We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

12

Discriminative Universal Background Model Training for Speaker Recognition

Wei-Qiang Zhang and Jia Liu
Department of Electronic Engineering, Tsinghua University
China

1. Introduction

Speaker recognition (SRE), also called as voiceprint recognition, is the problem of determining the identity of the speaker from a sample of speech signal. It is an important branch of speech signal processing and has many potential applications such as in telephone banking, access control, information security, law enforcement and other forensic applications (Bimbot et al., 2004; Campbell Jr., 1997; Cole et al., 1997; Kinnunen & Li, 2010; Reynolds, 2002).

Compared with other biometrics techniques, speaker recognition has its own advantages: (1) It is very convenient, natural and low-cost to acquire the speech sample: it does not need the special devices; the telephone, mobile phone or ordinary microphone is adequate. (2) It can be used remotely: with the ubiquitous telecommunications networks and the Internet, the speech sample can be easily transferred through telephone or VoIP, which makes the remote recognition possible. (3) The speech sample contains many inborn characters: from the speech, we can extract some information about vocal tract, mouth, tongue, soft palate, nasal cavity, and etc. (4) The speech sample also contains some acquired characters, such as tone, volume, pace, rhythm, rhetoric, which reflect speaker's place of living, education level, and some personal habits information.

In speaker recognition, the Gaussian mixture model - universal background model (GMM-UBM) is a classical yet widely used method for text-independent speaker verification (Reynolds et al., 2000). In this method, the target speaker is modeled as a GMM and the imposters are modeled as a UBM. When testing, the speech sample is scored as likelihood by the GMM and UBM respectively, and then the likelihood ratio hypothesis test is used for speaker verification. Besides the GMM-UBM, several other methods are developed recently. The most successful ones include the support vector machine using GMM supper vector (GSV-SVM) (Campbell et al., 2006), which concatenate the GMM mean vectors as the input for SVM training and test, and joint factor analysis (JFA) (Kenny et al., 2007), which jointly models the channel subspace and the speaker subspace. Although other methods achieve rapid progress, GMM-UBM is still the basis for their developments.

As the meanwhile, the discriminative technologies, such as minimum classification error (MCE), maximum mutual information (MMI), minimum phone error (MPE), feature domain MPE (fMPE), have been achieved great success in speech recognition and language recognition (Burget et al., 2006; Juang & Katagiri, 1992; Povey & Kingsbury, 2007; Woodland & Povey, 2002).

In speaker recognition, many discriminative approaches have been reported. As for the GMM-UBM method, the approaches can be divided into two catalogs. (1) Some approaches aim to jointly train the target speaker model and corresponding anti-model. For example, In (Korkmazskiy & Juang, 1996), the MCE criterion is used to adapt talker model (i.e., speaker model) parameters and the corresponding anti-talker model parameters. In (Rosenberg et al., 1998), the minimum verification error (MVE) criterion is used to train the speaker and anti-speaker models and also the decision threshold. In (Ma & Chang, 2003), MMI, MCE, figure of merit (FOM) criteria are used to train the target speaker model and corresponding imposter model. In (Angkititrakul & Hansen, 2007), the training process is divided into two stages: in the first stage, the MCE is used minimize the classification error among the in-set speaker models; in the second stage, the MVE is used to minimize the verification error between the in-set and background models. In (Chao et al., 2008; 2009), the MVE methods are used to reinforce the discriminability between the target speaker model and the target speaker dependent anti-model. (2) Other approaches attempt to discriminatively adapt the target speaker model from the UBM, which can be viewed as the modification of the classical maximum a posteriori (MAP) adaptation (Gauvain & Lee, 1994). For example, In (Zhao et al., 2006), a new speaker adaptation method which combines MAP and reference speaker weighting (RSW) adaptation is presented in a hierarchical multigrained mode. In (Longworth & Gales, 2006), an MMI based adaptation method is reported.

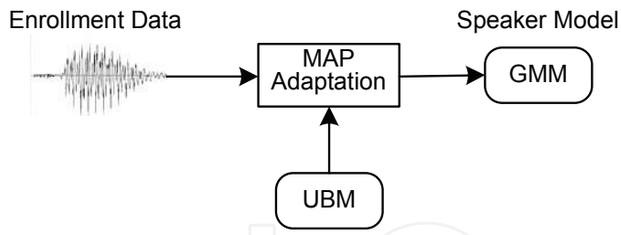
From the discriminative approaches mentioned above, we can find that the UBM is either unchanged or adapted to the target speaker dependent anti-model. If the anti-model is target speaker dependent, it will not be the *universal* background model anymore. But sometimes we have to use the UBM. For example, for fast scoring in GMM-UBM method, we need UBM to determine the orders of mixtures; in the state-of-the-art JFA and GSV-SVM methods, we need UBM to calculate the statistics or the GMM mean vectors. So herein, we want to discriminatively train the UBM to improve its performance.

In order to improve the quality of UBM, many researchers try to select suitable data. For example, in (Hasan et al., 2010; Huang & Li, 2010; Zhang et al., 2010), the data selection based on sub-sampling, maximum entropy and vocal tract length methods are introduced. But as the authors known, there is little report on training the UBM discriminately.

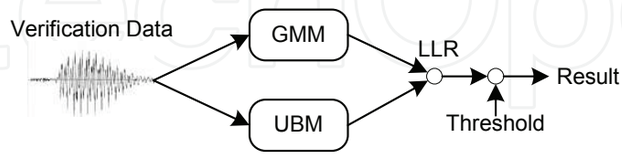
In this chapter, we will discuss the discriminative UBM training method. Firstly we will give a brief review of the GMM-UBM method. After that, we propose our discriminative UBM training method. We will discuss its principle and implementation details. At last, the presented method will be evaluated through large-scale experiments. The results on NIST speaker recognition evaluation dataset will be reported.

2. Overview of GMM-UBM

The GMM-UBM can be viewed as a likelihood-ratio detector: the UBM is trained to represent the speaker-independent distribution of features while the GMM is adapted from the UBM to depict the individual speaker characteristics. In GMM-UBM system, as shown in Fig. 1, a UBM is firstly trained to capture the general characteristics of all the speakers, so it is called *universal* background model. The UBM parameters include weights, mean vectors and covariance matrices, usually denoted by $\boldsymbol{\lambda} = \{w_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}_{m=1}^{M}$, where M is the number of Gaussian mixtures. In speaker recognition, usually the value of M is large, varied from several hundred to several thousand, and the covariance matrices are often set in diagonal form, which facilitates the fast computation.



(a) Speaker enrollment



(b) Speaker verification

Fig. 1. The basic framework of GMM-UBM system

For the t-th frame of feature vector $\boldsymbol{x}_t$, the UBM gives the likelihood as

$$p(\boldsymbol{x}_t|\boldsymbol{\lambda}) = \sum_{m=1}^{M} w_m \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \tag{1}$$

For a T-frame segment $\boldsymbol{x} = \{\boldsymbol{x}_t\}_{t=1}^{T}$, the likelihood is approximated via frame independent assumption as

$$p(\boldsymbol{x}|\boldsymbol{\lambda}) = \prod_{t=1}^{T} p(\boldsymbol{x}_t|\boldsymbol{\lambda}) \tag{2}$$

Usually, the logarithm form of likelihood is used for calculation.
The UBM is usually trained by using the Baum-Welch algorithm (Huang et al., 2000) based on a maximum likelihood (ML) criterion. The Baum-Welch algorithm is in fact a type of expectation-maximization (EM) algorithm and can be implemented iteratively. Suppose the current parameters are obtained, then the new parameters can be updated as

$$w_m^{\text{new}} = \frac{n_m}{T} \tag{3}$$

$$\boldsymbol{\mu}_m^{\text{new}} = \frac{\boldsymbol{f}_m}{n_m} \tag{4}$$

$$\boldsymbol{\Sigma}_m^{\text{new}} = \frac{\boldsymbol{S}_m}{n_m} \tag{5}$$

where n_m, $\boldsymbol{f}_m$ and $\boldsymbol{S}_m$ are the zero-th order, first order and second order statistics

$$n_m = \sum_{t=1}^{T} \gamma_m(\boldsymbol{x}_t) \tag{6}$$

$$\boldsymbol{f}_m = \sum_{t=1}^{T} \gamma_m(\boldsymbol{x}_t)\boldsymbol{x}_t \tag{7}$$

$$\boldsymbol{S}_m = \sum_{t=1}^{T} \gamma_m(\boldsymbol{x}_t)\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}} \tag{8}$$

$\gamma_m(\boldsymbol{x}_t)$ is m-th mixture of occupation probability

$$\gamma_m(\boldsymbol{x}_t) = \frac{w_m \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{m'=1}^{M} w_{m'} \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{m'}, \boldsymbol{\Sigma}_{m'})} \tag{9}$$

The initial parameters can be set as: $w_m = 1/M$, $\boldsymbol{\Sigma}_m = \boldsymbol{I}$ and each $\boldsymbol{\mu}_m$ can be randomly selected from the training samples or use the finer Lind-Buzo-Gray (LBG) algorithm to get the initial values. Through enough iterations, the local maximum of the likelihood can be achieved and the parameters become stable.
After the UBM is trained, in the enrollment stage, the mean vectors of UBM is adapted by using enrollment data $\boldsymbol{x}^s$ of speaker s under MAP criterion (Gauvain & Lee, 1994).

$$\boldsymbol{\mu}_m^s = \frac{n_m}{n_m + \gamma} \frac{\boldsymbol{f}_m}{n_m} + \frac{\gamma}{n_m + \gamma} \boldsymbol{\mu}_m \tag{10}$$

where n_m and $\boldsymbol{f}_m$ are calculated by using enrollment segment $\boldsymbol{x}^s$, γ is the relevance factor, and usually set as 16 (Reynolds et al., 2000). Note that, the weights and covariance matrices are not updated. Thus, the parameters for GMM of speaker s are $\boldsymbol{\lambda}^s = \{w_m, \boldsymbol{\mu}_m^s, \boldsymbol{\Sigma}_m\}_{m=1}^M$.
In the speaker verification stage, the log-likelihood-ratio (LLR) of the test segment $\boldsymbol{x}^r$ is calculated by using the GMM and the UBM, and compared with threshold to give the last acceptance or rejection decision.

$$s(\boldsymbol{x}^r, \boldsymbol{\lambda}^s, \boldsymbol{\lambda}) = \frac{1}{T_r}(\log p(\boldsymbol{x}^r|\boldsymbol{\lambda}^s) - \log p(\boldsymbol{x}^r|\boldsymbol{\lambda})) \gtrless s_{\text{th}} \tag{11}$$

where T_r is the number of frames of verification segment $\boldsymbol{x}^r$. This equation can be expanded as

$$s(\boldsymbol{x}^r, \boldsymbol{\lambda}^s, \boldsymbol{\lambda}) = \frac{1}{T_r} \sum_{t=1}^{T_r} (\log \sum_{m=1}^{M} w_m^s \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_m^s, \boldsymbol{\Sigma}_m^s) - \log \sum_{m=1}^{M} w_m \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)) \tag{12}$$

Note that in our case, $w_m^s = w_m$, $\boldsymbol{\Sigma}_m^s = \boldsymbol{\Sigma}_m$, and $\boldsymbol{\mu}_m^s$ is adapted from $\boldsymbol{\mu}_m$. This means that the scores calculated by the corresponding mixtures of GMM and UBM are approximately equal. According to the property of GMM, we know that each feature frame is located at a local region, that is to say, most mixtures will give very small scores for each frame. So we can neglect these mixtures and only calculate top N mixtures for LLR scoring.

$$s(\boldsymbol{x}^r, \boldsymbol{\lambda}^s, \boldsymbol{\lambda}) = \frac{1}{T_r} \sum_{t=1}^{T_r} (\log \sum_{n=1}^{N} w_{m_n(t)}^s \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_{m_n(t)}^s, \boldsymbol{\Sigma}_{m_n(t)}^s) - \log \sum_{n=1}^{N} w_{m_n(t)} \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_{m_n(t)}, \boldsymbol{\Sigma}_{m_n(t)})) \tag{13}$$

where $\{m_n(t)\}_{n=1}^N$ are the top N scoring mixture indices calculated by UBM for the frame $\boldsymbol{x}_t$. This fast scoring strategy is introduced in (Reynolds et al., 2000) and widely used in GMM-UBM method and other similar circumstances.

3. Discriminative UBM training

From the above section, we can see that the UBM is trained under ML criterion. This criterion is asymptotically optimal, in another word, it is optimal if there are infinite amount of training data. In practice, this condition can not be satisfied. The available training data is always limited. As a consequence, likelihood based training can not guarantee optimal performance.

For speaker verification systems, the most important performance measure is the verification errors. So we borrow the minimum verification error (MVE) criterion (Rosenberg et al., 1998) to develop a discriminative UBM training method.
Note that our motivation is different to other discriminative approaches for speaker recognition: we only want to obtain a high quality UBM. The flowchart is showed in Fig. 2. We can observed that, the enrollment data and verification data are all our *training* data.

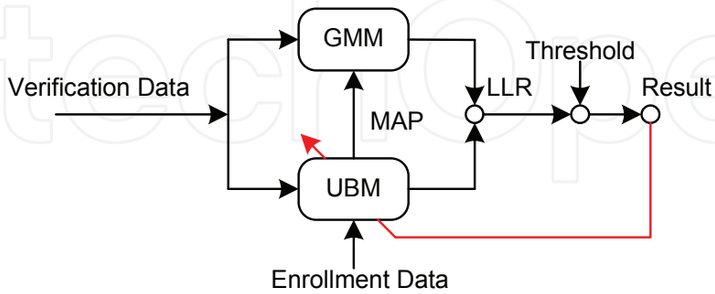


Fig. 2. The flowchart for discriminative UBM training

3.1 Discriminative framework

Similar to MCE criterion (Juang & Katagiri, 1992), the MVE criterion also can optimized by using generalized probabilistic descent (GPD) framework. To implement it, a *smoothed* loss function should be defined first and then the gradient descent method is used to obtain the (local) minimum of the loss function.
Firstly, we define the false verification function (similar to discriminant function in MCE) as

$$d(i,\boldsymbol{\lambda}) = [\log p(\boldsymbol{x}^r|\boldsymbol{\lambda}^s) - \log p(\boldsymbol{x}^r|\boldsymbol{\lambda}) - s_{\text{th}}]\,\text{sign}(i) \tag{14}$$

where i denotes the i-th trial which involves s-th speaker model and r-th verification segment, and

$$\text{sign}(i) = \begin{cases} -1 & \text{if } i \text{ is target trial} \\ 1 & \text{if } i \text{ is non-target trial} \end{cases} \tag{15}$$

From (14), we can see that $d(i,\boldsymbol{\lambda}) > 0$ indicates trial i is a false verification and $d(i,\boldsymbol{\lambda}) < 0$ implies a correct verification. The value of the false verification function indicates the distortion between the models and the corresponding training data. The larger the false verification function is, the more adjustment of the model parameters is required to improve the verification performance.
Next, we will define the loss function. In general, the loss function is a function of the false verification function. Obviously, the loss function and the false verification function can be defined individually. Loss function is used to show the cost of mis-verification a trial. It is required that the loss function should be a differentiable, and monotonically non-decreasing function. Usually, sigmoid function is a good choice. The gradients of this function are easy to be obtained. The loss function is defined as

$$l(i,\boldsymbol{\lambda}) = \frac{\text{cost}(i)}{1+\exp\{-\alpha d(i,\boldsymbol{\lambda})\}} \tag{16}$$

where α is the slop parameter of sigmoid function. $\text{cost}(i)$ is the cost of false verification of i-th trial.

Then, the objective function (total loss) need to be minimized is

$$L(\boldsymbol{\lambda}) = \sum_{i=1}^{I} l(i,\boldsymbol{\lambda}) u(l(i,\boldsymbol{\lambda}) + \delta) \tag{17}$$

where $u(\cdot)$ is a unit function

$$u(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if x} < 0 \end{cases} \tag{18}$$

and δ is a small positive number.
From (17), it is clear that the incorrectly verified trials (for the trials such that $l(i,\boldsymbol{\lambda}) > 0$) and the correctly verified but near the decision boundary trials (for the trials such that $0 \geq l(i,\boldsymbol{\lambda}) \geq -\delta$) are used for training.
We can use the gradient descent algorithm to optimize this objective function. Note that herein we only discuss the mean vectors. Other parameters can be obtained similarly. The update formula is

$$\boldsymbol{\mu}_m(n+1) = \boldsymbol{\mu}_m(n) - \varepsilon_n \frac{\partial L(\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} \tag{19}$$

where ε_n is the step factor.
In practise, we can use Baum-Welch algorithm to obtain the parameters of UBM initially, then use (19) to update its mean vectors discriminatively.

3.2 Gradients

For the gradient descent algorithm, the most important step is to obtain the gradients of the objective function. It is not easy but straightforward. We will solve this problem step by step. The gradient of the objective function w.r.t the mean vector is

$$\begin{aligned} \frac{\partial L(\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} &= \sum_{i=1}^{I} \frac{\partial l(i,\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} \\ &= \sum_{i=1}^{I} \frac{\partial l(i,\boldsymbol{\lambda})}{\partial d(i)} \frac{\partial d(i,\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} \\ &= \sum_{i=1}^{I} \frac{\alpha}{\text{cost}(i)} l(i,\boldsymbol{\lambda}) \left[\text{cost}(i) - l(i,\boldsymbol{\lambda})\right] \frac{\partial s(i,\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} \text{sign}(i) \end{aligned} \tag{20}$$

where $\partial s(i,\boldsymbol{\lambda})/\partial \boldsymbol{\mu}_m$ consists of two items

$$\frac{\partial s(i,\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} = \frac{1}{T_r} \left(\frac{\partial \log p(\boldsymbol{x}^r|\boldsymbol{\lambda}^s)}{\partial \boldsymbol{\mu}_m} - \frac{\partial \log p(\boldsymbol{x}^r|\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} \right) \tag{21}$$

For the first item, because

$$\begin{aligned} \log p(\boldsymbol{x}^r|\boldsymbol{\lambda}) &= \sum_{t=1}^{T_r} \log p(\boldsymbol{x}_t^r|\boldsymbol{\lambda}) \\ &= \sum_{t=1}^{T_r} \log \sum_{m=1}^{M} w_m \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \end{aligned} \tag{22}$$

thus, we can obtain

$$\frac{\partial \log p(\boldsymbol{x}^r|\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} = \sum_{t=1}^{T_r} \frac{1}{p(\boldsymbol{x}_t^r|\boldsymbol{\lambda})} \frac{\partial w_m \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\partial \boldsymbol{\mu}_m}$$
$$= \sum_{t=1}^{T_r} -2\gamma_m(\boldsymbol{x}_t^r)\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_t^r - \boldsymbol{\mu}_m) \tag{23}$$

For the second item, according to (10), we know that $\{\boldsymbol{\mu}_m^s\}_{m=1}^M$ is a function of $\{\boldsymbol{\mu}_m\}_{m=1}^M$. Based on the chain rule for derivation, we have

$$\frac{\partial \log p(\boldsymbol{x}_t^r|\boldsymbol{\lambda}^s)}{\partial \boldsymbol{\mu}_m} = \sum_{t=1}^{T_r} \frac{1}{p(\boldsymbol{x}^r|\boldsymbol{\lambda}^s)} \sum_{m'=1}^{M} \frac{\partial (\boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}}{\partial \boldsymbol{\mu}_m} \frac{\partial w_{m'}^s \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_{m'}^s, \boldsymbol{\Sigma}_{m'}^s)}{\partial \boldsymbol{\mu}_{m'}^s} \tag{24}$$

Similar to (23), we can obtain

$$\frac{1}{p(\boldsymbol{x}_t^r|\boldsymbol{\lambda}^s)} \frac{\partial w_{m'}^s \mathcal{N}(\boldsymbol{x}_t^r; \boldsymbol{\mu}_{m'}^s, \boldsymbol{\Sigma}_{m'}^s)}{\partial \boldsymbol{\mu}_{m'}^s} = -2\gamma_{m'}^s(\boldsymbol{x}_t^r)(\boldsymbol{\Sigma}_{m'}^s)^{-1}(\boldsymbol{x}_t^r - \boldsymbol{\mu}_{m'}^s) \tag{25}$$

where $\gamma_{m'}^s(\boldsymbol{x}_t^r)$ is the m'-th mixture occupation of $\boldsymbol{x}_t^r$ calculated by GMM of speaker s. Substitute (25) to (24), we get

$$\frac{\partial \log p(\boldsymbol{x}_t^r|\boldsymbol{\lambda}^s)}{\partial \boldsymbol{\mu}_m} = \sum_{t=1}^{T_r} \sum_{m'=1}^{M} -2\gamma_{m'}^s(\boldsymbol{x}_t^r) \frac{\partial (\boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}}{\partial \boldsymbol{\mu}_m} (\boldsymbol{\Sigma}_{m'}^s)^{-1}(\boldsymbol{x}_t^r - \boldsymbol{\mu}_{m'}^s) \tag{26}$$

Next, we will get $\partial(\boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}/\partial \boldsymbol{\mu}_m$. This can be divided into two cases. When $m' = m$

$$\frac{\partial (\boldsymbol{\mu}_m^s)^{\mathrm{T}}}{\partial \boldsymbol{\mu}_m} = \frac{(\sum_{t=1}^{T_s} \frac{\partial \gamma_m(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m}(\boldsymbol{x}_t^s)^{\mathrm{T}} + \gamma \boldsymbol{I})(n_m + \gamma) - \frac{\partial n_m}{\partial \boldsymbol{\mu}_m}(\sum_{t=1}^{T_s} \gamma_m(\boldsymbol{x}_t^s)\boldsymbol{x}_t^s + \gamma \boldsymbol{\mu}_m)^{\mathrm{T}}}{(n_m + \gamma)^2}$$
$$= \frac{\sum_{t=1}^{T_s} \frac{\partial \gamma_m(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m}(\boldsymbol{x}_t^s)^{\mathrm{T}} + \gamma \boldsymbol{I}}{n_m + \gamma} - \frac{\partial n_m}{\partial \boldsymbol{\mu}_m} \frac{(\boldsymbol{\mu}_m^s)^{\mathrm{T}}}{n_m + \gamma}$$
$$= \frac{\sum_{t=1}^{T_s} \frac{\partial \gamma_m(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m}(\boldsymbol{x}_t^s - \boldsymbol{\mu}_m^s)^{\mathrm{T}} + \gamma \boldsymbol{I}}{n_m + \gamma} \tag{27}$$

where T_s is the number of frames of enrollment segment $\boldsymbol{x}^s$ and

$$\frac{\partial \gamma_m(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m} = 2(\gamma_m(\boldsymbol{x}_t^s) - \gamma_m^2(\boldsymbol{x}_t^s))\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_t^s - \boldsymbol{\mu}_m) \tag{28}$$

When $m' \neq m$

$$\frac{\partial (\boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}}{\partial \boldsymbol{\mu}_m} = \frac{(\sum_{t=1}^{T_s} \frac{\partial \gamma_{m'}(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m}(\boldsymbol{x}_t^s)^{\mathrm{T}})(n_{m'} + \gamma) - \frac{\partial n_{m'}}{\partial \boldsymbol{\mu}_m}(\sum_{t=1}^{T_s} \gamma_{m'}(\boldsymbol{x}_t^s)\boldsymbol{x}_t^s + \gamma \boldsymbol{\mu}_{m'})^{\mathrm{T}}}{(n_{m'} + \gamma)^2}$$
$$= \frac{\sum_{t=1}^{T_s} \frac{\partial \gamma_{m'}(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m}(\boldsymbol{x}_t^s)^{\mathrm{T}}}{n_{m'} + \gamma} - \frac{\partial n_{m'}}{\partial \boldsymbol{\mu}_m} \frac{(\boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}}{n_{m'} + \gamma}$$
$$= \frac{\sum_{t=1}^{T_s} \frac{\partial \gamma_{m'}(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m}(\boldsymbol{x}_t^s - \boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}}{n_{m'} + \gamma} \tag{29}$$

where

$$\frac{\partial \gamma_{m'}(\boldsymbol{x}_t^s)}{\partial \boldsymbol{\mu}_m} = 2\gamma_{m'}(\boldsymbol{x}_t^s)\gamma_m(\boldsymbol{x}_t^s)\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_t^s - \boldsymbol{\mu}_m) \tag{30}$$

Until now, we have obtained all the gradients through manual derivation. The computation of these gradients are not easy to implement, so we only consider the diagonal elements of $\partial(\boldsymbol{\mu}_{m'}^s)^{\mathrm{T}}/\partial\boldsymbol{\mu}_m$. We define

$$\boldsymbol{D} = \mathrm{diag}\left\{\frac{\sum_{t=1}^{T_s} 2(\gamma_m(\boldsymbol{x}_t^s) - \gamma_m^2(\boldsymbol{x}_t^s))\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_t^s - \boldsymbol{\mu}_m)(\boldsymbol{x}_t^s - \boldsymbol{\mu}_m^s)^{\mathrm{T}} + \gamma \boldsymbol{I}}{n_m + \gamma}\right\} \tag{31}$$

Using this diagonal matrix, (21) will become

$$\frac{\partial s(i,\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} = -\frac{2}{T_r}\left\{\sum_{t=1}^{T_r}\left[\gamma_m^s(\boldsymbol{x}_t^r)\boldsymbol{D}(\boldsymbol{\Sigma}_m^s)^{-1}(\boldsymbol{x}_t^r - \boldsymbol{\mu}_m^s) - \gamma_m(\boldsymbol{x}_t^r)\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}_t^r - \boldsymbol{\mu}_m)\right]\right\} \tag{32}$$

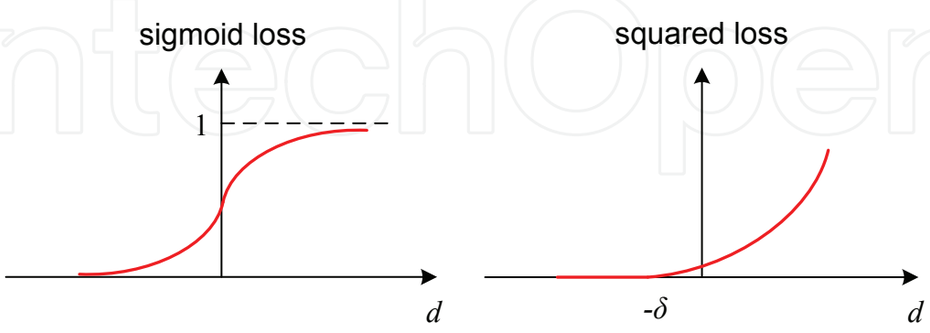
Substitute (32) to (20), we can get the simplified version of gradients.

4. Squared loss function

In the gradient-type descent algorithms, the loss function decrease as the false verification function decreases. However, if the loss function is defined improperly, the verification performance will not be improved through discriminative training.
Besides the sigmoid loss function, the squared loss function (Chao et al., 2008) is also used. It can be expressed as

$$l(i,\boldsymbol{\lambda}) = \begin{cases} \mathrm{cost}(i)\alpha(d(i,\boldsymbol{\lambda}) + \delta)^2 & \text{if } d(i,\boldsymbol{\lambda}) \geq \delta \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

where α and δ are control parameters. Unlike sigmoid function, the squared loss function has greater gradient for large d, which gives more penalty for the severe false verification segments. To give an intuitive illustration, we borrow a figure from (Chao et al., 2009) and show it in Fig. 3.



(a) sigmoid loss function (b) squared loss function

Fig. 3. Comparison of sigmoid loss function and squared loss function (Chao et al., 2009)

Using this squared loss function, the gradient of the objective function w.r.t the mean vector will be

$$\frac{\partial L(\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} = \sum_{i=1}^{I} 2\alpha \text{cost}(i)(d(i,\boldsymbol{\lambda}) + \delta)\frac{\partial s(i,\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m}\text{sign}(i) \tag{34}$$

Other derivations are the same as that in Section 3.

5. Approximate conjugate gradient algorithm

To decrease the object function, gradient descent algorithms in Section 3. In fact, in optimization, other methods, such as conjugate gradient algorithm, are usually used. The gradient descent algorithm is simple to implement, since it only requires first-order derivatives. But its convergent rate is slow. In contrast, the conjugate gradient algorithm has good convergent property, but unfortunately it requires second-order derivatives. In Section 3, we can see that the first-order derivatives are very difficult to deal with, not to mention the second-order derivatives. Herein, we introduce another optimization method, namely, approximate conjugate gradient algorithm (Dixon, 1972), which only needs the first-order derivatives but with fast convergent rate. For convenient expressing, we first define

$$\boldsymbol{g} = \frac{\partial L(\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_m} \tag{35}$$

By using the approximate conjugate gradient algorithm, the update formula will be

$$\boldsymbol{\mu}_m(n+1) = \boldsymbol{\mu}_m(n) - \varepsilon_n \boldsymbol{p}_n \tag{36}$$

where ε_n is the step factor and $\boldsymbol{p}_n$ can be viewed as modified gradient, which can be expressed as

$$\boldsymbol{p}_n = \boldsymbol{g}_n - \beta \boldsymbol{p}_{n-1} \tag{37}$$

where

$$\beta = \frac{\boldsymbol{g}_n^{\mathrm{T}}(\boldsymbol{g}_n - \boldsymbol{g}_{n-1})}{\|\boldsymbol{g}_n\|_2^2} \tag{38}$$

and $\|\cdot\|_2^2$ is the squared 2-norm.

6. Experimental results

6.1 Experimental setup

In this section, the experiments are carried out on NIST speaker recognition evaluation corpora (NIST, 2010). The UBM training (i.e., ML traing) data are selected from SRE04 1-side training set. The discriminative UBM training data come from SRE05 core test condition (1conv4w-1conv4w) dataset. The test data come from SRE06 core test condition (1conv4w-1conv4w) dataset. The numbers of trials of SRE05 and SRE06 are summarized in Table 1.

For the frontend, speech/silence segmentation is performed by a G.723.1 VAD detector. 12 MFCC coefficients plus C0 are computed using 20 ms window and 10ms shift. Cepstral mean subtraction and feature warping (Pelecanos & Sridharan, 2001) with a 3 s window are

Dataset	Target trial	Non-target trial
SRE05 female	1540	16238
SRE05 male	1226	12398
SRE06 female	2712	27913
SRE06 male	2061	21211

Table 1. NIST SRE05 and SRE06 1conv4w-1conv4w trial summary

applied for channel mismatch compensation. Delta, acceleration and triple-delta coefficients are appended to each feature vector, which results in a dimensionality of 52. After that, 25% of low energy frames are discarded using a dynamic threshold. Then, HLDA is employed to decorrelate features and reduce the dimensionality from 52 to 39. Finally, a feature domain latent factor analysis (fLFA) (Vair et al., 2006) is applied to compensate the channel distortion. The performance measures are the same as NIST speaker recognition evaluation (NIST, 2010), using equal error rate (EER) and minimum detection cost function (DCF). DCF is defined as

$$\mathrm{DCF} = 0.1P_{\mathrm{miss}} + 0.99P_{\mathrm{fa}} \tag{39}$$

where P_{miss} is the miss probability and P_{fa} is false alarm probability. We vary the decision threshold, the EER is achieved when P_{miss} is equal to P_{fa}; the min DCF is achieved when DCF get its minimum.

6.2 Baseline performance

A GMM-UBM system has been built as baseline for contrastive analysis. The gender-dependant UBMs with 256 mixtures are trained. No score normalization technology is used.

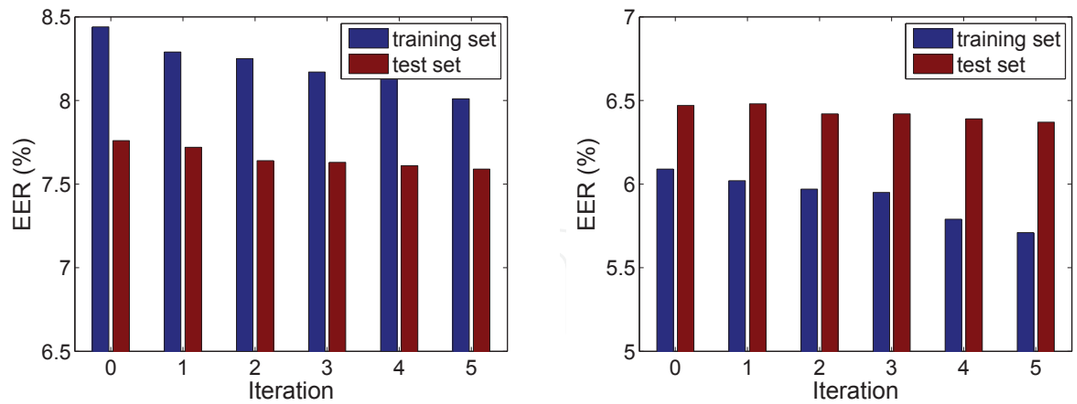
The performance of GMM-UBM system on SRE06 dataset is listed in Table 2. The EERs for female is 7.76% and for male is 6.47%. For 256-mixture GMM-UBM system, this is a quite good baseline.

Gender	EER (%)	min DCF ($\times$100)
female	7.76	3.63
male	6.47	2.90

Table 2. Performance of baseline GMM-UBM system

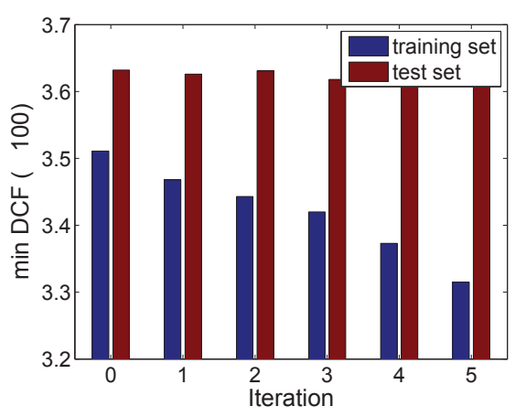
6.3 Sigmoid loss function

In this section, discriminative UBM training with sigmoid loss function is tested. We use SRE05 as training set and SRE06 as test set. The performance on training set and test set are both given in Fig. 4, and the results on test set are listed in Table 3. We can see that after discriminative UBM training, the EERs and min DCFs for female and male are all decreased slightly. This shows that the discriminative UBM training is better than the generative training.

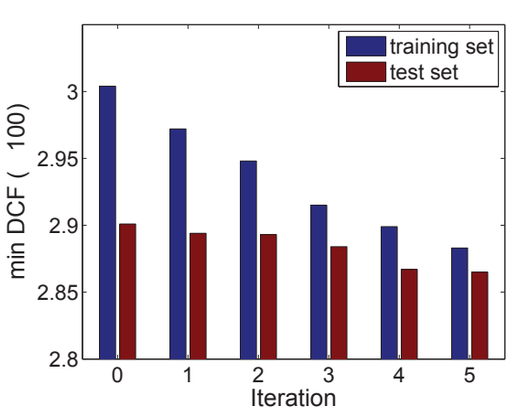


(a) Female EER (%)

(b) Male EER (%)



(c) Female min DCF ($\times$ 100)



(d) Male min DCF ($\times$ 100)

Fig. 4. Performance of discriminative UBM training with sigmoid loss function

Gender	EER (%)	min DCF ($\times$100)
female	7.59	3.61
male	6.37	2.87

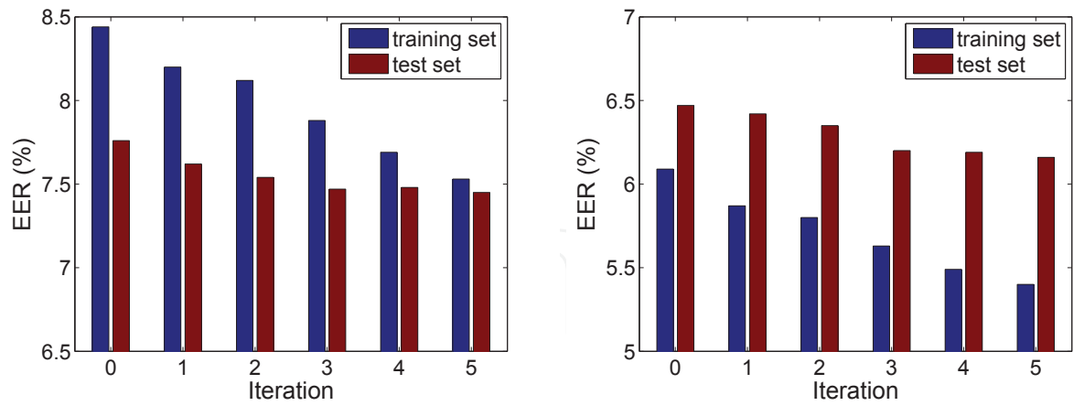
Table 3. Performance of discriminative UBM training with sigmoid loss function

6.4 Squared loss function

In this section, we change the sigmoid loss function to squared loss function. The performance on training set and test set are both given in Fig. 5, and the results on test set are listed in Table 4. Compared these results with that in Section 6.3, it can be observed that the squared loss function is better than the sigmoid loss function. This is due to the more penalty for the falser verification segments.

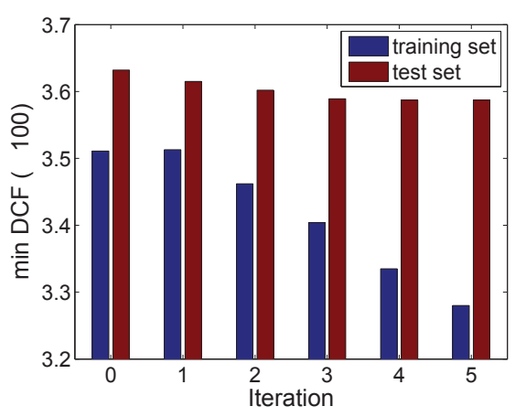
6.5 Approximate conjugate gradient algorithm

In this section, we change the gradient descent algorithm to approximate conjugate gradient algorithm. The EERs and min DCFs are showed in Fig. 6 and Table 5. We can see that

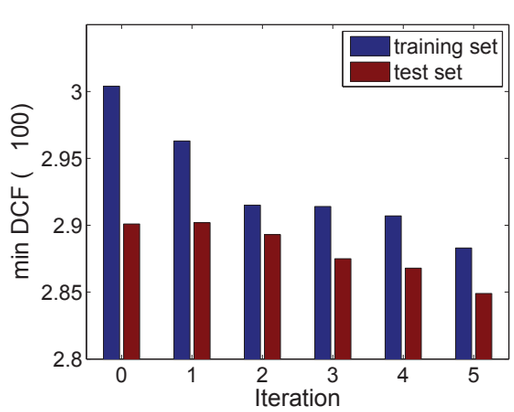


(a) Female EER (%)

(b) Male EER (%)



(c) Female min DCF ($\times$ 100)



(d) Male min DCF ($\times$ 100)

Fig. 5. Performance of discriminative UBM training with squared loss function

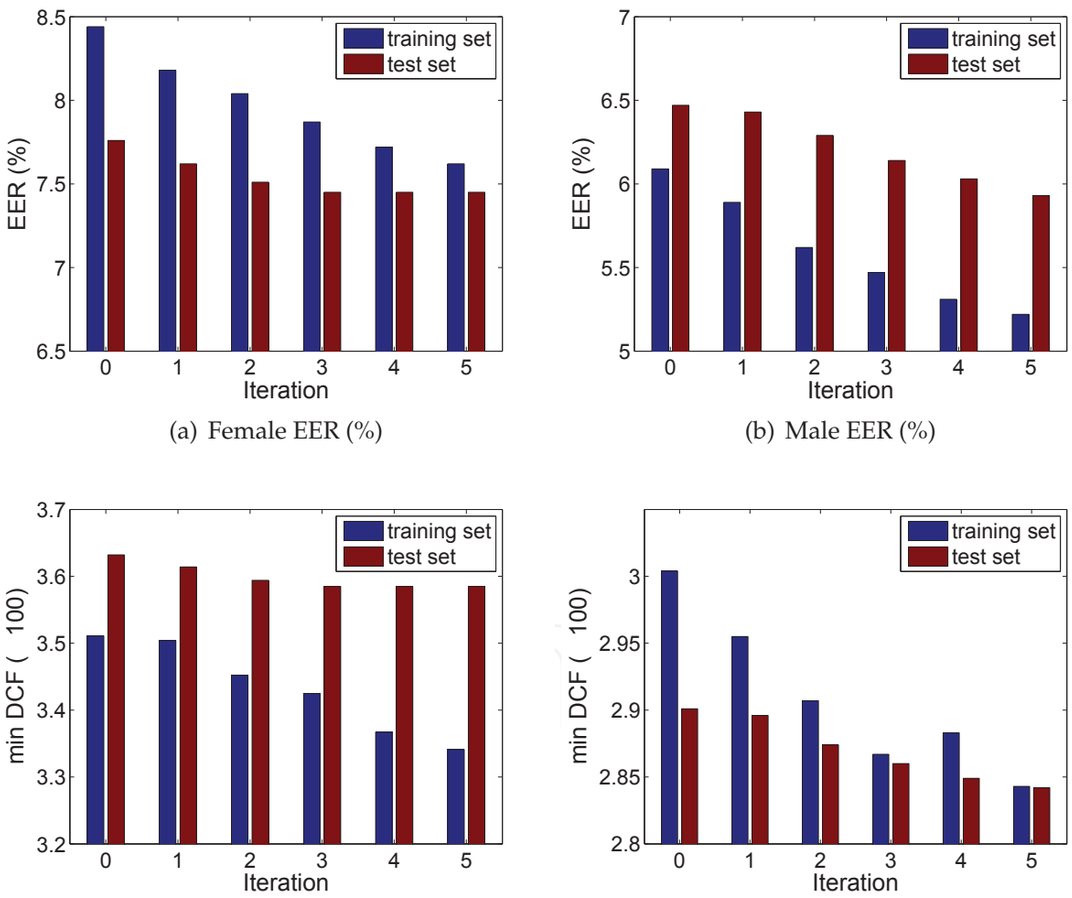
Gender	EER (%)	min DCF ($\times$100)
female	7.45	3.59
male	6.16	2.85

Table 4. Performance of discriminative UBM training with squared loss function

the last female performance of approximate conjugate gradient algorithm is similar to that of gradient descent algorithm, but with faster convergence speed. For the male gender, the approximate conjugate gradient algorithm is better than the gradient descent algorithm. This shows the effectiveness of the approximate conjugate gradient algorithm. At last, we compare the detection error tradeoff (DET) curves (Martin et al., 1997) of the the baseline system and discriminative UBM training with approximate conjugate gradient algorithm in Fig. 7. In the figures, The circles denote the min DCF operating points. From the DET curves, we can see that our proposed discriminative UBM training method achieves slightly better performance.

Gender	EER (%)	min DCF (×100)
female	7.45	3.59
male	5.93	2.84

Table 5. Performance of discriminative UBM training with approximate conjugate gradient algorithm



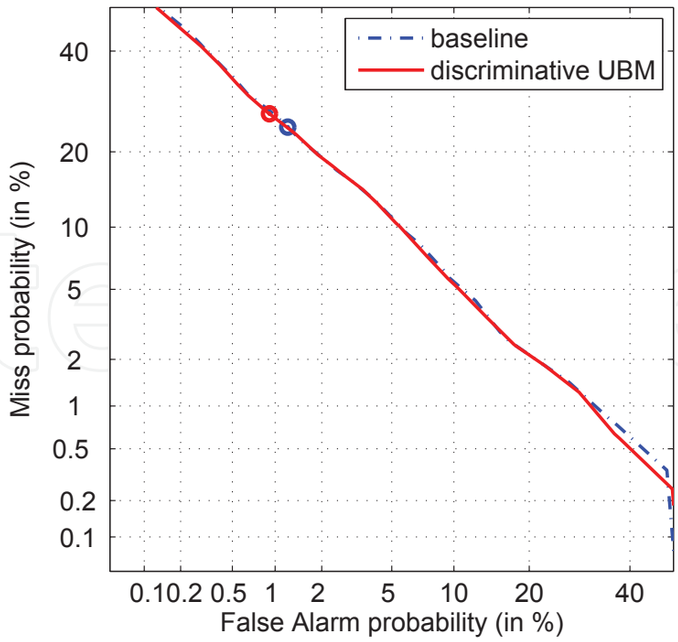
(a) Female EER (%)

(b) Male EER (%)

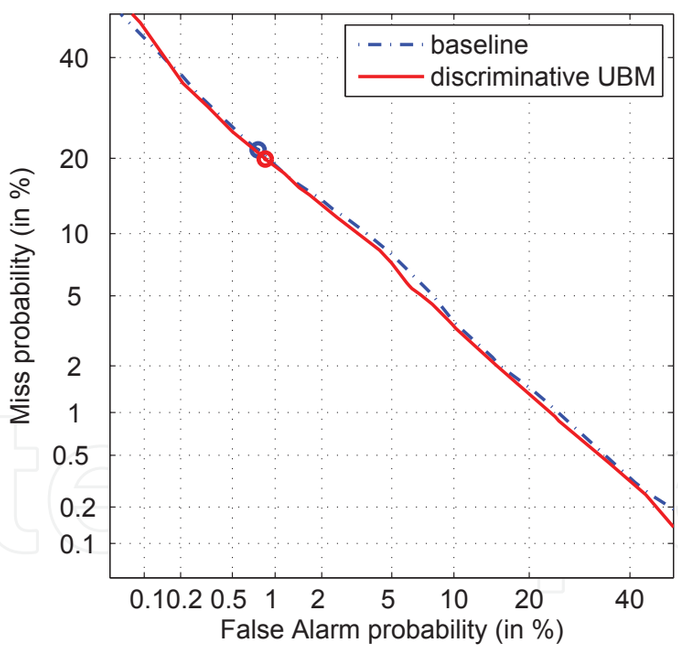
(c) Female min DCF (× 100)

(d) Male min DCF (× 100)

Fig. 6. Performance of discriminative UBM training with approximate conjugate gradient algorithm



(a) Female



(b) Male

Fig. 7. The DET curves of baseline system and discriminative UBM system

7. Conclusion

In this chapter, we present a discriminative UBM training method for speaker recognition. We build the discriminative framework and derive the update formula under minimum

verification error criterion. In this framework, we compare the sigmoid loss function and squared loss function, the gradient descent algorithm and the approximate conjugate gradient algorithm. The experimental results show that the our proposed discriminative UBM training method is better than the prevalent ML training method.

8. Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 61005019, No. 90920302 and No. 60931160443, and in part by the National High Technology Development Program of China under Grant No. 2008AA040201.

9. References

Angkititrakul, P. & Hansen, J. H. L. (2007). Discriminative in-set/out-of-set speaker recognition, *IEEE Transactions on Audio, Speech and Language Processing* 15(2): 498 – 508.

Bimbot, F., Bonastre, J.-F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-Garcia, J., Petrovska-Delacretaz, D. & Reynolds, D. A. (2004). A tutorial on text-independent speaker verification, *EURASIP Journal on Applied Signal Processing* 2004(4): 430–451.

Burget, L., Matejka, P. & Cernocky, J. (2006). Discriminative training techniques for acoustic language identification, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, Toulouse, pp. 209–212.

Campbell Jr., J. P. (1997). Speaker recognition: A tutorial, *Proceedings of the IEEE* 85(9): 1437–1462.

Campbell, W., Sturim, D. & Reynolds, D. (2006). Support vector machines using GMM supervectors for speaker verification, *IEEE Signal Processing Letters* 13(5): 308 – 311.

Chao, Y.-H., Tsai, W.-H. & Wang, H.-M. (2008). Discriminative feedback adaptation for GMM-UBM speaker verification, *Proc. International Symposium on Chinese Spoken Language Processing*, Kunming, pp. 169–172.

Chao, Y.-H., Tsai, W.-H. & Wang, H.-M. (2009). Improving GMM-UBM speaker verification using discriminative feedback adaptation, *Computer Speech and Language* 23(3): 376–388.

Cole, R. A., Mariani, J., Uszkoreit, H. et al. (1997). *Survey of the State of the Art in Human Language Technology*, The Press Syndicate of the University of Cambridge, New York.

Dixon, L. C. W. (1972). *Nonlinear Optimisation*, The English University Press, London.

Gauvain, J.-L. & Lee, C.-H. (1994). Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains, *IEEE Transactions on Speech and Audio Processing* 2(2): 291 – 298.

Hasan, T., Lei, Y., Chandrasekaran, A. & Hansen, J. H. L. (2010). A novel feature sub-sampling method for efficient universal background model training in speaker verification, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, pp. 4494 – 4497.

Huang, C.-L. & Li, H. (2010). UBM data selection for effective speaker modeling, *Proc. International Symposium on Chinese Spoken Language Processing*, Taiwan, pp. 162–165.

Huang, X.-D., Acero, A. & Hon, H.-W. (2000). *Spoken Language Processing*, Prentice Hall, New Jersey.

Juang, B.-H. & Katagiri, S. (1992). Discriminative learning for minimum error classification, *IEEE Transactions on Signal Processing* 40(12): 3043 – 3054.

Kenny, P., Boulianne, G., Ouellet, P. & Dumouchel, P. (2007). Joint factor analysis versus eigenchannels in speaker recognition, *IEEE Transactions on Audio, Speech and Language Processing* 15(4): 1435 – 1447.

Kinnunen, T. & Li, H. (2010). An overview of text-independent speaker recognition: From features to supervectors, *Speech Communication* 52(1): 12 – 40.

Korkmazskiy, F. & Juang, B.-H. (1996). Discriminative adaptation for speaker verification, *Proc. International Conference on Spoken Language Processing*, Vol. 3, Philadelphia, pp. 1744–1747.

Longworth, C. & Gales, M. (2006). Discriminative adaptation for speaker verification, *Proc. InterSpeech 2006 and 9th International Conference on Spoken Language Processing*, Vol. 3, Pittsburgh, pp. 1467–1470.

Ma, C. & Chang, E. (2003). Comparison of discriminative training methods for speaker verification, *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing*, Vol. 1, Hong Kong, pp. 192–195.

Martin, A., Doddington, G., Kamm, T., Ordowski, M. & Przybocki, M. (1997). The DET curve in assessment of detection task performance, *Proc. European Conference on Speech Communication and Technology*, Rhodes, pp. 1895–1898.

NIST (2010). NIST Speaker Recognition Evaluation, [Online], Available: http://www.itl.nist.gov/iad/mig/tests/sre.

Pelecanos, J. & Sridharan, S. (2001). Feature warping for robust speaker verification, *Proc. IEEE Odyssey - The Speaker and Language Recognition Workshop*, Crete, pp. 213–218.

Povey, D. & Kingsbury, B. (2007). Evaluation of proposed modifications to MPE for large scale discriminative training, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4, Honolulu, pp. 321–324.

Reynolds, D. A. (2002). An overview of automatic speaker recognition technology, *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing*, Vol. 4, Orlando, pp. 4072–407.

Reynolds, D. A., Quatieri, T. F. & Dunn, R. B. (2000). Speaker verification using adapted gaussian mixture models, *Digital Signal Processing* 10(1-3): 19–41.

Rosenberg, A. E., Siohan, O. & Parthasarathy, S. (1998). Speaker verification using minimum verification error training, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, Seattler, pp. 105–108.

Vair, C., Colibro, D., Castaldo, F. et al. (2006). Channel factors compensation in model and feature domain for speaker recognition, *Proc. IEEE Odyssey - The Speaker and Language Recognition Workshop*, San Juan.

Woodland, P. C. & Povey, D. (2002). Large scale discriminative training of hidden Markov models for speech recognition, *Computer Speech and Language* 16(1): 25–47.

Zhang, W.-Q., Shan, Y. & Liu, J. (2010). Multiple background models for speaker verification, *Proc. Odyssey - The Speaker and Language Recognition Workshop*, Brno, pp. 47–51.

Zhao, X., Dong, Y., Luo, J., Yang, H. & Wang, H. (2006). Multigrained model adaptation with MAP and reference speaker weighting for text independent speaker verification, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, Toulouse, pp. 913–916.

Speech and Language Technologies
Edited by Prof. Ivo Ipsic

ISBN 978-953-307-322-4
Hard cover, 344 pages
Publisher InTech
Published online 21, June, 2011
Published in print edition June, 2011

This book addresses state-of-the-art systems and achievements in various topics in the research field of speech and language technologies. Book chapters are organized in different sections covering diverse problems, which have to be solved in speech recognition and language understanding systems. In the first section machine translation systems based on large parallel corpora using rule-based and statistical-based translation methods are presented. The third chapter presents work on real time two way speech-to-speech translation systems. In the second section two papers explore the use of speech technologies in language learning. The third section presents a work on language modeling used for speech recognition. The chapters in section Text-to-speech systems and emotional speech describe corpus-based speech synthesis and highlight the importance of speech prosody in speech recognition. In the fifth section the problem of speaker diarization is addressed. The last section presents various topics in speech technology applications like audio-visual speech recognition and lip reading systems.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Wei-Qiang Zhang and Jia Liu (2011). Discriminative Universal Background Model Training for Speaker Recognition, Speech and Language Technologies, Prof. Ivo Ipsic (Ed.), ISBN: 978-953-307-322-4, InTech, Available from: http://www.intechopen.com/books/speech-and-language-technologies/discriminative-universal-background-model-training-for-speaker-recognition

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen