experiment. Additional simulations should be done in order to draw more general conclusions. It should be noticed that MFCC and PLP are relatively similar front ends [14], so there is no much complementary information between them.

V. Conclusions

In this work the adoption of DGMM in speaker verification is discussed. Simulation results using the IME corpus showed that DGMM e SVM can improve the performance when compared to GMM. The preliminary results showed that DGMM outperformed GMM, while SVM could be made to reach the same results as GMM when the number of parameters was increased from 39 to 87. However, the main goal was not to achieve improvements in accuracy, but get insight about the pitfalls of applying discriminative learning to speaker verification. This is done by comparing GMM and its discriminative counterpart, the DGMM, which is similar to SVM and other kernel methods in many aspects, especially when they use the Gaussian kernel. Besides, simulations with SVM revealed some aspects of its adoption in speaker recognition.

Among many factors, such as the training set size and input space dimension, the one that impacts discriminative learning the most, is the model selection stage. A proper model selection is essential, for example, to make SVM competitive in speaker verification. Generative classifiers are more robust to overfitting and require less care when choosing the validation set. Future research points towards comparing GMM, DGMM and SVM using the whole IME corpus, increasing the number of front end parameters, mixing utterances from cellular and wired phone calls, and testing different ways of performing model selection.

References

[1] Y. Rubinstein and T. Hastie. Discriminative vs informative learning. In *Knowledge Discovery and Data Mining*, pages 49–53, 1997.

[2] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NIPS*, 2002.

[3] V. Wan and S. Renals. Speaker verification using sequence discriminant support vector machines. *IEEE Transactions on Speech and Audio Processing*, 13(2):203–10, March 2005.

[4] A. Klautau, N. Jevtić, and A. Orlitsky. Discriminative Gaussian mixture models: A comparison with kernel classifiers. In *ICML*, pages 353–360, 2003.

[5] T. Imbiriba, A. Klautau, N. Parihar, S. Raghavan, and J. Picone. GMM and kernel-based speaker recognition with the ISIP toolkit. In *Proceedings of the 2004 IEEE International Workshop on Machine Learning for Signal Processing*, pages 371–380, September 2004.

[6] X. Huang, A. Acero, and H.-W. Hon. *Spoken language processing*. Prentice-Hall, 2001.

[7] N. Smith and M. Gales. Speech recognition using SVMs. In MIT Press, editor, *Advances in Neural Information Processing Systems 14*, 2002.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer Verlag, 2001.

[9] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[10] B. Scholkopf and A. Smola. *Learning with kernels*. MIT Press, 2002.

[11] A. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. Winston, 1977.

[12] G. Kimeldorf and G. Wahba. Some results on Tchebychean spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.

[13] R. Duda, P. Hart, and D. Stork. *Pattern classification*. Wiley, 2001.

[14] A. Klautau. *Speech Recognition Using Discriminative Classifiers*. PhD thesis, UCSD, 2003.

[15] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society (B)*, 39:pp. 1–22, 1977.

[16] A. Nádas, D. Nahamoo, and M. Picheny. On a model-robust training method for speech recognition. *IEEE Trans. on ASSP*, 36:1432–6, 1988.

[17] D. Reynolds. Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, 17:91–108, August 1995.

[18] L. Rabiner and R. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, 1978.