Discriminative speaker recognition using Large Margin GMM

Reda Jourani, Khalid Daoudi, Régine André-Obrecht, Driss Aboutajdine

Neural Comput & Applic manuscript No.
(will be inserted by the editor)

Discriminative speaker recognition using Large Margin GMM

Reda Jourani · Khalid Daoudi · Régine André-Obrecht · Driss Aboutajdine

Received: date / Accepted: date

Abstract Most state-of-the-art speaker recognition systems are based on discriminative learning approaches. On the other hand, generative Gaussian mixture models (GMM) have been widely used in speaker recognition during the last decades. In an earlier work, we proposed an algorithm for discriminative training of GMM with diagonal covariances under a large margin criterion. In this paper, we propose an improvement of this algorithm which has the major advantage of being computationally highly efficient, thus well suited to handle large scale databases. We also develop a new strategy to detect and handle the outliers that occur in the training data. To evaluate the performances of our new algorithm, we carry out full NIST speaker identification and verification tasks using NIST-SRE'2006 data, in a Symmetrical Factor Analysis compensation scheme. The results show that our system significantly outperforms the traditional discriminative Support Vector Machines (SVM) based system of SVM-GMM supervectors, in the two speaker recognition tasks.

Keywords Large margin training · Gaussian mixture models · discriminative learning · speaker recognition · session variability modeling

R. Jourani · R. André-Obrecht
SAMoVA Group, IRIT - UMR 5505 du CNRS
University Paul Sabatier, 118 Route de Narbonne, Toulouse, France
E-mail: {jourani, obrecht}@irit.fr

K. Daoudi
GeoStat Group, INRIA Bordeaux-Sud Ouest
351, cours de la libération, Talence. France
E-mail: khalid.daoudi@inria.fr

R. Jourani · D. Aboutajdine
Laboratoire LRIT. Faculty of Sciences, Mohammed 5 Agdal University
4 Av. Ibn Battouta B.P. 1014 RP, Rabat, Morocco
E-mail: aboutaj@fsr.ac.ma

1 Introduction

Generative (or informative) training of Gaussian Mixture Models (GMM) using maximum likelihood estimation and maximum a posteriori estimation (MAP) [1] has been the paradigm of speaker recognition for many decades. Generative training does not however directly address the classification problem because it uses the intermediate step of modeling system variables, and because classes are modeled separately. For this reason, discriminative training approaches have been an interesting and valuable alternative since they focus on adjusting boundaries between classes [2,3], and lead generally to better performances than generative methods. Hybrid learning approaches have also gained a big interest. For instance, Support Vector Machines (SVM) combined with GMM supervectors are among state-of-the-art approaches in speaker verification [4,5].

In speaker recognition applications, mismatch between the training and testing conditions can decrease considerably the performances. The session variability remains the most challenging problem to solve. The Factor Analysis techniques [6,7], e.g., Symmetrical Factor Analysis (SFA) [8,9], were proposed to address that problem in GMM based systems. While the Nuisance Attribute Projection (NAP) [10] compensation technique is designed for SVM based systems.

Recently a new discriminative approach for multiway classification has been proposed, the Large Margin Gaussian mixture models (LM-GMM) [11]. The latter have the same advantage as SVM in term of the convexity of the optimization problem to solve. However they differ from SVM because they draw nonlinear class boundaries directly in the input space, and thus no kernel trick/matrix is required. While LM-GMM have been used in speech recognition, they have not been used in speaker recognition (to the best of our knowledge). In an earlier work [12], we proposed a simplified version of LM-GMM which exploit the fact that traditional GMM systems use diagonal covariances and only the mean vectors are MAP adapted. We then applied this simplified version to a "small" speaker identification task. While the resulting training algorithm is more efficient than the original one, we found however that it is still not efficient enough to process large databases such as in NIST Speaker Recognition Evaluation (NIST-SRE) campaigns [13].

In order to address this problem, we propose in this paper a new approach for fast training of Large-Margin GMM which allow efficient processing in large scale applications. To do so, we exploit the fact that in general not all the components of the GMM are involved in the decision process, but only the k-best scoring components. We also exploit the property of correspondence between the MAP adapted GMM mixtures and the world model mixtures. Moreover, we develop a new strategy to detect outliers and reduce their negative effect in training. This strategy leads to a further improvement in performances.

In order to show the effectiveness of the new algorithm, we carry out full NIST speaker identification and verification tasks using NIST-SRE'2006 (core condition) data. We evaluate our fast algorithm in a Symmetrical Factor Anal-

ysis compensation scheme, and we compare it with the NAP compensated GMM supervector Linear Kernel system (GSL-NAP) [5]. The results show that our Large Margin compensated GMM outperform the state-of-the-art discriminative approach GSL-NAP, in the two speaker recognition tasks.

The paper is organized as follows. After an overview on Large-Margin GMM training with diagonal covariances in section 2, we describe our new fast training algorithm in section 3. To make the paper self-contained, the GSL-NAP system and SFA are described in sections 4 and 5, respectively. Experimental results are reported in section 6.

2 Overview on Large Margin GMM with diagonal covariances (LM-dGMM)

In this section we start by recalling the original Large Margin GMM training algorithm developed in [11,14]. We then recall the simplified version of this algorithm that we introduced in [12].

In Large Margin GMM [11,14], each class c is modeled by a mixture of ellipsoids in the D-dimensional input space. The m^{th} ellipsoid of the class c is parameterized by a centroid vector μ_{cm} (mean vector), a positive semidefinite (orientation) matrix Ψ_{cm} and a nonnegative scalar offset $\theta_{cm} \geq 0$. These parameters are then collected into a single enlarged matrix Φ_{cm}:

$$\Phi_{cm} = \begin{pmatrix} \Psi_{cm} & -\Psi_{cm}\mu_{cm} \\ -\mu_{cm}^T\Psi_{cm} & \mu_{cm}^T\Psi_{cm}\mu_{cm} + \theta_{cm} \end{pmatrix}. \quad (1)$$

A GMM is first fit to each class using maximum likelihood estimation. Let $\{o_{n,t}\}_{t=1}^{T_n}$ ($o_{n,t} \in \mathcal{R}^D$) be the T_n feature vectors of the n^{th} segment (i.e. n^{th} speaker training data). Then, for each $o_{n,t}$ belonging to the class y_n, $y_n \in \{1, 2, ..., C\}$ where C is the total number of classes, we determine the index $m_{n,t}$ of the Gaussian component of the GMM modeling the class y_n which has the highest posterior probability. This index is called *proxy label*.

The training algorithm aims to find matrices Φ_{cm} such that "all" examples are correctly classified by at least one margin unit, leading to the LM-GMM criterion:

$$\forall c \neq y_n,\ \forall m, \qquad z_{n,t}^T\Phi_{cm}z_{n,t}\ \geq 1 + z_{n,t}^T\Phi_{y_n m_{n,t}}z_{n,t}, \quad (2)$$

where $z_{n,t} = \begin{bmatrix} o_{n,t} \\ 1 \end{bmatrix}$. Eq. (2) states that for each competing class $c \neq y_n$ the match (in term of Mahalanobis distance) of any centroid in class c is worse than the target centroid by a margin of at least one unit.

In speaker recognition, most of state-of-the art systems use diagonal covariances GMM. In these GMM based speaker recognition systems, a speaker-independent *world model* or *Universal Background Model* (UBM) is first trained with the EM algorithm [15] from tens or hundreds of hours of speech data gathered from a large number of speakers. The background model represents

speaker-independent distribution of the feature vectors. When enrolling a new speaker to the system, the parameters of the UBM are adapted to the feature distribution of the new speaker. It is possible to adapt all the parameters, or only some of them from the background model. Traditionally, in the GMM-UBM approach, the target speaker GMM is derived from the UBM model by updating only the mean parameters using a *maximum a posteriori* (MAP) algorithm [1], while the (diagonal) covariances and the weights remain unchanged.

Making use of this assumption of diagonal covariances, we proposed in [12] a simplified algorithm to learn GMM with a large margin criterion. This algorithm has the advantage of being more efficient than the original LM-GMM one [11,14] while it still yielded similar or better performances on a speaker identification task. In our Large Margin diagonal GMM (LM-dGMM) [12], each class (speaker) c is initially modeled by a GMM with M diagonal mixtures (trained by MAP adaptation of the UBM in the setting of speaker recognition). For each class c, the m^{th} Gaussian is parameterized by a mean vector μ_{cm}, a diagonal covariance matrix $\Sigma_m = diag(\sigma^2_{m1}, ..., \sigma^2_{mD})$, and the scalar factor θ_m which corresponds to the weight of the Gaussian.

With this relaxation on the covariance matrices, for each example $o_{n,t}$, the goal of the training algorithm is now to force the log-likelihood of its proxy label Gaussian $m_{n,t}$ to be at least one unit greater than the log-likelihood of each Gaussian component of all competing classes. That is, given the training examples $\{(o_{n,t}, y_n, m_{n,t})\}_{n=1}^{N}$, we seek mean vectors μ_{cm} which satisfy the LM-dGMM criterion:

$$\forall c \neq y_n,\ \forall m, \qquad d(o_{n,t}, \mu_{cm}) + \theta_m \geq 1 + d(o_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}}, \quad (3)$$

where $d(o_{n,t}, \mu_{cm}) = \sum_{i=1}^{D} \frac{(o_{n,ti} - \mu_{cmi})^2}{2\sigma^2_{mi}}$.

Afterward, these M constraints are fold into a single one using the soft-max inequality $\min_m a_m \geq -\log \sum_m \exp(-a_m)$. The segment-based LM-dGMM criterion becomes thus:

$$\begin{aligned} &\forall c \neq y_n, \\ &\frac{1}{T_n} \sum_{t=1}^{T_n} \left(-\log \sum_{m=1}^{M} \exp(-d(o_{n,t}, \mu_{cm}) - \theta_m) \right) \\ &\geq\ 1 + \frac{1}{T_n} \sum_{t=1}^{T_n} d(o_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}}. \end{aligned} \quad (4)$$

The loss function to minimize for LM-dGMM is then given by:

$$\begin{aligned} \text{Ł} = \sum_{n=1}^{N} \sum_{c \neq y_n} max \Bigg(0\ ,\ 1 + \frac{1}{T_n} \sum_{t=1}^{T_n} \Bigg(& d(o_{n,t}, \mu_{y_n m_{n,t}}) \\ & + \theta_{m_{n,t}} + \log \sum_{m=1}^{M} \exp(-d(o_{n,t}, \mu_{cm}) - \theta_m) \Bigg) \Bigg). \end{aligned} \quad (5)$$

3 LM-*d*GMM training with *k*-best Gaussians

3.1 Description of the new LM-dGMM modeling

Despite the fact that our LM-dGMM is computationally much faster than the original LM-GMM of [11,14], we still encountered efficiency problems when dealing with high number of Gaussian mixtures. Indeed, even for an easy 50 speakers identification task as the one presented in [12], we could not run the training in a relatively short time with our current implementation. This would imply that large scale applications such as NIST-SRE, where hundreds or thousands of target speakers are available, would be infeasible in reasonable time (for instance, 5460 target speakers are included in the NIST-SRE'2010 core condition, with 610748 trials to process involving 13325 test segments [16]).

In order to develop a fast training algorithm which could be used in large scale applications, we propose to drastically reduce the number of constraints to satisfy in Eq. (4). By doing so, we would drastically reduce the computational complexity of the loss function and its gradient, which are the quantities responsible for most of the computational time. To achieve this goal we propose to use another property of state-of-the-art GMM systems, that is, decision is not made upon all mixture components but only using the k-best scoring Gaussians.

In other words, for each o_n and each class c, instead of summing over the M mixtures in the left side of equation Eq. (4), we would sum only over the k Gaussians with the highest posterior probabilities selected using the GMM of class c. In order to further improve efficiency and reduce memory requirement, we exploit the property reported in [1] about correspondence between MAP adapted GMM mixtures and UBM mixtures. We use the UBM to select one unique set $S_{n,t}$ of k-best Gaussian components per frame $o_{n,t}$, instead of $(C-1)$ sets. This leads to a $(C-1)$ times faster and less memory consuming selection. Thus, the higher the number of target speakers is, the greater computation and memory saving is. More precisely, we now seek mean vectors μ_{cm} that satisfy the large margin constraints in Eq. (6):

$$\begin{aligned} &\forall c \neq y_n, \\ &\frac{1}{T_n}\sum_{t=1}^{T_n}\left(-\log \sum_{m\in S_{n,t}} \exp(-d(o_{n,t},\mu_{cm}) - \theta_m)\right) \\ &\geq\ 1 + \frac{1}{T_n}\sum_{t=1}^{T_n} d(o_{n,t},\mu_{y_n m_{n,t}}) + \theta_{m_{n,t}}. \end{aligned} \tag{6}$$

The loss function becomes:

$$\text{Ł} = \sum_{n=1}^{N} \sum_{c \neq y_n} max\Bigg(0\ ,\ 1 + \frac{1}{T_n} \sum_{t=1}^{T_n} \Bigg(d(o_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}} + \log \sum_{m \in S_{n,t}} \exp(-d(o_{n,t}, \mu_{cm}) - \theta_m) \Bigg) \Bigg). \tag{7}$$

This loss function remains convex and can still be solved using dynamic programming.

3.2 Handling of outliers

We have adopted in our previous work [17] the strategy of [11] to detect outliers and reduce their negative effect on learning. Outliers are detected using the initial GMM models. The original strategy consists on computing the accumulated hinge loss incurred by violations of the large margin constraints in Eq. (6):

$$h_n = \sum_{c \neq y_n} max\Bigg(0\ ,\ 1 + \frac{1}{T_n} \sum_{t=1}^{T_n} \Bigg(d(o_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}} + \log \sum_{m \in S_{n,t}} \exp(-d(o_{n,t}, \mu_{cm}) - \theta_m) \Bigg) \Bigg), \tag{8}$$

and then re-weighting[1] the hinge loss terms in Eq. (7) by using segment weights $s_n = min\Big(1, \frac{1}{h_n}\Big)$.

We propose in this paper a novel and better strategy that outperforms the previous one. We keep the global large margin constraints segmental, but we will apply now a *frame* (feature vectors) weighting scheme. For each feature vector $o_{n,t}$, we calculate $(C-1)$ weights $s^{\mathbf{c}}_{n,t}$ relative to each class $c \neq y_n$. for each $o_{n,t}$ and each competing class c, we compute the loss incurred by violations of the large margin constraints:

$$h^{\mathbf{c}}_{n,t} = \frac{1 + d(o_{n,t}, \mu_{y_n m_{n,t}}) + \theta_{m_{n,t}} + \log \sum_{m \in S_{n,t}} \exp\Big(- d(o_{n,t}, \mu_{cm}) - \theta_m \Big)}{T_n}. \tag{9}$$

$h^{\mathbf{c}}_{n,t}$ measures the decrease in the loss function when an initially misclassified feature vector is corrected during the course of learning. We associate outliers

[1] Note that by setting the segment weights to one, i.e., no handling of outliers is done, the experiments show that the performances degrade.

with values of $h^{\mathbf{c}}_{n,t} > 1$, and in this case we multiply this term by the frame weight $s^{\mathbf{c}}_{n,t} = \frac{1}{h^{\mathbf{c}}_{n,t}}$. The new loss function becomes thus:

$$\mathbf{L} = \sum_{n=1}^{N} \sum_{c \neq y_n} \max \left(0 \ , \ \sum_{t=1}^{T_n} s^{\mathbf{c}}_{n,t} h^{\mathbf{c}}_{n,t} \right). \tag{10}$$

We solve this unconstrained non-linear optimization problem using the second order optimizer LBFGS [18].

In summary, our new and fast training algorithm of LM-dGMM is the following:

- For each class (speaker), initialize with the GMM trained by MAP adaptation of the UBM,
- select Proxy labels using these GMM,
- select the set of k-best UBM Gaussian components for each training frame,
- compute the point weights $s^{\mathbf{c}}_{n,t}$,
- using the LBFGS algorithm, solve the unconstrained non-linear minimization problem:

$$min \quad \text{Ł}. \tag{11}$$

3.3 Evaluation phase

During test, we use the same principle as in the training to achieve fast scoring. Given a test segment of T frames, for each test frame o_t we use the UBM to select the set E_t of k-best scoring proxy labels.

In an identification task, we compute the LM-dGMM likelihoods using only these k labels. The decision rule is thus given as:

$$y = \underset{c}{argmin} \left\{ \sum_{t=1}^{T} -\log \sum_{m \in E_t} \exp(-d(o_t, \mu_{cm}) - \theta_m) \right\}. \tag{12}$$

In a verification task, we compute a match score depending on both the target model $\{\mu_{cm}, \Sigma_m, \theta_m\}$ and the UBM $\{\mu_{Um}, \Sigma_m, \theta_m\}$ for the test hypothesis (trial). The average log likelihood ratio is calculated using only the k labels:

$$\begin{aligned} LLR_{avg} = \frac{1}{T} \sum_{t=1}^{T} \Bigg(& log \sum_{m \in E_t} \exp(-d(o_t, \mu_{cm}) - \theta_m) \\ & - log \sum_{m \in E_t} \exp(-d(o_t, \mu_{Um}) - \theta_m) \Bigg). \end{aligned} \tag{13}$$

This quantity provides a score for the test segment to be uttered by the target model/speaker c.

4 The GSL-NAP system

In this section we briefly describe the GMM supervector linear kernel SVM system (GSL)[4] and its associated channel compensation technique, the Nuisance attribute projection (NAP) [10].

4.1 SVM-GMM supervector

Given an M-components GMM adapted by MAP from the UBM, one forms a GMM supervector by stacking the D-dimensional mean vectors, leading to an MD supervector. This GMM supervector can be seen as a mapping of variable-length utterances into a fixed-length high-dimensional vector, through GMM modeling:

$$\phi(\mathrm{x}) = \begin{bmatrix} \mu_{x1} \\ \vdots \\ \mu_{xM} \end{bmatrix}, \tag{14}$$

where the GMM $\{\mu_{xm}, \Sigma_m, w_m\}$ is trained on the utterance x.

For two utterances x and y, the Kullback-Leibler divergence kernel is defined as:

$$\mathrm{K}(\mathrm{x},\mathrm{y}) = \sum_{m=1}^{M} \left(\sqrt{w_m}\Sigma_m^{-(1/2)}\mu_{xm}\right)^T \left(\sqrt{w_m}\Sigma_m^{-(1/2)}\mu_{ym}\right). \tag{15}$$

The UBM weight and variance parameters are used to normalize the Gaussian means before feeding them into a linear kernel SVM training. This system is referred to as GSL in the rest of the paper.

4.2 Nuisance attribute projection (NAP)

NAP is a pre-processing method that aims to compensate the supervectors by removing the directions of undesired sessions variability, before the SVM training [10]. NAP transforms a supervector ϕ to a compensated supervector $\hat{\phi}$:

$$\hat{\phi} = \phi \ - \ \mathbf{S}(\mathbf{S}^T\phi), \tag{16}$$

using the eigenchannel matrix $\mathbf{S}$, which is trained using several recordings (sessions) of various speakers.

In the following, (h, s) will indicate the *session* h of the *speaker* s. Given a set of expanded recordings:

$$\{\phi(1,s_1)\cdots\phi(h_1,s_1)\cdots\phi(1,s_N)\cdots\phi(h_N,s_N)\}, \tag{17}$$

of N different speakers, with h_i different sessions for each speaker s_i, one first removes the speakers variability by subtracting the mean of the supervectors within each speaker $\{\overline{\phi(s_i)}\}$:

$$\forall s_i, \forall h, \qquad \phi(h,s_i) = \phi(h,s_i) \ - \ \overline{\phi(s_i)}. \tag{18}$$

The resulting supervectors are then pooled into a single matrix:

$$\mathbf{C} = \left[\phi(1, s_1) \cdots \phi(h_1, s_1) \cdots \phi(1, s_N) \cdots \phi(h_N, s_N)\right], \tag{19}$$

representing the intersession variations. One identifies finally the subspace of dimension R where the variations are the largest by solving the eigenvalue problem on the covariance matrix $\mathbf{CC}^T$, getting thus the projection matrix $\mathbf{S}$ of a size $MD \times R$. This system is referred to as GSL-NAP in the rest of the paper.

5 Symmetrical Factor Analysis (SFA)

In this section we describe the symmetrical variant of the Factor Analysis model (SFA) [8,9] (Factor Analysis was originally proposed in [6,7]). In the mean supervector space, a speaker model can be decomposed into three different components:

- a session-speaker independent component (the UBM model),
- a speaker dependent component,
- a session dependent component.

The session-speaker model, can be written as [8]:

$$\mathbf{M}_{(h,s)} \;=\; \mathbf{M} \;+\; \mathbf{D}\mathbf{y}_s \;+\; \mathbf{U}\mathbf{x}_{(h,s)}, \tag{20}$$

where

- $\mathbf{M}_{(h,s)}$ is the session-speaker dependent supervector mean (an MD vector),
- $\mathbf{M}$ is the UBM supervector mean (an MD vector),
- $\mathbf{D}$ is a $MD \times MD$ diagonal matrix, where $\mathbf{DD}^T$ represents the a priori covariance matrix of $\mathbf{y}_s$,
- $\mathbf{y}_s$ is the speaker vector (speaker offset), an MD vector assumed to follow a standard normal distribution $\mathcal{N}(0, \mathrm{I})$,
- $\mathbf{U}$ is the session variability matrix of low rank R (an $MD \times R$ matrix),
- $\mathbf{x}_{(h,s)}$ are the channel factors (session offset), an R vector (theoretically, not dependent on s) assumed to follow a standard normal distribution $\mathcal{N}(0, \mathrm{I})$.

$\mathbf{D}\mathbf{y}_s$ and $\mathbf{U}\mathbf{x}_{(h,s)}$ represent respectively the speaker dependent component and the session dependent component [9].

The factor analysis modeling starts by estimating the $\mathbf{U}$ matrix, using different recordings per speaker. The matrix $\mathbf{U}$ is theoretically similar to the channel matrix $\mathbf{S}$ of NAP, and it also requires many recordings to identify accurately the subspace where intersession variability is high. However, the matrix $\mathbf{U}$ estimation is computationally less efficient than the matrix $\mathbf{S}$ one. Given the fixed parameters $(\mathbf{M}, \mathbf{D}, \mathbf{U})$, the target models are then compensated by eliminating the session mismatch directly in the model domain. Whereas, the compensation in the test is performed at the frame level (feature domain).

6 Experimental results

We perform experiments on the NIST-SRE'2006 [19] speaker identification and verification tasks and compare the performances of the baseline GMM, the LM-dGMM and the SVM systems, with and without using channel compensation techniques. The comparisons are made on the male part of the NIST-SRE'2006 core condition (1conv4w-1conv4w). In the identification task, performances are measured in term of the speaker identification rate. In the verification task, they are assessed using Detection Error Tradeoff (DET) plots and measured in terms of equal error rate (EER) and minimum of detection cost function (minDCF) which is calculated following NIST criteria [20].

For front-end processing, we follow the same procedure as in [9]. The feature extraction is carried out by the filter-bank based cepstral analysis tool Spro [21]. Bandwidth is limited to the 300-3400Hz range. 24 filter bank coefficients are first computed over 20ms Hamming windowed frames at a 10ms frame rate and transformed into Linear Frequency Cepstral Coefficients (LFCC) [22]. Consequently, the feature vector is composed of 50 coefficients including 19 LFCC, their first derivatives, their 11 first second derivatives and the delta-energy. The LFCCs are preprocessed by Cepstral Mean Subtraction and variance normalization [23]. We applied an energy-based voice activity detection to remove silence frames, hence keeping only the most informative frames. Finally, the remaining parameter vectors are normalized to fit a zero mean and unit variance distribution.

We use the state-of-the-art open source software ALIZE/Spkdet [9, 24] for GMM, SFA, GSL and GSL-NAP modeling. A male-dependent UBM is trained using all the telephone data from the NIST-SRE'2004. Then we train a MAP adapted GMM for the 349 target speakers belonging to the primary task. The identification is made on a list of 539554 trials (involving 1546 test segments), whereas the verification task uses a shorter list of 22123 trials (involving 1601 test segments) for test. Score normalization techniques are not used in our experiments. The so MAP adapted GMM define the baseline GMM system, and are used as initialization for the LM-dGMM one. The GSL system uses a list of 200 impostor speakers from the NIST-SRE'2004, on the SVM training. The LM-dGMM-SFA system is initialized by model domain compensated GMM, which are then discriminated using feature domain compensated data. The session variability matrix $\mathbf{U}$ of SFA and the channel matrix $\mathbf{S}$ of NAP, both of rank $R = 40$, are estimated on NIST-SRE'2004 data using 2934 utterances of 124 different male speakers.

Table 1 presents the speaker identification accuracy scores of the various systems. Table 2 presents the speaker verification scores (EER and minDCF). We show performances using GMMs with 256 and 512 Gaussian components ($M = 256, 512$). All the scores are obtained with the 10 best proxy labels selected using the UBM, $k = 10$. The actual large margin systems adopt a segmental weighting approach.

The results of Table 1 and Table 2 show that, without SFA channel compensation, the LM-dGMM system outperforms the classical generative GMM one,

Table 1 Speaker identification rates with GMM, Large Margin diagonal GMM and GSL models, with and without channel compensation

System	Speaker identification rate	
	256 Gaussians	512 Gaussians
GMM	75.87%	77.88%
LM-dGMM	77.62%	78.40%
GSL	81.50%	82.21%
GSL-NAP	87.26%	87.77%
GMM-SFA	89.26%	90.75%
LM-dGMM-SFA	89.65%	**91.27%**

Table 2 EERs(%) and minDCFs(x100) of GMM, Large Margin diagonal GMM and GSL systems with and without channel compensation

System	256 Gaussians		512 Gaussians	
	EER	minDCF(x100)	EER	minDCF(x100)
GMM	9.43%	4.26	9.74%	4.18
LM-dGMM	8.97%	3.97	9.66%	4.12
GSL	7.39%	3.41	7.23%	3.44
GSL-NAP	6.40%	2.72	5.90%	2.73
GMM-SFA	6.15%	2.41	5.53%	2.18
LM-dGMM-SFA	5.58%	2.29	**5.02%**	**2.18**

Table 3 GSL performance using different values of C and average number of support vectors ($M = 256$)

C	Number of support vectors	Identification rate	EER	minDCF(x100)
2^{-4}	46	78.33%	7.81%	3.71
2^{-3}	49	78.20%	7.85%	3.72
2^{-2}	51	78.20%	7.85%	3.72
2^{-1}	52	78.20%	7.83%	3.72
2^{0}	52	81.50%	7.40%	3.41
2^{1}	52	**81.50%**	**7.39%**	**3.41**
2^{2}	52	81.50%	7.39%	3.41
2^{3}	52	81.50%	7.39%	3.41
2^{4}	52	81.50%	7.40%	3.41

Table 4 GSL-NAP performance using different values of C and average number of support vectors ($M = 256$)

C	Number of support vectors	Identification rate	EER	minDCF(x100)
2^{-4}	63	84.22%	6.77%	2.99
2^{-3}	70	84.15%	6.77%	3.00
2^{-2}	75	84.09%	6.78%	2.98
2^{-1}	77	84.15%	6.80%	2.98
2^{0}	78	**87.26%**	**6.40%**	**2.72**
2^{1}	78	87.19%	6.44%	2.71
2^{2}	78	87.19%	6.44%	2.71
2^{3}	78	87.19%	6.44%	2.71
2^{4}	78	87.19%	6.44%	2.71

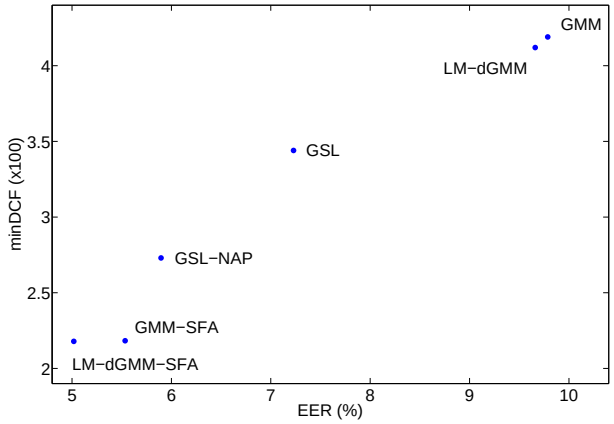

Fig. 1 EER and minDCF performances for GMM, LM-dGMM and GSL systems with and without channel compensation

however it does yield worse performances than the discriminative approach GSL. Nonetheless, when applying channel compensation techniques, compensated models outperform the non-compensated ones as expected, but the LM-dGMM-SFA system significantly outperforms the GSL-NAP and GMM-SFA ones in the two tasks. Our best system achieves 91.27% speaker identification rate, while the best GSL-NAP achieves 87.77%. This leads to a 3.5% improvement. In verification, the LM-dGMM-SFA and GSL-NAP achieve respectively 5.02% and 5.90% equal error rates, and $2.18 * 10^{-2}$ and $2.73 * 10^{-2}$ minDCF values. This shows that LM-dGMM-SFA yields relative reductions of EER and minDCF of about 14.92% and 20.15% over the GSL-NAP system. Moreover, The performances of the GMM-SFA system show that LM-dGMM-SFA yields relative reductions of speaker identification rate and EER of about 0.57% and 9.22% over this system.

It is known that SVM performances are sensitive to the C parameter. We have thus used different values of C and reported the best scores of the SVM systems in Table 1 and Table 2. This can be seen in Table 3 and Table 4 which show the scores obtained using different values of C for GSL and GSL-NAP with $M = 256$. We also report in these tables the average number of support vectors.

Figure 1 displays the EER and minDCF performances of all systems, with and without channel compensation, for models with 512 Gaussian components ($M = 512$). Figure 2 shows DET plots for LM-dGMM and GSL systems with and without channel compensation, for models with 512 Gaussian components. One can see that LM-dGMM-SFA outperforms GSL and GSL-NAP at all operating points.

Table 5 gives the EER scores of LM-dGMM and LM-dGMM-SFA systems using the two weighting strategies, for models with 512 Gaussian components.

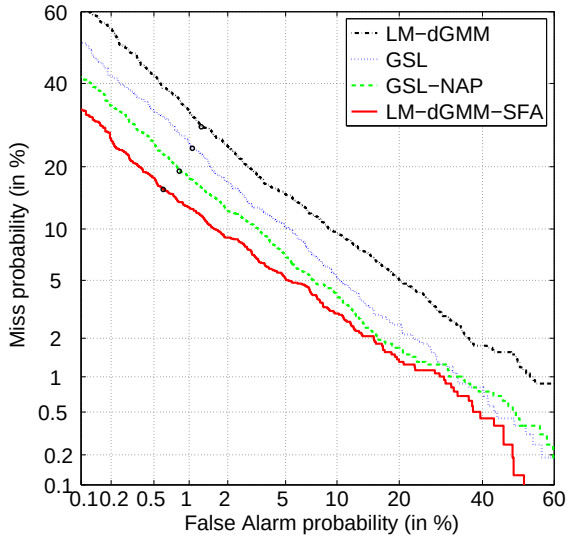

Fig. 2 DET plots for LM-dGMM and GSL systems with and without channel compensation

Table 5 Segmental weighting strategy vs frame weighting strategy

System		EER
Segmental weighting	LM-dGMM	9.66%
	LM-dGMM-SFA	5.02%
Frame weighting	LM-dGMM	**9.47%**
	LM-dGMM-SFA	**4.89%**

One can see that the frame weighting approach further improves the LM-dGMM (+SFA) performance. All these results show that our fast Large Margin GMM discriminative learning algorithm not only allows efficient training but also achieves better speaker recognition (identification and verification) performances than a state-of-the-art discriminative technique.

7 Conclusion

We proposed a new algorithm for discriminative learning of diagonal GMM under a Large-Margin criterion. Our algorithm is highly efficient which makes it well suited to process large scale databases such as in NIST-SRE campaigns. We also developed a frame weighting strategy to detect and handle outliers in training data. This strategy yields further improvement in performances. We carried out experiments on full speaker identification and verification tasks under the NIST-SRE'2006 core condition. Combined with the SFA channel compensation technique, the resulting algorithm significantly outperforms the state-of-the-art speaker recognition discriminative approach GSL-NAP. An-

other major advantage of our method is that it outputs diagonal GMM models. Thus, broadly used GMM techniques/softwares such as SFA or ALIZE/Spkdet can be readily applied in our framework. Our future work will consist in improving margin selection. Like in SVM, this should indeed significantly improve the performances. We emphasize also that, while we have applied our algorithm to speaker recognition, it can be actually applied in any other classification task which involves supervised learning of diagonal GMM.

Acknowledgements The authors would like to thank the anonymous reviewers for their helpful comments.

References

1. Reynolds DA, Quatieri TF, Dunn RB (2000) Speaker verification using adapted Gaussian mixture models. Digit Signal Processing 10(1-3):19-41
2. Keshet J, Bengio S (2009) Automatic speech and speaker recognition: Large margin and kernel methods. Wiley, Hoboken, New Jersey
3. Louradour J, Daoudi K, Bach F (2007) Feature space mahalanobis sequence kernels: Application to svm speaker verification. IEEE Trans Audio Speech Lang Processing 15(8):2465-2475
4. Campbell WM, Sturim DE, Reynolds DA (2006) Support vector machines using GMM supervectors for speaker verification. IEEE Signal Processing Lett 13(5):308-311
5. Campbell WM, Sturim DE, Reynolds DA, Solomonoff A (2006) SVM based speaker verification using a GMM supervector kernel and NAP variability compensation. In: Proc. of ICASSP, vol 1, pp I-97-I-100
6. Kenny P, Boulianne G, Dumouchel P (2005) Eigenvoice modeling with sparse training data. IEEE Trans Speech Audio Processing 13(3):345-354
7. Kenny P, Boulianne G, Ouellet P, Dumouchel P (2007) Speaker and session variability in GMM-based speaker verification. IEEE Trans Audio Speech Lang Processing 15(4):1448-1460
8. Matrouf D, Scheffer N, Fauve BGB, Bonastre J-F (2007) A straightforward and efficient implementation of the factor analysis model for speaker verification. In: Proc. of Interspeech, pp 1242-1245
9. Fauve BGB, Matrouf D, Scheffer N, Bonastre J-F, Mason JSD (2007) State-of-the-Art Performance in Text-Independent Speaker Verification through Open-Source Software. IEEE Trans Audio Speech Lang Processing 15(7):1960-1968
10. Solomonoff A, Campbell WM, Boardman I (2005) Advances in Channel Compensation for SVM Speaker Recognition. In: Proc. of ICASSP, vol 1, pp 629-632
11. Sha F, Saul LK (2006) Large margin Gaussian mixture modeling for phonetic classification and recognition. In: Proc. of ICASSP, vol 1, pp 265-268
12. Jourani R, Daoudi K, André-Obrecht R, Aboutajdine D (2010) Large Margin Gaussian mixture models for speaker identification. In: Proc. of Interspeech, pp 1441-1444
13. http://www.itl.nist.gov/iad/mig//tests/sre/
14. Sha F (2007) Large margin training of acoustic models for speech recognition. Ph.D. dissertation, University of Pennsylvania
15. Bishop CM (2006) Pattern recognition and machine learning. Springer Science+Business Media, LLC, New York
16. NIST (2010) The NIST Year 2010 Speaker Recognition Evaluation Plan. http://www.itl.nist.gov/iad/mig//tests/sre/2010/NIST_SRE10_evalplan.r6.pdf. Accessed 10 February 2010
17. Daoudi K, Jourani R, André-Obrecht R, Aboutajdine D (2011) Speaker Identification Using Discriminative Learning of Large Margin GMM. In: Lu B-L, Zhang L, Kwok J (eds.) Neural Information Processing. LNCS, vol 7063. Springer, Heidelberg, pp 300-307
18. Nocedal J, Wright SJ (1999) Numerical optimization. Springer verlag, New York

19. NIST (2006) The NIST Year 2006 Speaker Recognition Evaluation Plan. http://www.itl.nist.gov/iad/mig/tests/spk/2006/sre-06_evalplan-v9.pdf. Accessed 30 November 2009
20. Przybocki M, Martin A (2004) NIST Speaker Recognition Evaluation Chronicles. In: Proc. of Odyssey-The Speaker, Language Recognition Workshop, pp 15-22
21. Gravier G (2003) SPro: "Speech Signal Processing Toolkit". https://gforge.inria.fr/projects/spro. Accessed 30 November 2009
22. Davis S, Mermelstein P (1980) Comparison of parametric representations for mono-syllabic word recognition in continuously spoken sentences. IEEE Trans Acoust Speech Signal Processing 28(4):357-366
23. Viikki O, Laurila K (1998) Cepstral domain segmental feature vector normalization for noise robust speech recognition. Speech Communication 25(1-3):133-147
24. Bonastre J-F, Scheffer N, Matrouf D, Fredouille C, Larcher A, Preti A, Pouchoulin G, Evans N, Fauve BGB, Mason JSD (2008) ALIZE/SpkDet: a state-of-the-art open source software for speaker recognition. In: Proc. of Odyssey-The Speaker and Language Recognition Workshop, paper 020