FUSING GENERATVE AND DISCRIMINATIVE UBM-BASED SYSTEMS FOR SPEAKER VERIFICATION

Nicolas Scheffer, Jean-François Bonastre

LIA, Université d'Avignon
Agroparc, BP 1228
84911 Avignon CEDEX 9, France
{nicolas.scheffer,jean-francois.bonastre}@univ-avignon.fr

ABSTRACT

In the past few years, discriminative approaches to perform speaker detection have shown good results and an increasing interest. Among these methods, SVM based systems have lots of advantages, especially their ability to deal with a high dimension feature space. Generative systems such as UBM-GMM systems show the greatest performance among other systems in speaker verification tasks. Combination of generative and discriminative approaches is not a new idea and has been studied several times by mapping a whole speech utterance onto a fixed length vector.
This paper presents a straight-forward, cost friendly method to combine the two approaches with the use of a UBM model only to drive the experiment. We show that the use of the TFLLR kernel, while closely related to a reduced form of the *Fisher mapping*, implies a performance that is close to a standard GMM/UBM based speaker detection system. Moreover, we show that a combination of both outperforms the systems taken independently.

1. INTRODUCTION

Current state-of-the-art speaker detection systems are based on generative speaker models such as Gaussian Mixture Models (GMM). Using a UBM/GMM based system [1] is now compulsory to obtain good performance in evaluation campaigns such as the NIST-SRE evaluation. Lately, discriminative approaches to perform speaker detection had been successfully applied [2] and interests in these methods do not seem to stop increasing. Among these methods, Support Vector Machines have lots of advantages, especially their ability to treat a high dimension feature space, the realization of *Vapnik Structural Risk Minimization* principle, their easy integration as it is generally faster than a UBM-GMM system and also due to the multiple opensource tools available to the research community.
Combination of generative and discriminative classifiers is not a new idea and has been studied in [3], [4] and [5]. by mapping a variable length input data (such as speech data) onto a fixed length vector. However, these methods need the training of a GMM for each speaker, thus increasing the complexity of the problem. Moreover, the derivation of this mapping is complex to produce and the choice of the proper space needs to be done. This paper presents a straight-forward, cost-friendly method to map the test utterances into a fixed-length vector with the use of a UBM model only to drive the experiment. The derivation of the TFLLR (Term Frequency Log Likelihood Ratio) kernel proposed by [6] is slightly modified to take the Gaussian component weight of the UBM into account. Indeed, the useful information to discriminate speakers in this work is the associated statistics of the UBM Gaussian component and the speaker training data.
In section 2, a baseline GMM/UBM speaker verification system will be presented, as well as the description of a Universal Background Model. Next, the SVM classifier, the brief description of sequence discrimination techniques, as well as the kernel derived for our task will be presented in section 3. Section 5 presents the experimental protocol as well as results of this UBM-SVM on a part of the NIST-SRE-2005 database [7]. We also address the issue of feature selection at the input of the SVM in this section. We finally conclude with a combination of a state-of-the-art UBM-GMM system and the approach presented in this paper.

2. GMM BASED SPEAKER DETECTION SYSTEM

This section describes the UBM-GMM approach, as well as the LIA_SpkDet UBM-GMM system. Performance of the system is presented in section 6, on NIST-SRE05 protocol, in order to merge scores of both systems.

2.1. UBM/GMM approach

GMM-UBM is the predominant approach used in speaker recognition systems, particularly for text-independent task [8]. Given a segment of speech Y and a speaker S, the speaker verification task consists in determining if Y was spoken

by S or not. This task is often stated as basic hypothesis test between two hypotheses: Y is from the hypothesized speaker S ($H0$), and Y is not from the hypothesized speaker S ($H1$). A likelihood ratio (LR) between these two hypotheses is estimated and compared to a decision threshold θ. The LR test is given by:

$$LR(Y, H0, H1) = \frac{p(Y|H0)}{p(Y|H1)} \quad (1)$$

where Y is the observed speech segment, $p(Y|H0)$ is the likelihood function for the hypothesis $H0$ evaluated for Y, $p(Y|H1)$ is the likelihood function for $H1$ and θ is the decision threshold for accepting or rejecting $H0$. If $LR(Y, H0, H1) > \theta$, $H0$ is accepted else $H1$ is accepted.
A model denoted λ_{hyp} represents $H0$, it is learned using an extract of speaker S voice. The model $\lambda_{\overline{hyp}}$ represents the alternative hypothesis, $H1$, and is usually learned using data gathered from a large set of speakers.
The likelihood ratio statistic becomes $\frac{p(Y|\lambda_{hyp})}{p(Y|\lambda_{\overline{hyp}})}$. Often, the logarithm of this statistic is used giving the $logLR$ (LLR):

$$LLR(Y) = log(p(Y|\lambda_{hyp})) - log(p(Y|\lambda_{\overline{hyp}})). \quad (2)$$

In the presented approach, the models are Gaussian Mixture Models which estimate a probability density function by:

$$p(x|\lambda) = \Sigma_{i=1}^{M} w_i N(x|\mu_i, \Sigma_i) \quad (3)$$

where w_i, μ_i and Σ_i are weights, means and covariances associated with the Gaussian components in the mixture. Usually a large number of components in the mixture and diagonal covariance matrices are used.

Universal Background Model

The UBM has been introduced and successfully applied by [1] to speaker verification. It aims at representing the inverse hypothesis in the Bayesian test, i.e. it is designed to compute the data probability not to belong to the targeted speaker, ie $\lambda_{\overline{hyp}}$. A UBM is learned with multiple audio files from different speakers, usually several hundreds. For speaker verification, some approaches consist in having specific UBM models, such as a UBM model per gender or per channel.
The UBM is trained with the EM algorithm on its training data. For the speaker verification process, it fulfills two main roles:

- It is the *apriori* model for all target speakers when applying Bayesian adaptation to derive speaker models.
- It helps to compute log likelihood ratio much faster by selecting the best Gaussian for each frame on which likelihood is relevant.

This work proposes to use the UBM as a guide to discriminative training of speakers.

2.2. The LIA_SpkDet system

The background model used for the experiments is the same as the background model used by the LIA for the NIST SRE 2005 campaign (male only). The training is performed based on NIST SRE 1999 and 2002 databases, and consists in 1.3 millions of speech frames (3,5 hours). Training was performed using the ALIZE and LIA_SpkDet toolkits[1] [9]. Speaker models are derived by Bayesian adaptation on the Gaussian component means, with a relevance factor of 14. Frames are composed of 16 LFCC parameters and its derivatives. A normalization process is applied, so that the distribution of each cepstral coefficient is 0-mean and 1-variance. The background model has 2048 components and no component variance is above 0.5.

3. DISCRIMINATIVE TRAINING USING UBM/GMM

This section presents the methodology adopted in order to build a whole speaker detection system. The UBM-GMM presented in section 2 is the foundation of the system. The first part briefly describes the SVM classifier that will be used for the task. Next, techniques consisting in using the GMM parameters in a SVM (called *mapping*) are presented. Finally, the TFLLR kernel is introduced and derived to suit our problem.

3.1. Support Vector Machine classification

Support Vector Machines are described by Vapnik [10] and are usually used as a binary classifier in speaker verification (target/non-target). To answer a linearly separable problem, the SVM gives the optimal hyperplane that maximizes the margin between the two classes, among the several possible hyperplane.
Let the separating hyperplane be defined by $x\dot{w}+b = 0$ where w is the normal to the hyperplane. For linearly separable data labelled $\{x_i, y_i\}$, $x_i \in \Re_d^N, y_i \in \{-1, 1\}, i = 1 \dots N$, the optimum boundary chosen according to the maximum margin criterion is found by minimizing the objective function:

$$E = \|w\|_2^2$$
$$\text{with} \quad (x_i w + b)y_i \geq 1 \, , \forall i$$

The solution for the optimal boundary, w_0, is a linear combination of a subset of the training data, x_s called the *support vectors*. Data may be classified depending on the sign of $xw_0 + b$.

In speaker detection, this means that a speaker is modelled by its training data and by an optimum subset of impostor data, the closest impostors. This reduces considerably the problem size and is one of the reason of the SVM ability of dealing

[1] http://www.lia.univ-avignon.fr/heberges/ALIZE/

with large size of input training feature set.
Generally, data is not linearly separable, and the introduction of *slack variables* is necessary.

3.2. Discriminative sequence classification

Discriminative classification of sequences with different length, such as speech data, is a very difficult task. However, techniques aiming at mapping a complete utterance to a fixed length vector exist and can achieve speaker detection tasks. Such methods have been applied in [2], with polynomial kernels (and the GLDS kernel) showing good performance at the NIST SRE evaluations.
Such mapping were first developed by Jaakola and Haussler [11] and is known as *Fisher Kernel*, then generalized by Smith and Gales [12] as a technique referred to as *score-spaces*. The concept of mapping may be interpreted as an SVM kernel (such as the Fisher kernel being a dot product between Fisher mapping).
Interest will be given to the likelihood score space in this paper. The reader is invited to look at [3] for a detailled derivation of other spaces.
Let us consider a GMM model M parameterized by θ, the *Fisher mapping* of a sequence X of T frames, is known as being the first derivative of the score function, precisely:

$$\Psi_{Fisher}(X) = \nabla_\theta log(\ell(X|M,\theta)) \tag{4}$$

The resulting vector will contain all the derivatives with respect to each parameter in θ.
The derivation of this mapping with respect to a GMM component G_j, with weight α_j is given below:

$$\frac{\partial}{\partial \alpha_j} log(\ell(X|M,\theta)) = \sum_{t=1}^{T} \frac{\ell(x_t|G_j)}{\sum_{i=1}^{N_g} \alpha_i \ell(x_t|G_i)} \tag{5}$$

Thus, an input vector in the SVM could contain this partial mapping without the derivatives of means and variances. The dimension of this vector is equal to the number of component of the initial GMM.

3.3. Using UBM and SVM for speaker verification

The approach presented in this paper basically relies on the information given by a single GMM. Instead of learning client models by MAP adaptation (or MLE criterion) and then perform discrimination with a SVM, a method using only the UBM component weights to drive the discriminative learning is proposed.

3.4. Applying TFLLR kernel to GMM weights

The features used at the input of the SVM - which are extracted from the UBM parameters - shall represent the behavior of this model on speaker training data. The TFLLR kernel method presented in [6] is used to produce feature vectors for Ngram type approaches. Its formulation is used to derive a proper kernel to suit our problem.
Let us consider tokens k belonging to a bag-of-Ngram. Let the token k likelihood on a data sequence X be defined as $p(k|X)$, the TFLLR kernel is computed as follows:

$$\sum_k \frac{p(k|X_1)}{\sqrt{p(k|X_W)}} \frac{p(k|X_2)}{\sqrt{p(k|X_W)}} - 1 \tag{6}$$

where X_1,X_2,X_W are the respective training data of two speakers and the background model. The kernel construction finally resides in the weighting of speaker likelihoods by the likelihood of the background model.
Let now assume the token is a UBM Gaussian component (defined as W_k for the k^{th} componenent), and consider its probability as its associated data occupation. It ends up that for a specific sequence X, the following quantity is produced:

$$\frac{p(W_k|X)}{\sqrt{p(W_k)}} = \sum_{i=1}^{T} \sqrt{p(W_k)}^{-1} \frac{\ell(x_t|W_k)p(W_k)}{\sum_l \ell(x_t|W_l)p(W_l)} \tag{7}$$

$$= \sqrt{p(W_k)} \sum_{i=1}^{T} \frac{\ell(x_t|W_k)}{\sum_l \ell(x_t|W_l)p(W_l)} \tag{8}$$

$$= \sqrt{p(W_k)} \nabla_{\alpha_k} log(\ell(X|W,\theta_W)) \tag{9}$$

This kernel is closely related to the *Fisher mapping* component described in (5). The additional square root of the Gaussian component weight can be seen as a normalization in order to smooth the dynamic of the features.
The estimation of $p(W_k|X)$ is the hidden variable computed during the EM algorithm.

4. PROTOCOL

4.1. Database

Speaker verification experiments, presented in section 5, are performed based upon the NIST 2005 database, common condition, male speakers only. This condition consists of 274 speakers. Train and test utterances contain 2.5 minutes of speech in average (telephone conversation).
The whole speaker detection experiment consists in 13624 tests (951 target tests). Each test is made independently and the use of information from other tests to take a decision on the current test is forbidden.

4.2. Using the SVM in a speaker detection experiment

In order to build impostor models (i.e. negative labelled data), speakers coming from the background model are used, here 161 speakers. During the training, the input of the classifier is the concatenation of all impostor vectors and the speaker vector issued from its training data. During the verification

process, the test vector is given as an input to SVM models. The maximum margin decision is found by processing this input through a linear kernel.
We used the SVM-Light toolkit by Thorsten Joachims [13] to induce SVMs and classify instances. To compensate for the severe imbalance between the target and background data, we adopted a cost model to weight the positive examples 200-fold with respect to the negative examples. The scores obtained in this manner were then normalized using TNORM (except when explicitly mentioned).

5. EXPERIMENTS

For the experiments, two different sizes of UBM models were used, 128 and 2048 Gaussian component.

Model size effect on performance

Figure 1 show the difference in performance between the two models. The results clearly shows that a 2048 model size outperform a 128. Indeed, an absolute gain of 5% is observed. As in a standard UBM-GMM speaker verification system, the number of dimensions (Gaussians) is critical and performance improves as this number increases (at least until 2048).

Effect of score normalization on performance

Figure 2 shows the effect of score normalization known as T-Normalization. Impostor speakers are the same as the negative labelled examples, i.e. speakers that composed the background model. For some SVM based methods, the TNormalization technique is done implicitly and does not bring any effect. In our case, it still has two main advantages:

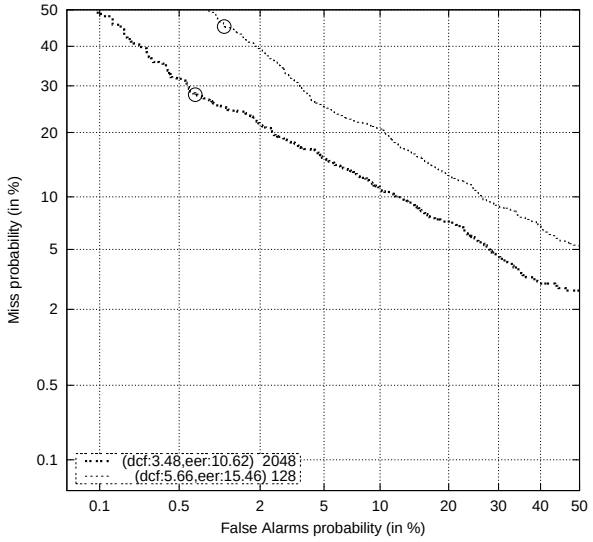

Fig. 1. *Comparing UBM/SVM systems with different UBM size: 128 (dotted line) and 2048 (dashed line)*

- It brings a significant gain, particularly at the DCF,
- It scales scores to the same space as the UBM-GMM, thus making the fusion process easier.

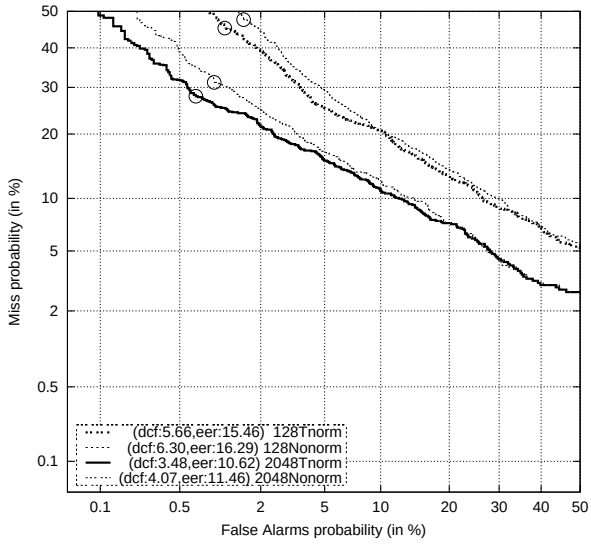

Fig. 2. *Effect of Tnorm score normalization on two different model sizes: below 2048, up 128. Tnorm DET curves (thick), NoNorm DET curves (thin)*

6. COMBINATION WITH A UBM-GMM SYSTEM

The baseline system presented in section 2 is the UBM-GMM submission of the LIA in the NIST SRE 2005 campaign. Surprisingly, the latter is close to the UBM-SVM system in terms of performance. Indeed, this UBM-SVM system performs as well as the GMM at the decision cost point. A significant gain is then expected when fusing the two systems. To conclude this work, two fusions are presented, both are an arithmetic mean of the scores of both systems. The first one is an equally weighted fusion, the second one is a fusion with weights of 0.3 for UBM-SVM and 0.7 for UBM-GMM (these parameters were found empirically where one set optimize the min DCF while the other optimize the overall performance). Table 1 and figure 3 show results for the equally weighted fusion.
A simple fusion shows a significant gain brought by the combination of the two classifiers. Depending on the fusion weights, gain can be observed at different operating points. Equally weighted fusion improves the DCF by a relative gain of around 12%, the other improves both the DCF and EER by 9% and 6% relative respectively.
This states that complementary information has been found by adding discriminative information to the UBM-GMM system.

Table 1. *Arithmetic mean fusion between UBM-SVM and UBM-GMM: 1) Equally weighted 2) 0.3/0.7*

System	DCF	EER
1:UBM-GMM	3.49	8.73
2:UBM-SVM	3.48	10.62
Fusion 1:50% / 2:50%	3.06	8.41
Fusion 2:70% / 2:30%	3.17	8.20

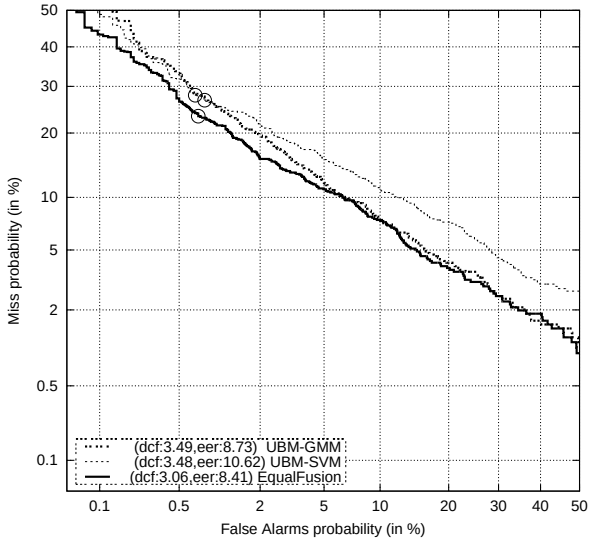

Fig. 3. *UBM-GMM system, SVM-GMM system as well as the equally weighted fusion*

7. CONCLUSION

While the issue of mapping speech data utterances on a fixed dimension vector has been addressed several times, the work presented here proposes an easy and very performant scheme to take benefit from both generative and discriminative systems by extending the use of the TFLLR kernel.
Indeed, it has been shown in section 3 that the input feature vectors computed by the TFLLR kernel are closely related to the Fisher mapping when only weights are derived. It is very easy to compute and costless if one has already a GMM/UBM system in its range of speaker detection system.
One originality of this approach is to demonstrate that the UBM only can be used to perform the verification task. While other methods have to build and use GMMs (with an MLE or MAP criterion), we claim that an UBM can produce sufficient information for the task. Finally, we showed that this system combined with a pure UBM-GMM based system can bring a relative gain of around 12% at the DCF by capturing other information.
This work will be continued with an effort on finding a analytical criterion for feature selection.

8. REFERENCES

[1] D. A. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," in *Speech Communication*, 1995, vol. 171-2, pp. 91–108.

[2] V. Wan and W. M. Campbell, "Support vector machines for speaker verification and identification," in *Neural Network for Signal Processing*, 2000, pp. 775–784.

[3] V. Wan and S. Renals, "Speaker verification using sequence discriminant support vector machines," in *Transactions in Speech and Audio Processing*, 2002.

[4] L. Quan and S. Bengio, "Hybrid generative and discriminative models for speech and speaker recognition," in *Tech. Rep. IDIAP-RR 02-06, IDIAP*, March 2002.

[5] S. Fine, J. Navratil, and R. A. Gopinath, "A hybrid gmm/svm approach to speaker identification," in *ICASSP*, 2001, pp. 417–420.

[6] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R Leek, "High-level speaker verification with support vector machines," in *ICASSP Conference, Montreal, CANADA*, May 2004, pp. 73–76.

[7] NIST, "The NIST year 2005 speaker recognition evaluation plan," `http://www.nist.gov/speech/tests/spk/2005/sre-05_evalplan-v5.pdf`, April 2005.

[8] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing, Special issue on biometric signal processing*, 2004.

[9] J-F. Bonastre, F. Wils, and S. Meigner, "ALIZE, a free toolkit for speaker recognition," in *ICASSP Conference, Philadelphia, USA*, March 2005.

[10] V. N. Vapnik, *Statistical Learning Theory*, 1998.

[11] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," 1998.

[12] N. Smith, M. Gales, and M. Niranjan, "Data-dependent kernels in svm classification of speech patterns," in *Tech. Rep. CUED/F-INFENG/TR.387, Cambridge University Engineering Dept.*, 2001.

[13] T. Joachims, "Making large-scale svm learning," in *Practical. Advances in Kernel Methods - Support Vector Learning,B. Schokopf and C. Burges and A. Smola, MIT Press*, 1999.