國立交通大學

資訊科學與工程研究所

博士論文

鑑別式訓練法於語者驗證之研究

Discriminative Training Methods for Speaker Verification

研 究 生：趙怡翔

指導教授：王新民　教授

張瑞川　教授

中華民國九十八年一月

鑑別式訓練法於語者驗證之研究
Discriminative Training Methods for Speaker Verification

研 究 生：趙怡翔　　Student：Yi-Hsiang Chao

指導教授：王新民 博士　　Advisor：Dr. Hsin-Min Wang

張瑞川 博士　　Dr. Ruei-Chuan Chang

國立交通大學

資訊科學與工程研究所

博士論文

A Dissertation

Submitted to Department of Computer Science

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

January 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年一月

鑑別式訓練法於語者驗證之研究

學生：趙怡翔

指導教授：王新民 博士
張瑞川 博士

國立交通大學資訊科學與工程研究所

摘要

語者驗證(speaker verification)常被表示成統計上的假說測定(hypothesis testing)問題，用似然比例 (likelihood ratio, LR)檢定的方法來解。一個語者驗證系統性能的好壞高度依賴於目標語者聲音的模型化(空假說)與非目標語者聲音的描述(替代假說)。然而，替代假說因為包含未知的冒充者，通常很難被事先描述地好。在這篇論文，我們提出一個描述替代假說的較佳架構，其目標是希望將目標語者與冒充者做最佳化的鑑別。該架構是建構在一群事先訓練好的背景語者的可用資訊的加權算術組合(weighted arithmetic combination, WAC)或加權幾何組合(weighted geometric combination, WGC)上。我們提出使用二種鑑別式訓練法來最佳化 WAC 或 WGC 的相關參數，分別是最小驗證誤差(minimum verification error, MVE)訓練法與演化式最小驗證誤差(evolutionary minimum verification error, EMVE)訓練法，希望使得錯誤接受(false acceptance)機率與錯誤拒絕(false rejection)機率都能最小。此外，我們也提出二種基於 WAC 與 WGC 的新的決策函數(decision functions)，其可以被視為非線性鑑別分類器(nonlinear discriminant classifiers)。為了求解加權向量 **w**，我們提出使用二種基於核心的鑑別技術(kernel-based discriminant techniques)，分別是基於核心的費氏鑑別器(Kernel Fisher Discriminant, KFD)與支持向量機器(Support Vector Machine, SVM)，因為它們擁有能將目標語者與非目標語者的樣本(samples)有效分開的能力。

在內文不相依(text-independent)語者驗證技術中，GMM-UBM 系統是最常被使用的主流方法。其優點是目標語者模型與通用背景模型(universal background model, UBM) 都具有概括性(generalization)的能力。然而，因為這二種模型是分別根據不同的訓練準則所求出，訓練過程皆沒有考慮到目標語者模型與 UBM 之間的鑑別性(discriminability)。為了改進 GMM-UBM 方法，我們提出一個鑑別式反饋調適(discriminative feedback adaptation, DFA)架構，希望可以同時兼顧概括性與鑑別性。此架構不但保留了原本 GMM-UBM 方法的概括性能力，而且再強化了目標語者模型與 UBM 之間的鑑別性能力。在 DFA 架構下，我們不是使用一個統一的通用背景模型，而是建構一個具鑑別性的特定目標語者反模型(anti-model)。

在我們的實驗中，我們共使用 XM2VTSDB、ISCSLP2006-SRE 與 NIST2001-SRE 這三套語者驗證資料庫(database)，實驗結果顯示我們所提出的方法優於所有傳統上基於 LR 的語者驗證技術。

Discriminative Training Methods for Speaker Verification

Student：Yi-Hsiang Chao

Advisors：Dr. Hsin-Min Wang
Dr. Ruei-Chuan Chang

Department of Computer Science
National Chiao Tung University

ABSTRACT

Speaker verification is usually formulated as a statistical hypothesis testing problem and solved by a likelihood ratio (LR) test. A speaker verification system's performance is highly dependent on modeling the target speaker's voice (the null hypothesis) and characterizing non-target speakers' voices (the alternative hypothesis). However, since the alternative hypothesis involves unknown impostors, it is usually difficult to characterize a priori. In this dissertation, we propose a framework to better characterize the alternative hypothesis with the goal of optimally distinguishing the target speaker from impostors. The proposed framework is built on a weighted arithmetic combination (WAC) or a weighted geometric combination (WGC) of useful information extracted from a set of pre-trained background models. The parameters associated with WAC or WGC are then optimized using two discriminative training methods, namely the minimum verification error (MVE) training method and the proposed evolutionary MVE (EMVE) training method, such that both the false acceptance probability and the false rejection probability are minimized. Moreover, we also propose two new decision functions based on WGC and WAC, which can be regarded as nonlinear discriminant classifiers. To solve the weight vector **w**, we propose using two kernel-based discriminant techniques, namely the Kernel Fisher Discriminant (KFD) and Support Vector Machine (SVM), because of their ability to separate samples of target speakers from those of non-target speakers efficiently.

In recent years, the GMM-UBM system is the predominant approach for the text-independent speaker verification task. The advantage of the approach is that both the target speaker model and the impostor model (UBM) have generalization ability. However, since both models are trained according to separate criteria, the optimization procedure can not distinguish a target speaker from background speakers optimally. To improve the GMM-UBM approach, we propose a discriminative feedback adaptation (DFA) framework that allows generalization and discrimination to be considered jointly. The framework not only preserves the generalization ability of the GMM-UBM approach, but also reinforces the discriminability between the target speaker model and the UBM. Under DFA, rather than use a unified UBM, we construct a discriminative anti-model exclusively for each target speaker.

The results of speaker-verification experiments conducted on three speech corpora, the Extended M2VTS Database (XM2VTSDB), the ISCSLP2006-SRE database and the NIST2001-SRE database, show that the proposed methods outperform all of the conventional LR-based approaches.

ACKNOWLEDGEMENTS

First, I am grateful to my advisor, Dr. Hsin-Min Wang, for his intensive suggestions, patient guidance, and enthusiasm of research. Second, I am grateful to my another advisor, Dr. Ruei-Chuan Chang, for his kindly helping and encouragement in my research. Third, I am grateful to Dr. Wei-Ho Tsai for his helpful suggestions and assistance on my work. Thanks are also given to all members of the Spoken Language Group, Chinese Information Processing Laboratory at IIS, Academia Sinica, for their discussions. Finally, I would like to express my appreciation to my family for their supporting and encouragement. I dedicate this dissertation to my parents.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

In many practical pattern recognition applications, it is necessary to make a binary decision, such as “yes/no” or “accept/reject”, with respect to an uncertain hypothesis that can only be validated through its observable consequences. In a statistical framework, the problem is generally formulated as a test that involves a null hypothesis, H_0, and an alternative hypothesis, H_1, regarding some decision function $L(\cdot)$ for a given observation X:

$$\begin{aligned} H_0 &: L(X) \geq \theta \\ H_1 &: L(X) < \theta, \end{aligned} \tag{1.1}$$

where θ is the decision threshold. Depending on the application, various decision functions can be designed. The most popular decision function computes the ratio of possibilities between the null hypothesis and the alternative hypothesis as follows:

$$L(X) = \frac{L_{H_0}(X)}{L_{H_1}(X)} \begin{cases} \geq \theta & \text{accept } H_0 \\ < \theta & \text{accept } H_1 \text{ (i.e., reject } H_0\text{)}, \end{cases} \tag{1.2}$$

where $L_{H_i}(X)$, $i = 0,1$, denotes a certain possibility measure of X with respect to the hypothesis H_i. For example, $L_{H_i}(X)$ could be the likelihood probability $p(X \mid H_i)$ that hypothesis H_i gives X, and the resulting $L(\cdot)$ represents a so-called likelihood ratio (LR) function. If we represent the observation X as a sequence of r–dimensional feature vectors $\{o_1, \ldots, o_T\}$ and assume that the feature vector sequence of X is independent and identically distributed (i.i.d.), the likelihood of the observation X given the hypothesis H_i, $i = 0$ or 1, can be computed by

$$p(X \mid H_i) = \prod_{t=1}^{T} p(o_t \mid H_i). \tag{1.3}$$

In the implementation, H_0 and H_1 can be characterized by some parametric models, which are usually denoted as λ (the null hypothesis model or target model) and $\bar{\lambda}$ (the alternative hypothesis model or anti-model). Suppose both λ and $\bar{\lambda}$ are characterized by Gaussian mixture models (GMMs) [Reynolds 1995, 2000], the probability density functions (pdf) of each feature vector o_t given H_0 and H_1 can be respectively defined as

$$p(o_t \mid H_0) = p(o_t \mid \lambda) = \sum_{m=1}^{M_0} p_m^0 p(o_t \mid \mathbf{g}_m^0) \tag{1.4}$$

and

$$p(o_t \mid H_1) = p(o_t \mid \bar{\lambda}) = \sum_{m=1}^{M_1} p_m^1 p(o_t \mid \mathbf{g}_m^1), \tag{1.5}$$

where p_m^i, i = 0 or 1, $m = 1,\ldots,M_i$, is the mixture weight that satisfies the constraint $\sum_{m=1}^{M_i} p_m^i = 1$; $\mathbf{g}_m^i \sim N(\boldsymbol{\mu}_m^i, \boldsymbol{\Sigma}_m^i)$ is the m-th Gaussian mixture component of the target model λ (i = 0) or the alternative hypothesis model $\bar{\lambda}$ (i = 1) with the r×1 mean vector $\boldsymbol{\mu}_m^i$ and the r×r covariance matrix $\boldsymbol{\Sigma}_m^i$; and $p(o_t \mid \mathbf{g}_m^i)$ is the Gaussian density function that is expressed as

$$p(o_t \mid \mathbf{g}_m^i) = \frac{1}{(2\pi)^{r/2} \mid \boldsymbol{\Sigma}_m^i \mid^{1/2}} \exp\left\{ -\frac{1}{2} (o_t - \boldsymbol{\mu}_m^i)' (\boldsymbol{\Sigma}_m^i)^{-1} (o_t - \boldsymbol{\mu}_m^i) \right\}. \tag{1.6}$$

However, in most real applications, the alternative hypothesis model $\bar{\lambda}$ is usually ill-defined and difficult to characterize a priori. For example, in speaker verification [Bimbot

2004; Faundez-Zanuy 2005; Fauve 2007; Przybocki 2007; Van Leeuwen 2006], the problem of determining if a speaker is who he or she claims to be is normally formulated as follows: given an unknown utterance U, determine whether

H_0: U is from the target speaker, or

H_1: U is not from the target speaker.

Though H_0 can be modeled straightforwardly using speech utterances from the target speaker, H_1 does not involve any specific speaker, and hence lacks explicit data for modeling. As a result, various approaches have placed special emphasis on better characterization of H_1. One popular approach pools all the speech data from a large number of background speakers and trains a single speaker-independent GMM Ω, called the world model or the universal background model (UBM) [Reynolds 2000]. During a test, the logarithmic LR measure that an unknown utterance U was spoken by the claimed speaker can be evaluated by

$$L_{\text{UBM}}(U) = \log p(U \mid \lambda) - \log p(U \mid \Omega), \tag{1.7}$$

where λ is the target speaker GMM trained using speech from the claimed speaker. The larger the value of $L_{\text{UBM}}(U)$, the more likely it is that the utterance U was spoken by the claimed speaker. Due to the good generalization ability of the UBM, $L_{\text{UBM}}(U)$ (usually called the GMM-UBM method [Reynolds 2000] is considered as a current state-of-the-art solution to the text-independent speaker verification problem.

Instead of using a single model, an alternative approach is to train a set of GMMs $\{\lambda_1, \lambda_2, \ldots, \lambda_B\}$ using speech from several representative speakers, called a cohort [Rosenberg 1992], which simulates potential impostors. This leads to the following possible logarithmic LR measures, where the alternative hypothesis can be characterized by:

(i) the likelihood of the most competitive cohort model [Liu 1996], i.e.,

$$L_{\text{Max}}(U) = \log p(U \mid \lambda) - \max_{1 \le i \le B} \log p(U \mid \lambda_i), \tag{1.8}$$

(ii) the arithmetic mean of the likelihoods of the B cohort models [Reynolds 1995], i.e.,

$$L_{\text{Ari}}(U) = \log p(U \mid \lambda) - \log\left\{\frac{1}{B}\sum_{i=1}^{B} p(U \mid \lambda_i)\right\}, \tag{1.9}$$

(iii) the geometric mean of the likelihoods of the B cohort models [Liu 1996], i.e.,

$$L_{\text{Geo}}(U) = \log p(U \mid \lambda) - \frac{1}{B}\sum_{i=1}^{B} \log p(U \mid \lambda_i). \tag{1.10}$$

In a well-known score normalization method called *T-norm* [Auckenthaler 2000; Sturim 2005], $L_{\text{Geo}}(U)$ is divided by the standard deviation of the log-likelihoods of the B cohort models.

The LR measures in Eqs. (1.7) – (1.10) can be collectively expressed in the following general form [Reynolds 2000]:

$$L(U) = \frac{p(U \mid \lambda)}{\Psi\big(p(U \mid \lambda_1), p(U \mid \lambda_2),\ldots, p(U \mid \lambda_N)\big)}, \tag{1.11}$$

where $\Psi(\cdot)$ denotes a certain function of the likelihoods computed for a set of so-called background models $\{\lambda_1, \lambda_2,\ldots, \lambda_N\}$. For example, if the background model set is generated from a cohort, letting $\Psi(\cdot)$ be the maximum function gives $L_{\text{Max}}(U)$, while the arithmetic mean gives $L_{\text{Ari}}(U)$, and the geometric mean gives $L_{\text{Geo}}(U)$. When $\Psi(\cdot)$ is an identity function, $N = 1$, and $\lambda_1 = \Omega$, Eq. (1.11) becomes $L_{\text{UBM}}(U)$.

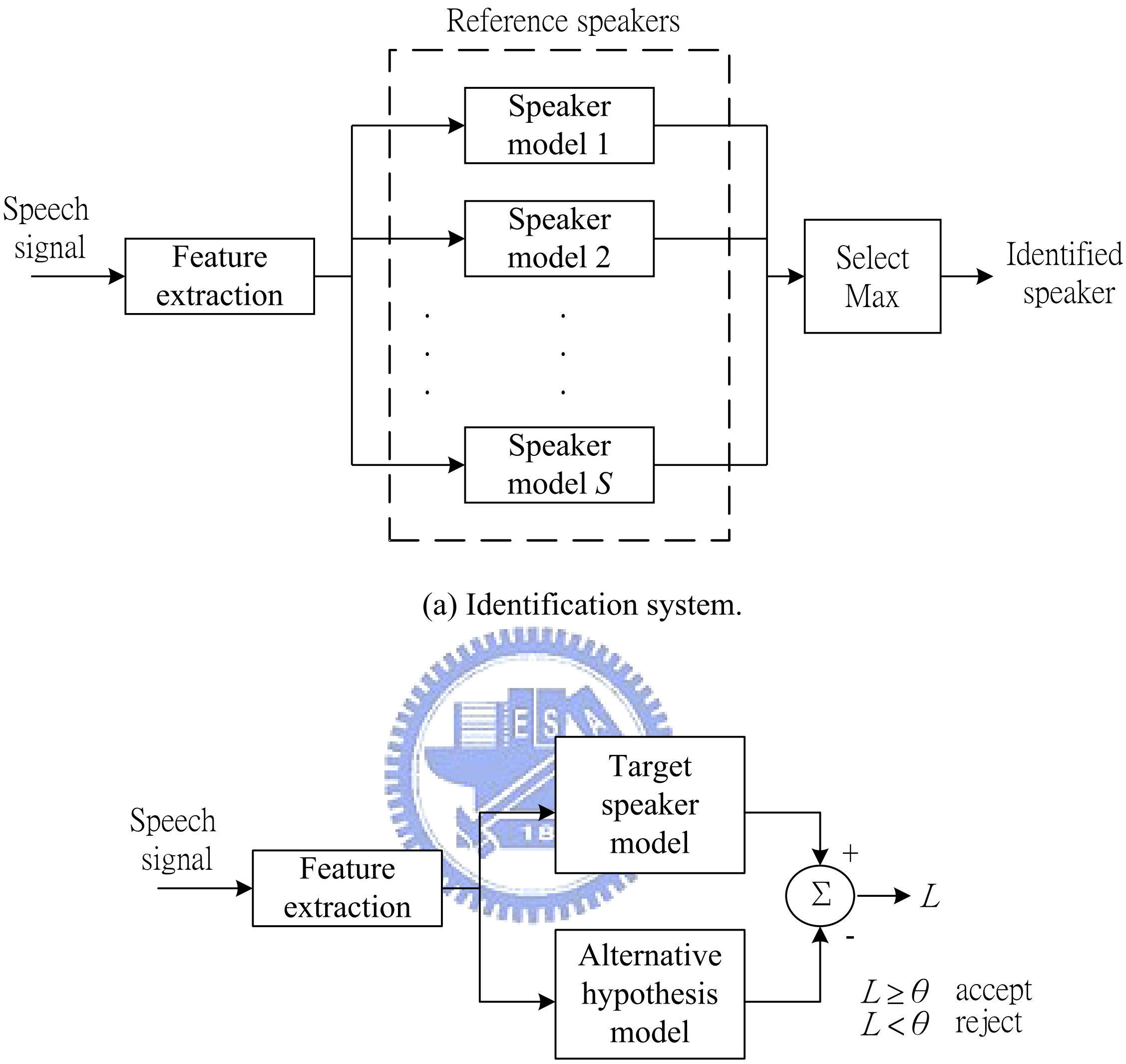
However, there is no theoretical evidence to indicate which method of characterizing H_1 is optimal, and the selection of $\Psi(\cdot)$ is usually application and training data dependent. More specifically, a simple function, such as the arithmetic mean, the maximum, or the geometric mean, is a heuristic that does not involve any optimization process. Thus, the resulting system is far from optimal in terms of verification accuracy. Although the GMM-UBM method is a current state-of-the-art solution to the text-independent speaker verification problem, there is no optimization process of characterizing H_1 to support its discriminability.

Before the presentation of the proposed frameworks for speaker verification problems, we introduce some backgrounds about the current GMM-based speaker recognition methods.

1.1. Background

Over the past several years, GMM has become the dominant modeling approach in speaker recognition applications. Speaker recognition can be classified into identification and verification. In speaker identification, the system has trained models for a certain amount of speakers and the task is to determine which one of these models best matches the current speaker. In verification, the identity of the current speaker is somehow transmitted to the system beforehand and the task is to determine whether the current speaker is the claimed one or not. Speaker recognition methods can also be divided into text-dependent and text-independent methods. The former requires the speaker to say keywords or sentences having the same text for both training and recognition trials, while the latter does not rely on a specific text being spoken.

Fig. 1.1 shows the block diagrams of the speaker identification and verification systems. The process of feature extraction is to transform the speech signal into a set of feature vectors, and the goal is to obtain a new representation which is more compact, less redundant, and more suitable for statistical modeling and the calculation of a distance or any other kind of score. In recent years, Mel-scale frequency cepstral coefficient (MFCC) [Huang 2001] is the most popular feature vector used in speech and speaker recognition systems. The mel-scale cepstrum is the discrete cosine transform (DCT) of the log-spectral energies of the speech segment. The spectral energies are calculated over logarithmically spaced filters with increasing bandwidths (mel-filters). MFCC-based GMMs [Reynolds 1995] have been successfully applied to speaker recognition systems recently. In the following, we introduce two commonly-used statistical modeling methods for estimating the parameters of GMMs.



(a) Identification system.

(b) Verification system.

Fig. 1.1. Speaker recognition systems.

1.1.1. Maximum Likelihood (ML) Estimation Technique

Given the training speech data from a speaker, $U = \{o_1, \ldots, o_T\}$, the goal of maximum likelihood (ML) estimation is to find the parameters of the GMM, λ, which maximize the likelihood of the GMM:

$$p(U \mid \lambda) = \prod_{t=1}^{T} p(o_t \mid \lambda). \tag{1.12}$$

Eq. (1.12) is a nonlinear function of the GMM parameters and direct maximization is infeasible. However, the ML parameter estimation can be achieved iteratively via the expectation-maximization (EM) algorithm [Huang 2001].

The basic idea of the EM algorithm is, beginning with an initial model λ, to estimate a new model $\hat{\lambda}$, such that $p(U \mid \hat{\lambda}) \geq p(U \mid \lambda)$. The new model then becomes the initial model for the next iteration and the process is repeated until some convergence condition is reached. This is the same basic technique used for estimating hidden Markov model (HMM) parameters via the Baum-Welch re-estimation algorithm [Huang 2001].

In each EM iteration, the following re-estimation formulae, which guarantee a monotonic increase in the model's likelihood value, are used:

Mixture Weights:

$$\hat{p}_m = \frac{1}{T} \sum_{t=1}^{T} p(m \mid o_t, \lambda). \tag{1.13}$$

Mean vectors:

$$\hat{\boldsymbol{\mu}}_m = \frac{\sum_{t=1}^{T} p(m \mid o_t, \lambda) o_t}{\sum_{t=1}^{T} p(m \mid o_t, \lambda)}. \tag{1.14}$$

Covariance matrices:

$$\hat{\boldsymbol{\Sigma}}_m = \frac{\sum_{t=1}^{T} p(m \mid o_t, \lambda)(o_t - \hat{\boldsymbol{\mu}}_m)(o_t - \hat{\boldsymbol{\mu}}_m)'}{\sum_{t=1}^{T} p(m \mid o_t, \lambda)}. \tag{1.15}$$

We usually assume that all covariance matrices $\boldsymbol{\Sigma}_m$ of the GMM, m = 1,…, M, are diagonal, and the variance vector $\boldsymbol{\sigma}_m^2 = \mathrm{diag}(\boldsymbol{\Sigma}_m)$. The *a posteriori* probability for the *m*-th Gaussian mixture component $\mathbf{g}_m$ is given by

$$p(m \mid o_t, \lambda) = \frac{p_m p(o_t \mid \mathbf{g}_m)}{\sum_{i=1}^{M} p_i p(o_t \mid \mathbf{g}_i)}, \tag{1.16}$$

where p_m is the original mixture weight and $p(o_t \mid \mathbf{g}_m)$ is the Gaussian density function defined in Eq. (1.6).

Selecting the order M of the mixture and initializing the model parameters prior to the EM algorithm are two critical factors in training a GMM. There are no good theoretical means to guide one in either of these selections, so they are best experimentally determined for a given task.

1.1.2. Maximum A Posteriori (MAP) Estimation Technique

Conventional GMMs trained from the EM algorithm perform well only when a large amount of training data is available to characterize the characteristics of the speaker. For each speaker, this approach needs a large amount of training data to train a GMM so as to cover all the possible pronunciations of this speaker, in particular when the speaker recognition is conducted under the text-independent mode. Due to this characteristic, the performance of GMM deteriorates drastically when the training data are sparse. However, client speakers definitely prefer to enroll with as little speech as possible. To solve this problem, speaker adaptation approaches have been investigated in recent years.

One successful adaptation approach, namely the UBM-MAP approach, has been widely used in text-independent speaker verification tasks. This approach first pools all speech data from a large number of background speakers to train a universal background model (UBM) via the EM algorithm. Unlike the standard approach of maximum likelihood training of the speaker model independently of the UBM, this approach then adapts the well-trained UBM to a speaker model λ using this speaker's training speech via the maximum a posteriori (MAP) estimation technique. The adapted GMM λ is effective because its generalization ability allows λ to handle acoustic patterns not covered by the limited training data of the speaker.

The specifics of the adaptation are as follows. Given a UBM, Ω, and training vectors from the target speaker, $U = \{o_1, \ldots, o_T\}$, we first computer the *a posteriori* probability $p(m \mid o_t, \Omega)$ for the *m*-th Gaussian mixture component $\mathbf{g}_m$, *m* = 1,…, *M*, of the UBM:

$$p(m \mid o_t, \Omega) = \frac{p_m p(o_t \mid \mathbf{g}_m)}{\sum_{i=1}^{M} p_i p(o_t \mid \mathbf{g}_i)}, \tag{1.17}$$

Then, we use $p(m \mid o_t, \Omega)$ and o_t to compute the sufficient statistics for the mixture weight, mean, and variance parameters: [‡]

$$n_m = \sum_{t=1}^{T} p(m \mid o_t, \Omega), \tag{1.18}$$

$$E_m(o) = \frac{1}{n_m} \sum_{t=1}^{T} p(m \mid o_t, \Omega) o_t, \tag{1.19}$$

$$E_m(o^2) = \frac{1}{n_m} \sum_{t=1}^{T} p(m \mid o_t, \Omega) o_t^2. \tag{1.20}$$

This is the same as the expectation step in the EM algorithm.

Finally, these new sufficient statistics from the training data are used to update the old UBM sufficient statistics from the *m*-th mixture to create the adapted parameters for the *m*-th mixture with the equations:

$$\hat{p}_m = [\varepsilon_m n_m / T + (1 - \varepsilon_m) p_m] c, \tag{1.21}$$

[‡] x^2 is shorthand for diag (xx').

$$\hat{\boldsymbol{\mu}}_m = \varepsilon_m E_m(o) + (1-\varepsilon_m)\boldsymbol{\mu}_m, \tag{1.22}$$

and

$$\hat{\boldsymbol{\sigma}}_m^2 = \varepsilon_m E_m(o^2) + (1-\varepsilon_m)(\boldsymbol{\sigma}_m^2 + \boldsymbol{\mu}_m^2) - \hat{\boldsymbol{\mu}}_m^2, \tag{1.23}$$

where the scale factor c is computed over all adapted mixture weights to ensure that they sum to 1 and the adaptation coefficients ε_m controlling the balance between old and new estimates is defined as

$$\varepsilon_m = \frac{n_m}{n_m + r}, \tag{1.24}$$

where r is a fixed relevance factor. After adaptation, the mixture components of the adapted GMM retain a correspondence with the mixtures of the UBM.

1.2. The Approaches of This Dissertation

In speaker recognition tasks, as the ML or MAP estimation technique has become the standard modeling method for characterizing the target speaker (the null hypothesis), this dissertation focuses on two issues: the improvement of the characterization of the alternative hypothesis and the improvement of the current state-of-the-art GMM-UBM method.

1.2.1. Using Minimum Verification Error Training

To handle the speaker-verification problem more effectively, it is necessary to design a trainable mechanism for $\Psi(\cdot)$ defined in Eq. (1.11). We therefore propose a framework to better characterize the alternative hypothesis with the goal of optimally distinguishing the target speaker from impostors. The proposed framework is built on a weighted arithmetic combination (WAC) or a weighted geometric combination (WGC) of useful information extracted from a set of pre-trained background models. The parameters associated with WAC or WGC are then optimized using two discriminative training methods, namely, the minimum verification error (MVE) training method [Chou 2003; Rosenberg 1998] and the proposed evolutionary MVE (EMVE) training method, such that both the false acceptance probability and the false rejection probability are minimized. The results of speaker verification experiments conducted on the Extended M2VTS Database (XM2VTSDB) [Messer 1999] demonstrate that the proposed frameworks along with the MVE or EMVE training outperform conventional LR-based approaches.

1.2.2. Using Kernel Discriminant Analysis

In contrast to the MVE training methods with the goal of minimizing both the false acceptance probability and the false rejection probability, we further propose two new decision functions based on WGC and WAC, which can be regarded as nonlinear discriminant classifiers. To obtain a reliable set of weights, the goal here is to separate the target speaker from imposters optimally. Thus, we apply kernel-based techniques, namely the Kernel Fisher Discriminant (KFD) [Mika 1999, 2002] and Support Vector Machine (SVM) [Burges 1998], to solve the weights, by virtue of their good discrimination ability. Our proposed approaches have two advantages over existing methods. The first is that they embed a trainable mechanism in the decision functions. The second is that they convert variable-length utterances into fixed-dimension characteristic vectors, which are easily processed by kernel discriminant analysis. The results of experiments conducted on both the XM2VTSDB and the ISCSLP2006-SRE database show that the proposed kernel-based decision functions outperform all of the conventional approaches.

1.2.3. Using Discriminative Feedback Adaptation

The GMM-UBM system [Reynolds 2000] is the predominant approach for text-independent speaker verification because both the target speaker model and the impostor model (UBM) have generalization ability to handle “unseen” acoustic patterns. However, since GMM-UBM uses a common anti-model, namely UBM, for all target speakers, it tends to be weak in rejecting impostors’ voices that are similar to the target speaker’s voice. To overcome this limitation, we propose a discriminative feedback adaptation (DFA) framework that reinforces the discriminability between the target speaker model and the anti-model, while preserving the generalization ability of the GMM-UBM approach. This is achieved by adapting the UBM to a target speaker dependent anti-model based on a minimum verification squared-error criterion, rather than estimating the model from scratch by applying the conventional discriminative training schemes. The results of experiments conducted on the NIST2001-SRE database show that DFA substantially improves the performance of the conventional GMM-UBM approach.

1.3. The Organization of This Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 and 3 describe, respectively, the MVE training methods and the kernel discriminant analysis techniques used to improve the characterization of the alternative hypothesis. Chapter 4 introduces the proposed DFA framework for improving the GMM-UBM method. Then, in Chapter 5, we present our conclusions.

Chapter 2

Improving the Characterization of the Alternative Hypothesis via Minimum Verification Error Training

To handle the speaker-verification problem more effectively, we propose a framework that characterizes the alternative hypothesis by exploiting information available from background models, such that the utterances of the impostors can be more effectively distinguished from those of the target speaker. The framework is built on either a weighted geometric combination (WGC) or a weighted arithmetic combination (WAC) of the likelihoods computed for background models. In contrast to the geometric mean in $L_{\mathrm{Geo}}(U)$ defined in Eq. (1.6) or the arithmetic mean in $L_{\mathrm{Ari}}(U)$ defined in Eq. (1.5), both of which are independent of the system training, our combination scheme treats the background models unequally according to how close each individual is to the target speaker model, and quantifies the unequal nature of the background models by a set of weights optimized in the training phase. The optimization is carried out with the minimum verification error (MVE) criterion [Chou 2003; Rosenberg 1998], which minimizes both the false acceptance probability and the false rejection probability. Since the characterization of the alternative hypothesis is closely related to the verification accuracy, the resulting system is expected to be more effective and robust than those of conventional methods.

The concept of MVE training stems from minimum classification error (MCE) training [Juang 1997; Siohan 1998; McDermott 2007; Ma 2003], where the former could be a special case of the latter when the classes to be distinguished are binary. Although MVE training has been extensively studied in the literature [Chou 2003; Rosenberg 1998; Sukkar 1996, 1998; Rahim 1997; Kuo 2003; Siu 2006], most studies focus on better estimating the parameters of the target model. In contrast, we try to improve the characterization of the alternative hypothesis by applying MVE training to optimize the parameters associated with the combinations of the likelihoods from a set of background models. Traditionally, MVE training has been realized by the gradient descent algorithms, e.g., the generalized probability descent (GPD) [Chou 2003], but the approach only guarantees to converge to a local optimum. To overcome such a limitation, we propose a new MVE training method, called evolutionary MVE (EMVE) training, for learning the parameters associated with WAC and WGC based on a genetic algorithm (GA) [Eiben 2003]. It has been shown in many applications that GA-based optimization is superior to gradient-based optimization, because of GA's global scope and parallel searching power. To facilitate the EMVE training, we designed a new mutation operator, called the one-step gradient descent operator (GDO), for the genetic algorithm. The results of speaker verification experiments conducted on the Extended M2VTS Database (XM2VTSDB) [Messer 1999] demonstrate that the proposed methods outperform conventional LR-based approaches.

The remainder of this chapter is organized as follows. Section 2.1 presents the proposed methods for characterizing the alternative hypothesis. Sections 2.2 and 2.3 describe, respectively, the gradient-based MVE training and the EMVE training used to optimize our methods. Section 2.4 contains the experiment results.

2.1. Characterization of the Alternative Hypothesis

To characterize the alternative hypothesis, we generate a set of background models using data that does not belong to the target speaker. Instead of using the heuristic arithmetic mean or geometric mean, our goal is to design a function $\Psi(\cdot)$ that optimally exploits the information available from background models. In this section, we present our approach, which is based on either the weighted arithmetic combination (WAC) or the weighted geometric combination (WGC) of the useful information available. Moreover, the LR measure based on WAC or WGC can be viewed as a generalized and trainable version of $L_{\mathrm{UBM}}(U)$ in Eq. (1.3), $L_{\mathrm{Max}}(U)$ in Eq. (1.4), $L_{\mathrm{Ari}}(U)$ in Eq. (1.5), or $L_{\mathrm{Geo}}(U)$ in Eq. (1.6).

2.1.1. The Weighted Arithmetic Combination (WAC)

First, we define the function $\Psi(\cdot)$ in Eq. (1.7) based on the weighted arithmetic combination as

$$p(U \mid \bar{\lambda}) = \Psi(p(U \mid \lambda_1), \ldots, p(U \mid \lambda_N)) = \sum_{i=1}^{N} w_i p(U \mid \lambda_i), \tag{2.1}$$

where w_i is the weight of the likelihood $p(U \mid \lambda_i)$ subject to $\sum_{i=1}^{N} w_i = 1$. This function assigns different weights to N background models to indicate their individual contribution to the alternative hypothesis. Suppose all the N background models are Gaussian Mixture Models (GMMs); then, Eq. (2.1) can be viewed as a mixture of Gaussian mixture density functions. From this perspective, the alternative hypothesis model $\bar{\lambda}$ can be viewed as a GMM with two layers of mixture weights, where one layer represents each background model and the other represents the combination of background models.

2.1.2. The Weighted Geometric Combination (WGC)

Alternatively, we can define the function $\Psi(\cdot)$ in Eq. (1.7) from the perspective of the weighted geometric combination as

$$p(U\,|\,\bar{\lambda}) = \Psi(p(U\,|\,\lambda_1),\ldots,p(U\,|\,\lambda_N)) = \prod_{i=1}^{N} p(U\,|\,\lambda_i)^{w_i}\,. \tag{2.2}$$

Similar to the weighted arithmetic combination, Eq. (2.2) considers the individual contribution of a background model to the alternative hypothesis by assigning a weight to each likelihood value. One additional advantage of WGC is that it avoids the problem where $p(U\,|\,\bar{\lambda}) \to 0$. The problem can arise with the heuristic geometric mean because some values of the likelihood may be rather small when the background models λ_i are irrelevant to an input utterance U, i.e., $p(U|\,\lambda_i) \to 0$. However, if a weight is attached to each background model, $\Psi(\cdot)$ defined in Eq. (2.2) should be less sensitive to a tiny value of the likelihood; hence, it should be more robust and reliable than the heuristic geometric mean.

2.1.3. Relation to Conventional LR Measures

We observe that Eq. (2.1) and Eq. (2.2) are equivalent to the arithmetic mean and the geometric mean, respectively, when $w_i = 1/N$, $i = 1,2,\ldots, N$; in other words, all the background models are assumed to contribute equally. It is also clear that both Eq. (2.1) and Eq. (2.2) will degenerate to a maximum function if we set $w_{i^*} = 1$, where $i^* = \arg\max_{1\le i\le N} p(U\,|\,\lambda_i)$, and $w_i = 0$, $\forall i \ne i^*$. Furthermore, the logarithmic LR measure based on Eq. (2.1) or Eq. (2.2) will degenerate to $L_{\mathrm{UBM}}(U)$ in Eq. (1.3) if only a UBM Ω is used as the background model. Thus, both WAC- and WGC-based logarithmic LR measures can be viewed as generalized and trainable versions of $L_{\mathrm{UBM}}(U)$ in Eq. (1.3), $L_{\mathrm{Max}}(U)$ in Eq. (1.4), $L_{\mathrm{Ari}}(U)$ in Eq. (1.5), or $L_{\mathrm{Geo}}(U)$ in Eq. (1.6).

In the WAC method, we refer to the alternative hypothesis model $\bar{\lambda}$ defined in Eq. (2.1) as a *2-layer GMM* (*GMM2*), since it involves both inner and outer mixture weights. GMM2 differs from the UBM Ω in that it characterizes the relationship between individual background models through the outer mixture weights, rather than simply pooling all the available data and training a single background model represented by a GMM. Note that the inner and outer mixture weights are trained by different algorithms. Specifically, the inner mixture weights are estimated using the standard expectation-maximization (EM) algorithm [Huang 2001], while the outer mixture weights are estimated using minimum verification error (MVE) training or evolutionary MVE (EMVE) training, which we will discuss in Sec. 2.2 and Sec. 2.3, respectively. In other words, GMM2 integrates the Bayesian learning and discriminative training algorithms. The objective is to optimize the LR measure by considering the null hypothesis and the alternative hypothesis jointly.

2.1.4. Background Model Selection

In general, the more speakers that are used as background models, the better the characterization of the alternative hypothesis will be. However, it has been found [Reynolds 1995; Rosenberg 1992; Liu 1996; Higgins 1991; Auckenthaler 2000; Sturim 2005] that using a set of pre-selected representative models usually makes the system more effective and efficient than using the entire collection of available speakers. For this reason, we present two approaches for selecting background models to strengthen our WAC- and WGC-based methods.

A. Combining cohort models and the world model

Our first approach selects B+1 background models, comprised of B cohort models used in $L_{\text{Max}}(U)$, $L_{\text{Ari}}(U)$, and $L_{\text{Geo}}(U)$, and one world model used in $L_{\text{UBM}}(U)$, for WAC in Eq. (2.1) and WGC in Eq. (2.2). Depending on the definition of a cohort, we consider two commonly-used methods [Reynolds 1995]. One selects the B closest speaker models $\{\lambda_{\text{cst }1}, \lambda_{\text{cst }2}, \ldots, \lambda_{\text{cst }B}\}$ for each target speaker; and the other selects the $B/2$ closest speaker models $\{\lambda_{\text{cst }1}, \lambda_{\text{cst }2}, \ldots, \lambda_{\text{cst }B/2}\}$, plus the $B/2$ farthest speaker models $\{\lambda_{\text{fst }1}, \lambda_{\text{fst }2}, \ldots, \lambda_{\text{fst }B/2}\}$, for each target speaker. Here, the degree of closeness is measured in terms of the pairwise distance defined in [Reynolds 1995]:

$$d(\lambda_i,\lambda_j) = \log\frac{p(U_i\,|\,\lambda_i)}{p(U_i\,|\,\lambda_j)} + \log\frac{p(U_j\,|\,\lambda_j)}{p(U_j\,|\,\lambda_i)}, \tag{2.3}$$

where λ_i and λ_j are speaker models trained using the i-th speaker's utterances U_i and the j-th speaker's utterances U_j, respectively. As a result, each target speaker has a sequence of background models, $\{\Omega, \lambda_{\text{cst }1}, \lambda_{\text{cst }2}, \ldots, \lambda_{\text{cst }B}\}$ or $\{\Omega, \lambda_{\text{cst }1}, \ldots, \lambda_{\text{cst }B/2}, \lambda_{\text{fst }1}, \ldots, \lambda_{\text{fst }B/2}\}$, for Eqs. (1.7), (2.1), and (2.2).

B. Combining multiple types of anti-models

As shown in Eqs. (1.3) – (1.6), various types of anti-models have been studied for conventional LR measures. However, none of the LR measures developed thus far has proved to be absolutely superior to any other. Usually, $L_{\text{UBM}}(U)$ tends to be weak in rejecting impostors with voices similar to the target speaker's voice, while $L_{\text{Max}}(U)$ is prone to falsely rejecting a target speaker; $L_{\text{Ari}}(U)$ and $L_{\text{Geo}}(U)$ are between these two extremes. The advantages and disadvantages of different LR measures motivate us to combine them into a unified LR measure because of the complementary information that each anti-model can contribute.

Consider K different LR measures $L_i(U)$, each with an anti-model $\bar{\lambda}_i$, i = 1,2,…, K. If we treat each anti-model $\bar{\lambda}_i$ as a background model, the function $\Psi(\cdot)$ in Eq. (1.7) can be rewritten as,

$$p(U\,|\,\bar{\lambda}) = \Psi\big(p(U\,|\,\bar{\lambda}_1), p(U\,|\,\bar{\lambda}_2)..., p(U\,|\,\bar{\lambda}_K)\big). \tag{2.4}$$

Using WAC or WGC to realize Eq. (2.4), we can form a trainable version of the conventional LR measures in Eqs. (1.3) – (1.6), where each anti-model $\bar{\lambda}_i$, i = 1,…,4, is computed, respectively, by

$$p(U\,|\,\bar{\lambda}_1) = p(U\,|\,\Omega), \tag{2.5}$$

$$p(U\,|\,\bar{\lambda}_2) = \max_{1 \le i \le B} p(U\,|\,\lambda_i), \tag{2.6}$$

$$p(U\,|\,\bar{\lambda}_3) = \frac{1}{B}\sum_{i=1}^{B} p(U\,|\,\lambda_i), \tag{2.7}$$

and

$$p(U\,|\,\bar{\lambda}_4) = \left(\prod_{i=1}^{B} p(U\,|\,\lambda_i)\right)^{\frac{1}{B}}. \tag{2.8}$$

As a result, for Eq. (1.7), each target speaker has the following sequence of background models, $\{\bar{\lambda}_1, \bar{\lambda}_2, \bar{\lambda}_3, \bar{\lambda}_4\}$. We denote systems that combine multiple anti-models as hybrid anti-model systems.

2.2. Gradient-based Minimum Verification Error Training

After representing $\Psi(\cdot)$ as a trainable combination of likelihoods, the task becomes a matter of solving the associated weights. To obtain an optimal set of weights, we propose using minimum verification error (MVE) training [Chou 2003, Rosenberg 1998].

The concept of MVE training stems from MCE training, where the former could be a special case of the latter when the classes to be distinguished are binary. To be specific, consider a set of class discriminant functions $g_i(U)$, i = 0,1,…, M - 1. The misclassification measure in the MCE method [Juang 1997] is defined as

$$d_i(U) = -g_i(U) + \log\left[\frac{1}{M-1}\sum_{j,j\neq i}\exp[g_j(U)\eta]\right]^{1/\eta}, \tag{2.9}$$

where η is a positive number. If $M = 2$, $\eta = 1$, and

$$g_i(U) = \begin{cases} \log p(U \mid \lambda) & \text{if } i = 0 \\ \log p(U \mid \bar{\lambda}) & \text{if } i = 1, \end{cases} \tag{2.10}$$

then $d_i(U)$ is reduced to the mis-verification measure defined in the MVE method:

$$d(U) = \begin{cases} d_0(U) = -L(U) & \text{if } U \in H_0 \\ d_1(U) = L(U) & \text{if } U \in H_1, \end{cases} \tag{2.11}$$

where $L(U)$ is the logarithmic LR. We further express $L(U)$ as the following equivalent test

$$L(U) = \log p(U \mid \lambda) - \log p(U \mid \bar{\lambda}) - \theta \begin{cases} \geq 0 & \text{accept } H_0 \\ < 0 & \text{accept } H_1, \end{cases} \tag{2.12}$$

so that the decision threshold θ can also be included in the optimization process. Then, the mis-verification measure is converted into a value between 0 and 1 using a sigmoid function

$$sg(d(U)) = \frac{1}{1 + \exp(-\varepsilon \cdot d(U))} , \tag{2.13}$$

where ε is a slope of the sigmoid function $sg(\cdot)$.

Next, we define the loss of each hypothesis as the average of the mis-verification measures of the training samples

$$\ell_i = \frac{1}{N_i}\sum_{U \in H_i} sg(d(U)), \tag{2.14}$$

where ℓ_0 denotes the loss associated with false rejection errors, ℓ_1 denotes the loss associated with false acceptance errors, and N_0 and N_1 are the numbers of utterances from true speakers and impostors, respectively. Finally, we define the overall expected loss as

$$D = x_0 \ell_0 + x_1 \ell_1, \tag{2.15}$$

where x_0 and x_1 indicate which type of error is of greater concern in a practical application.

Accordingly, our goal is to find the weights w_i in Eq. (2.1) and Eq. (2.2) such that Eq. (2.15) can be minimized. This can be achieved by using the gradient descent algorithm [Chou 2003]. To ensure that the weights satisfy $\sum_{i=1}^{N} w_i = 1$, we solve w_i by means of an intermediate parameter α_i, where

$$w_i = \frac{\exp(\alpha_i)}{\sum_{j=1}^{N} \exp(\alpha_j)}, \tag{2.16}$$

which is similar to the strategy used in [Juang 1997]. Parameter α_i is iteratively optimized using

$$\alpha_i^{(t+1)} = \alpha_i^{(t)} - \delta \frac{\partial D}{\partial \alpha_i}, \tag{2.17}$$

where δ is the step size, and

$$\begin{aligned}
\frac{\partial D}{\partial \alpha_i} &= x_0 \frac{\partial \ell_0}{\partial \alpha_i} + x_1 \frac{\partial \ell_1}{\partial \alpha_i} \\
&= x_0 \frac{\partial \ell_0}{\partial sg} \cdot \frac{\partial sg}{\partial d} \cdot \frac{\partial d}{\partial L} \cdot \frac{\partial L}{\partial \alpha_i} + x_1 \frac{\partial \ell_1}{\partial sg} \cdot \frac{\partial sg}{\partial d} \cdot \frac{\partial d}{\partial L} \cdot \frac{\partial L}{\partial \alpha_i} \\
&= x_0 \cdot \frac{1}{N_0} \sum_{U \in H_0} \left\{ a \cdot sg(-L(U))[1 - sg(-L(U))] \cdot \left(-\frac{\partial L}{\partial \alpha_i} \right) \right\} \\
&\quad + x_1 \cdot \frac{1}{N_1} \sum_{U \in H_1} \left\{ a \cdot sg(L(U))[1 - sg(L(U))] \cdot \frac{\partial L}{\partial \alpha_i} \right\},
\end{aligned} \tag{2.18}$$

where

$$\frac{\partial L}{\partial \alpha_i} = \sum_{j=1}^{N} \left(\frac{\partial L}{\partial w_j} \cdot \frac{\partial w_j}{\partial \alpha_i} \right) = w_i \left(\frac{\partial L}{\partial w_i} - \sum_{j=1}^{N} w_j \frac{\partial L}{\partial w_j} \right). \tag{2.19}$$

If WAC is used, then

$$\frac{\partial L}{\partial w_i} = \frac{-\partial}{\partial w_i}\log\left(\sum_{j=1}^{N} w_j p(U \mid \lambda_j)\right) = \frac{-p(U \mid \lambda_i)}{\sum_{j=1}^{N} w_j p(U \mid \lambda_j)}. \tag{2.20}$$

If WGC is used, then

$$\frac{\partial L}{\partial w_i} = \frac{-\partial}{\partial w_i}\left(\sum_{j=1}^{N} w_j \log p(U \mid \lambda_j)\right) = -\log p(U \mid \lambda_i). \tag{2.21}$$

The threshold θ in Eq. (2.12) can be estimated using

$$\theta^{(t+1)} = \theta^{(t)} - \delta \frac{\partial D}{\partial \theta}, \tag{2.22}$$

where

$$\begin{aligned}\frac{\partial D}{\partial \theta} &= x_0 \frac{\partial \ell_0}{\partial sg} \cdot \frac{\partial sg}{\partial d} \cdot \frac{\partial d}{\partial L} \cdot \frac{\partial L}{\partial \theta} + x_1 \frac{\partial \ell_1}{\partial sg} \cdot \frac{\partial sg}{\partial d} \cdot \frac{\partial d}{\partial L} \cdot \frac{\partial L}{\partial \theta} \\ &= x_0 \cdot \frac{1}{N_0} \sum_{U \in H_0} a \cdot sg(-L(U))\left[1 - sg(-L(U))\right] - x_1 \cdot \frac{1}{N_1} \sum_{U \in H_1} a \cdot sg(L(U))\left[1 - sg(L(U))\right].\end{aligned} \tag{2.23}$$

In our implementation, the overall expected loss is set as

$$D = C_{Miss} \times \ell_0 \times P_{Target} + C_{FalseAlarm} \times \ell_1 \times (1 - P_{Target}). \tag{2.24}$$

Eq. (2.24) simulates the Detection Cost Function (DCF) [Van Leeuwen 2006]

$$C_{DET} = C_{Miss} \times P_{Miss} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm} \times (1 - P_{Target}), \tag{2.25}$$

where C_{Miss} denotes the cost of the miss (false rejection) error; $C_{FalseAlarm}$ denotes the cost of the false alarm (false acceptance) error; $P_{Miss} \approx \ell_0$ is the miss (false rejection) probability; $P_{FalseAlarm} \approx \ell_1$ is the false alarm (false acceptance) probability; and P_{Target} is the *a priori* probability of the target speaker.

2.3. Evolutionary Minimum Verification Error Training

As the gradient descent approach may converge to an inferior local optimum, we propose an evolutionary MVE (EMVE) training method that uses a genetic algorithm (GA) to train the weights w_i and the threshold θ in WAC- and WGC-based LR measures. It has been shown in many applications that GA-based optimization is superior to gradient-based optimization, because of GA's global scope and parallel searching power.

Genetic algorithms belong to a particular class of evolutionary algorithms inspired by the process of natural evolution [Eiben 2003]. As shown in Fig. 2.1, the operators involved in the evolutionary process are: encoding, parent selection, crossover, mutation, and survivor selection. GAs maintain a population of candidate solutions and perform parallel searches in the search space via the evolution of these candidate solutions.

To accommodate GA to EMVE training, the fitness function of GA is set as the reciprocal of the overall expected loss D defined in Eq. (2.15), where $x_0 = C_{Miss} \times P_{Target}$ and $x_1 = C_{FalseAlarm} \times (1 - P_{Target})$. The details of the GA operations in EMVE training are described in the following.

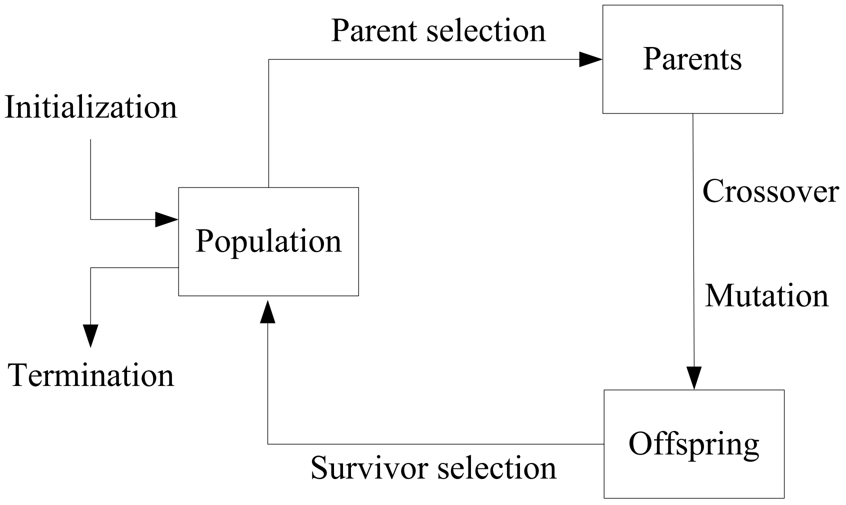


Fig. 2.1. The general scheme of a GA.

1) Encoding: Each chromosome is a string $\{\alpha_1, \alpha_2, ..., \alpha_N, \theta\}$ of length N + 1, which is the concatenation of all intermediate parameters α_i in Eq. (2.16) and the threshold θ in Eq. (2.12). Chromosomes are initialized by randomly assigning a real value to each gene.

2) Parent selection: Five chromosomes are randomly selected from the population with replacement, and the one with the best fitness value (i.e., with the smallest overall expected loss) is selected as a parent. The procedure is repeated iteratively until a pre-defined number (which is the same as the population size in this study) of parents is selected. This is known as *tournament selection* [Eiben 2003].

3) Crossover: We use the *N*-point crossover [Eiben 2003] in this work. Two chromosomes are randomly selected from the parent population with replacement. The chromosomes can interchange each pair of their genes in the same positions according to a crossover probability *pc*.

4) Mutation: In most cases, the function of the mutation operator is to change the allele of the gene randomly in the chromosomes. For example, while mutating a gene of a chromosome, we can simply draw a number from a normal distribution at random, and add it to the allele of the gene. However, the method does not guarantee that the fitness will improve steadily. We therefore designed a new mutation operator, called the one-step gradient descent operator (GDO). The concept of the GDO is similar to that of the one-step *K*-means operator (KMO) [Krishna 1999; Lu 2004; Cheng 2006], which guarantees to improve the fitness function after mutation by performing one iteration of the *K*-means algorithm.

The GDO performs one gradient descent iteration to update the parameters α_i, i = 1, 2, …, N as follows:

$$\alpha_i^{new} = \alpha_i^{old} - \delta \frac{\partial D}{\partial \alpha_i}, \tag{2.26}$$

where α_i^{new} and α_i^{old} are, respectively, the parameter α_i in a chromosome after and before mutation; δ is the step size ; and $\frac{\partial D}{\partial \alpha_i}$ is computed by Eq. (2.18). Similarly, the GDO for the threshold θ is computed by

$$\theta^{new} = \theta^{old} - \delta \frac{\partial D}{\partial \theta}, \tag{2.27}$$

where θ^{new} and θ^{old} are, respectively, the threshold θ in a chromosome after and before mutation; and $\frac{\partial D}{\partial \theta}$ is computed by Eq. (2.23).

5) Survivor selection: We adopt the generational model [Eiben 2003] in which the whole population is replaced by its offspring.

The process of fitness evaluation, parent selection, crossover, mutation, and survivor selection is repeated following the principle of survival of the fittest to produce better approximations of the optimal solution. Accordingly, it is hoped that the verification errors will decrease from generation to generation. When the maximum number of generations is reached, the best chromosome in the final population is taken as the solution of the weights.

As the proposed EMVE training method searches for the solution in a global manner, it is expected that its computational complexity is higher than that of the gradient-based MVE training. Assume that the population size of GA is *P*, while the numbers of iterations (or generations) of gradient-based MVE training and EMVE training are k_1 and k_2, respectively. The computational complexity of EMVE training is about Pk_2/k_1 times that of gradient-based MVE training. In our experiments (as shown in Fig. 2.2), the number of generations required for the convergence of EMVE training is roughly equal to the number of iterations required for the convergence of gradient-based MVE training; hence, the EMVE training roughly requires P times consumption of the gradient-based MVE training.

2.4. Experiments and Analysis

We evaluated the proposed approaches via speaker verification experiments conducted on speech data extracted from the Extended M2VTS Database (XM2VTSDB) [Messer 1999]. The first set of experiments followed Configuration II of XM2VTSDB, as defined in [Luettin 1998]. The second set of experiments followed a configuration that was modified from Configuration II of XM2VTSDB to conform to NIST Speaker Recognition Evaluation (NIST SRE) [Przybocki 2007; Van Leeuwen 2006].

In the experiments, the population size of the GA was set to 50, the maximum number of generations was set to 100, and the crossover probability *pc* was set to 0.5 for the EMVE training; the gradient-based MVE training for the WAC and WGC methods was initialized with an equal weight, w_i, and the threshold θ was set to 0. For the DCF in Eq. (2.25), the costs C_{Miss} and $C_{FalseAlarm}$ were both set to 1, and the *a priori* probability P_{Target} was set to 0.5. This special case of DCF is known as the Half Total Error Rate (HTER) [Lindberg 1998]. All the experiments were conducted on a 3.2 GHz Intel Pentium IV computer with 1.5 GB of RAM, running Windows XP.

2.4.1. Evaluation based on Configuration II

In accordance with Configuration II of XM2VTSDB, the database was divided into three subsets: “Training”, “Evaluation*”, and “Test”. We used the “Training” subset to build each

* This is usually called the “Development” set by the speech recognition community. We use “Evaluation” in accordance with the configuration of XM2VTSDB.

target speaker's model and the background models. The "Evaluation" subset was used to optimize the weights w_i in Eq. (2.1) or Eq. (2.2), along with the threshold θ. Then, the speaker verification performance was evaluated on the "Test" subset. As shown in Table 2.1, a total of 293 speakers[†] in the database were divided into 199 clients (target speakers), 25 "evaluation impostors", and 69 "test impostors". Each speaker participated in four recording sessions at about one-month intervals, and each recording session consisted of two shots. In each shot, the speaker was prompted to utter three sentences:

a) "0 1 2 3 4 5 6 7 8 9".

b) "5 0 6 9 2 8 1 3 7 4".

c) "Joe took father's green shoe bench out".

Table 2.1. Configuration II of XM2VTSDB.

Session	Shot	199 clients	25 impostors	69 impostors
1	1	Training	Evaluation	Test
	2			
2	1			
	2			
3	1	Evaluation		
	2			
4	1	Test		
	2			

Each utterance, sampled at 32 kHz, was converted into a stream of 24-order feature vectors by a 32-ms Hamming-windowed frame with 10-ms shifts; and each vector consisted of 12 Mel-scale frequency cepstral coefficients [Huang 2001] and their first time derivatives.

We used 12 (2×2×3) utterances/client from sessions 1 and 2 to train each client model,

[†] We omitted 2 speakers (ID numbers 313 and 342) because of partial data corruption.

represented by a GMM with 64 mixture components. For each client, we used the utterances of the other 198 clients in sessions 1 and 2 to generate the world model, represented by a GMM with 512 mixture components. We then chose B speakers from those 198 clients as the cohort. In the experiments, B was set to 50, and each cohort model was also represented by a GMM with 64 mixture components. Table 2.2 summarizes all the parametric models used in each system.

To optimize the weights, w_i, and the threshold, θ, we used 6 utterances/client from session 3 and 24 (4×2×3) utterances/evaluation-impostor over the four sessions, which yielded 1,194 (6×199) client samples and 119,400 (24×25×199) impostor samples. To speed up the gradient-based MVE and EMVE training processes, only 2,250 impostor samples randomly selected from the total of 119,400 samples were used. In the performance evaluation, we tested 6 utterances/client in session 4 and 24 utterances/test-impostor over the four sessions, which involved 1,194 (6×199) client trials and 329,544 (24×69×199) impostor trials.

Table 2.2. A summary of the parametric models used in each system.

System	H_0	H_1	
	a 64-mixture client GMM	a 512-mixture world model	B 64-mixture cohort GMMs
L_{UBM}	√	√	
L_{Max}	√		√
L_{Ari}	√		√
L_{Geo}	√		√
WGC	√	√	√
WAC	√	√	√

A. Experiment results

First, we compared the learning ability of gradient-based MVE training and EMVE training in the proposed WGC- and WAC-based LR measures. The background models comprised either (i) the world model and the 50 closest cohort models (“w_50c”), or (ii) the world model and the 25 closest cohort models, plus the 25 farthest cohort models (“w_25c_25f”). The WGC- and WAC-based LR systems were implemented in four ways:

a) Using gradient-based MVE training and “w_50c” (“WGC_MVE_w_50c”; “WAC_MVE_w_50c”),
b) Using gradient-based MVE training and “w_25c_25f” (“WGC_MVE_w_25c_25f”; “WAC_MVE_w_25c_25f”),
c) Using EMVE training and “w_50c” (“WGC_EMVE_w_50c”; “WAC_EMVE_w_50c”), and
d) Using EMVE training and “w_25c_25f” (“WGC_EMVE_w_25c_25f”; “WAC_EMVE_w_25c_25f”).

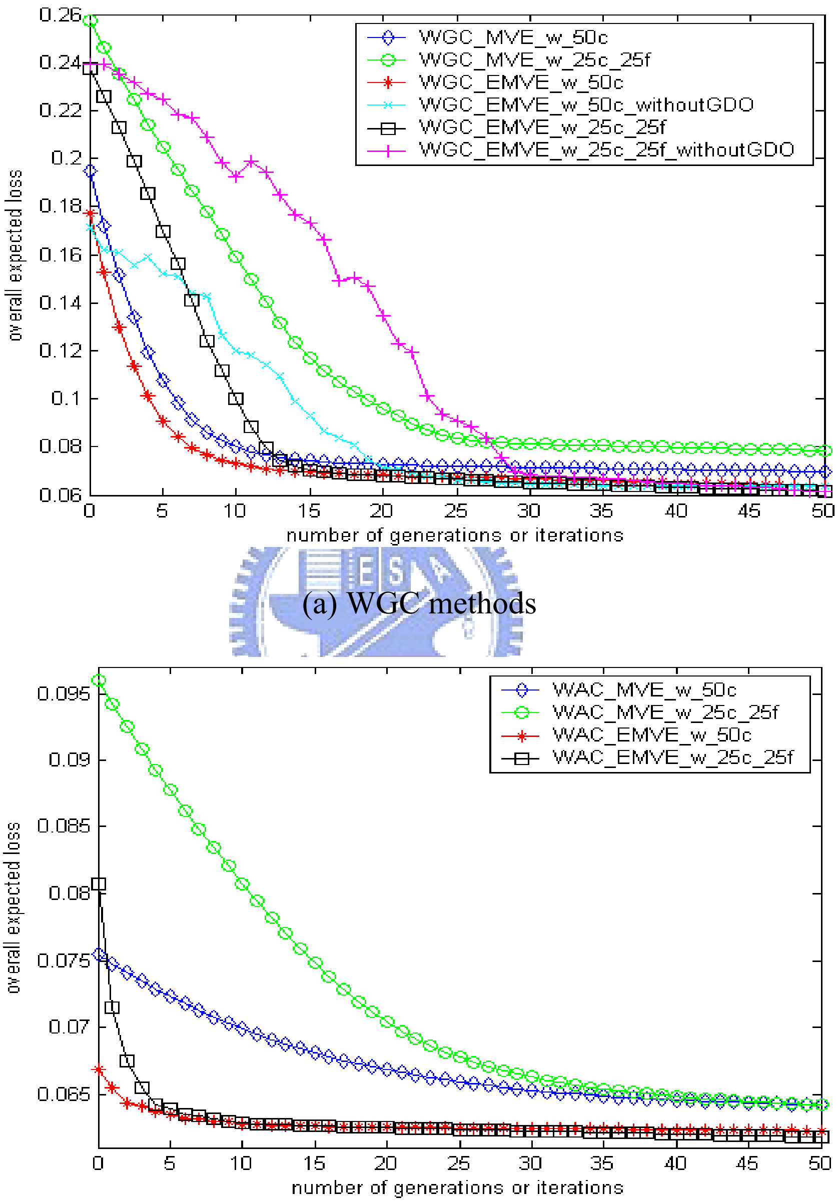
Figs. 2.2(a) and 2.2(b) show the learning curves of different MVE training methods for WGC and WAC on the “Evaluation” subset, respectively, where “WGC_EMVE_w_50c_withoutGDO” and “WGC_EMVE_w_25c_25f_withoutGDO” denote the EMVE training algorithms that use the conventional mutation operator, which changes the allele of the gene in a chromosome at random, while the others are based on the GDO mutation. From Fig. 2.2, we observe that the GDO-based EMVE training method reduces the overall expected loss more effectively and steadily than the EMVE training method without GDO and the gradient-based MVE training method.

For the performance comparison, we used the following LR systems as our baselines:

a) $L_{\text{UBM}}(U)$ (“Lubm”),
b) $L_{\text{Max}}(U)$ with the 50 closest cohort models (“Lmax_50c”),

c) $L_{\text{Geo}}(U)$ with the 50 closest cohort models (“Lgeo_50c”),

d) $L_{\text{Geo}}(U)$ with the 25 closest cohort models and the 25 farthest cohort models (“Lgeo_25c_25f”),



(a) WGC methods

(b) WAC methods

Fig. 2.2. The learning curves of gradient-based MVE and EMVE for the “Evaluation” subset in Configuration II.

e) $L_{\mathrm{Ari}}(U)$ with the 50 closest cohort models (“Lari_50c”), and

f) $L_{\mathrm{Ari}}(U)$ with the 25 closest cohort models and the 25 farthest cohort models (“Lari_25c_25f”).

Fig. 2.3 shows the Detection Error Tradeoff (DET) curves [Martin 1997] obtained by evaluating the above systems using the “Test” subset, where Fig. 2.3(a) compares the WGC-based approach and the geometric mean approach, while Fig. 2.3(b) compares the WAC-based approach and the arithmetic mean approach. From the figure, we observe that all the WGC-based LR systems outperform the baseline LR systems “Lubm”, “Lmax_50c”, “Lgeo_50c”, and “Lgeo_25c_25f”, while all the WAC-based LR systems outperform the baseline LR systems “Lubm”, “Lari_50c”, and “Lari_25c_25f”. From Fig. 2.3(a), we observe that “Lgeo_25c_25f” yields the poorest performance. This is because the heuristic geometric mean can produce some singular scores if any cohort model λ_i is poorly matched with the input utterance U, i.e., $p(U|\ \lambda_i) \to 0$. In contrast, the results show that the WGC-based LR systems sidestep this problem with the aid of the weighted strategy. Figs. 2.3(a) and 2.3(b) also show that “WGC_EMVE_w_50c”, “WGC_EMVE_w_25c_25f”, and “WAC_EMVE_w_25c_25f” outperform “WGC_MVE_w_50c”, “WGC_MVE_w_25c_25f”, and “WAC_MVE_w_25c_25f”, respectively. However, there is no significant difference between “WAC_MVE_w_50c” and “WAC_EMVE_w_50c”.

In addition to the above systems, we also evaluated the WAC- and WGC-based LR measures using the hybrid anti-model defined in Eq. (2.4). The hybrid anti-model comprised five conventional anti-models extracted from “Lubm”, “Lmax_50c”, “Lgeo_50c”, “Lari_50c”, and “Lari_25c_25f”. Note that the anti-model of “Lgeo_25c_25f” was not included because of its poor performance. The hybrid anti-model systems were implemented in the following ways:

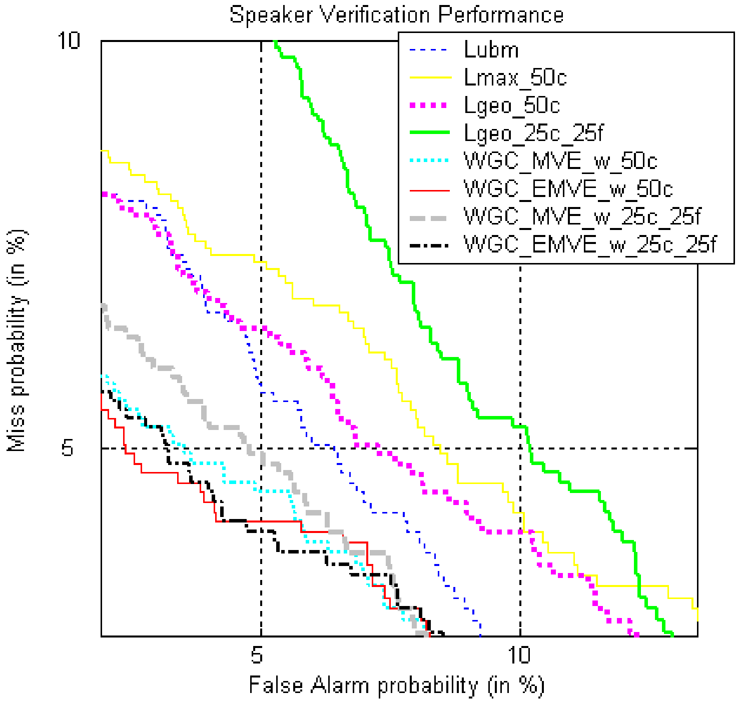
a) Using WAC and gradient-based MVE training (“WAC_MVE_5anti”),

b) Using WGC and gradient-based MVE training (“WGC_MVE_5anti”),

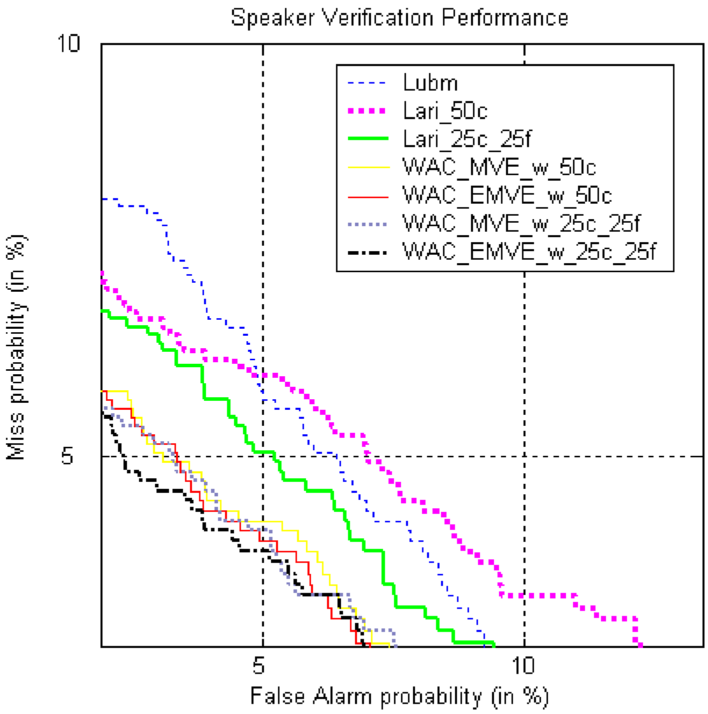
c) Using WAC and EMVE training (“WAC_EMVE_5anti”), and

d) Using WGC and EMVE training (“WGC_EMVE_5anti”).

Fig. 2.4 compares the performance of the hybrid anti-model systems with all the baselines systems, evaluated on the “Test” subset in DET curves. Clearly, all the hybrid anti-model systems using either WAC or WGC methods outperform any baseline LR system with a single anti-model.



(a) Geometric mean versus WGC



(b) Arithmetic mean versus WAC

Fig. 2.3. DET curves for the “Test” subset in Configuration II.

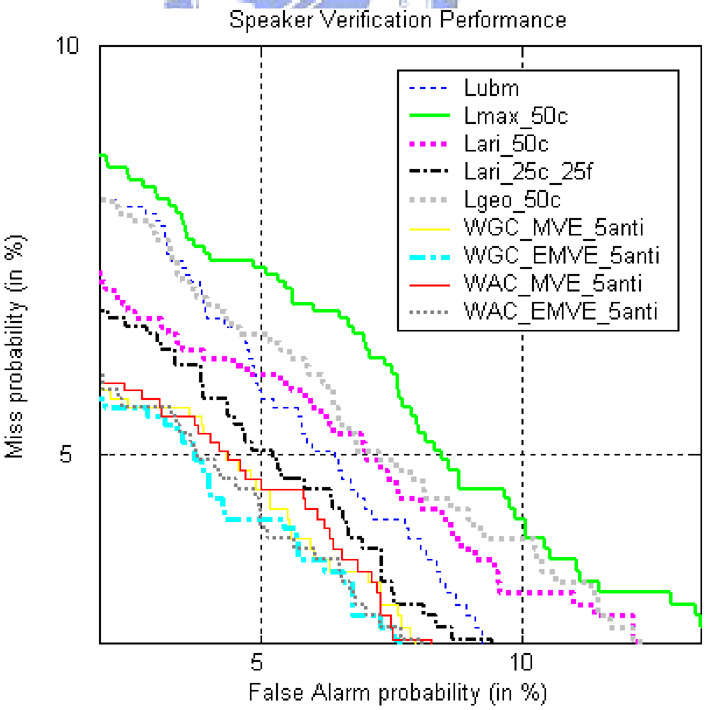


Fig. 2.4. Hybrid anti-model systems versus all baselines: DET curves for the “Test” subset in Configuration II.

B. Discussion

Table 2.3 summarizes the above experiment results in terms of the DCF, which reflects the performance at a specific operating point on the DET curve. For each baseline system, the value of the decision threshold θ was carefully tuned to minimize the DCF in the “Evaluation” subset, and then applied to the “Test” subset. However, the decision thresholds of the proposed WAC- and WGC-based LR measures were optimized automatically using the “Evaluation” subset, and then applied to the “Test” subset.

Table 2.3. DCFs for the “Evaluation” and “Test” subsets in Configuration II.

System	min DCF for “Evaluation”	DCF for “Test”
Lubm	0.0651	0.0545
Lmax_50c	0.0762	0.0575
Lari_50c	0.0677	0.0526
Lari_25c_25f	0.0587	0.0496
Lgeo_50c	0.0749	0.0542
WGC_MVE_w_50c	0.0576	0.0450
WGC_EMVE_w_50c	0.0488	0.0417
WGC_MVE_w_25c_25f	0.0633	0.0478
WGC_EMVE_w_25c_25f	0.0493	0.0429
WAC_MVE_w_50c	0.0576	0.0460
WAC_EMVE_w_50c	0.0571	0.0443
WAC_MVE_w_25c_25f	0.0573	0.0462
WAC_EMVE_w_25c_25f	0.0543	0.0444
WGC_MVE_5anti	0.0588	0.0475
WGC_EMVE_5anti	0.0568	0.0460
WAC_MVE_5anti	0.0634	0.0480
WAC_EMVE_5anti	0.0597	0.0469

Several conclusions can be drawn from Table 2.3. First, all the proposed WAC- and WGC-based LR systems with either the hybrid anti-model or the background model set (the world model plus a cohort) outperform all the baseline LR systems. Second, the performances of the proposed systems using the background model set are slightly better than those achieved using the hybrid anti-model. Third, the performances of the WAC- and WGC-based

LR systems are similar. Fourth, EMVE training is better than MVE training. Among the systems, “WGC_EMVE_w_50c” achieves the best performance with a 15.93% relative improvement in terms of the DCF for the “Test” subset, compared to the best baseline system “Lari_25c_25f”.

2.4.2. Evaluation based on the NIST SRE-like Configuration

To conform to NIST SRE [Przybocki 2007; Van Leeuwen 2006], we conducted another series of experiments on XM2VTSDB, which was re-configured as shown Table 2.4. The 293 speakers in XM2VTSDB were divided into 100 clients (target speakers), 100 background speakers, 24 “development impostors”, and 69 “test impostors”. As shown in the table, the “Development” set comprised two subsets: “Development training” and “Development test”. In the “Development training” subset, we pooled the utterances of 100 background speakers from sessions 1 and 2 to build a world model (UBM), represented by a GMM with 512 mixture components. For each background speaker, we used 12 (2×2×3) utterances/background-speaker from sessions 1 and 2 to generate his/her model. The cohort for each background speaker was selected from the other 99 background speakers. In the “Development test” subset, to estimate the weights w_i and the threshold θ, we used 12 (2×2×3) utterances/background-speaker from sessions 3 and 4 as well as 24 (4×2×3) utterances/development-impostor over the four sessions. This yielded 1,200 (12×100) client samples and 57,600 (24×24×100) impostor samples. To speed up the gradient-based MVE and EMVE training processes, only 5,760 impostor samples randomly selected from the total of 57,600 samples were used.

For each client (target speaker), we used 12 (2×2×3) utterances/client from sessions 1 and 2 to generate the client GMM. The cohort models for each client were selected from the

GMMs of the 100 background speakers in the “Development training” subset. The parametric models used in each system were the same as those in Table 2.2. In addition, we implemented two current state-of-the-art systems in the text-independent speaker verification task, namely T-norm [Auckenthaler 2000] and “Lubm_MAP”. “Lubm_MAP” is based on the UBM-MAP adaptation method [Reynolds 2000]; each client model with 512 mixture Gaussian components was adapted from the UBM via the maximum a posteriori (MAP) estimation [Gauvain 1994] according to the speaker’s 12 (2×2×3) “Training” utterances from sessions 1 and 2.

In the performance evaluation, we tested 12 (2×2×3) utterances/client from sessions 3 and 4, and 24 (4×2×3) utterances/test-impostor over the four sessions, which involved 1,200 (12×100) client trials and 165,600 (24×69×100) impostor trials, respectively.

Table 2.4. The NIST SRE-like configuration of XM2VTSDB.

Session	Shot	100 clients	100 background speakers	24 impostors	69 impostors
1	1	Training (client models)	Development training (UBM, a cohort)	Development test (w_i and θ)	Test
	2				
2	1				
	2				
3	1	Test	Development test (w_i and θ)		
	2				
4	1				
	2				

A. Experiment results

As in Section 2.4.1, we implemented four WGC-based LR systems: “WGC_MVE_w_50c”, “WGC_EMVE_w_50c”, “WGC_MVE_w_25c_25f”, and “WGC_EMVE_w_25c_25f”; four WAC-based LR systems: “WAC_MVE_w_50c”, “WAC_EMVE_w_50c”, “WAC_MVE_w_25c_25f”, and “WAC_EMVE_w_25c_25f”; and four hybrid anti-model systems: “WAC_MVE_5anti”, “WAC_EMVE_5anti”, “WGC_MVE_5anti”, and “WGC_EMVE_5anti”. For the performance comparison, we used five conventional LR systems: “Lubm”, “Lmax_50c”, “Lgeo_50c”, “Lari_50c”, and “Lari_25c_25f”, plus two state-of-the-art systems: “Lubm_MAP” and the T-norm system with the 50 closest cohort models (“Tnorm_50c”), as our baselines.

Since the experiment results in Section 2.4.1 show that the performance of the proposed WGC- and WAC-based LR systems using EMVE training is better than that of the systems using gradient-based MVE training, Fig. 2.5 only compares the performance of the proposed WGC- and WAC-based LR systems using EMVE training with two state-of-the-art systems and two best baseline systems in Section 2.4.1, namely “Lubm” and “Lari_25c_25f”, evaluated on the “Test” subset in DET curves. From the figure, we observe that all the proposed WGC- and WAC-based LR systems using EMVE training outperform “Lubm_MAP”, “Tnorm_50c”, “Lubm”, and “Lari_25c_25f”. Interestingly, the baseline system “Lubm” outperforms “Lubm_MAP”, which is widely recognized as a state-of-the-art method for the text-independent speaker verification task. This may be because the training and test utterances in XM2VTSDB have the same content.

Table 2.5 summarizes the experiment results for all systems in terms of the DCF. For each baseline system, the decision threshold θ was tuned to minimize the DCF on the “Development test” subset, and then applied to the “Test” subset. The decision thresholds of the proposed methods were optimized automatically using the “Development test” subset, and then applied to the “Test” subset. From Table 2.5, it is clear that all the proposed WGC- and WAC-based LR systems using either gradient-based MVE training or EMVE training outperform all the conventional LR systems “Lubm”, “Lmax_50c”, “Lgeo_50c”, “Lari_50c”, and “Lari_25c_25f”, and two state-of-the-art systems “Lubm_MAP” and “Tnorm_50c”. The DCFs for the “Test” subset demonstrate that “WGC_EMVE_w_50c” achieved a 13.01%

relative improvement over "Tnorm_50c" – the best baseline system.

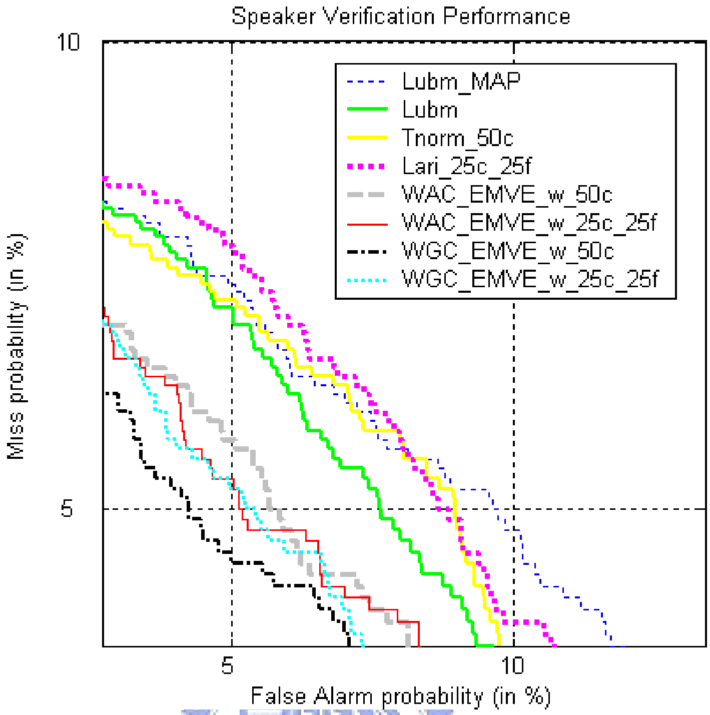


Fig. 2.5. DET curves for the "Test" subset in the NIST SRE-like configuration.

We also evaluated the training and verification time of the above systems. In the offline training phase, in addition to training 100 background speaker models and a UBM, the proposed WAC and WGC methods need to train the weight w_i. From the fourth column of Table 2.5, we observe that the EMVE training is slower than the gradient-based MVE training and the training time of WGC is slightly faster than that of WAC. The computational cost in gradient-based MVE or EMVE training mainly comes from the calculation of the likelihoods of each training utterance with respect to the background speaker models and the UBM and the selection of the cohort models for each background speaker. The fifth column of Table 2.5 shows the training time for enrolling a new target speaker. "Lubm_MAP" and "Lubm" need less enrollment time than the other systems because they need not select the cohort models for the new target speaker. The last column of Table 2.5 shows the verification time for an input test utterance. The average duration of the test utterances is around 1.5 sec. As expected, “Lubm_MAP” is the fastest method, since only one background model (i.e., UBM) is involved and the fast scoring scheme [Reynolds 2000] is used. Although the proposed systems are slightly slower than the baseline systems because both the cohort models and the UBM are involved, they are still capable of supporting a real-time response.

Table 2.5. DCFs for the “Development test” and “Test” subsets, together with the running time evaluation in the NIST SRE-like configuration.

System	min DCF for “Development test”	DCF for “Test”	Training time for the weights w_i in WAC/WGC (offline)	Training time for enrolling a target speaker	Verification time for an input test utterance
Lubm_MAP	0.0704	0.0601		5.79sec	0.08sec
Lubm	0.0575	0.0573		7.87sec	0.12sec
Tnorm_50c	0.0607	0.0569		27.46sec	0.75sec
Lmax_50c	0.0732	0.0734		27.46sec	0.75sec
Lari_50c	0.0653	0.0600		27.46sec	0.75sec
Lari_25c_25f	0.0611	0.0588		27.46sec	0.75sec
Lgeo_50c	0.0758	0.0692		27.46sec	0.75sec
WGC_MVE_w_50c	0.0578	0.0529	3hr 06min 22.31sec	27.46sec	0.86sec
WGC_EMVE_w_50c	0.0479	0.0495	3hr 22min 15.38sec	27.46sec	0.86sec
WGC_MVE_w_25c_25f	0.0610	0.0570	3hr 06min 22.31sec	27.46sec	0.86sec
WGC_EMVE_w_25c_25f	0.0485	0.0509	3hr 22min 15.40sec	27.46sec	0.86sec
WAC_MVE_w_50c	0.0575	0.0546	3hr 06min 25.09sec	27.46sec	0.86sec
WAC_EMVE_w_50c	0.0556	0.0533	3hr 24min 50.14sec	27.46sec	0.86sec
WAC_MVE_w_25c_25f	0.0564	0.0549	3hr 06min 25.09sec	27.46sec	0.86sec
WAC_EMVE_w_25c_25f	0.0543	0.0527	3hr 24min 50.15sec	27.46sec	0.86sec
WGC_MVE_5anti	0.0583	0.0541	3hr 06min 15.58sec	27.46sec	0.86sec
WGC_EMVE_5anti	0.0576	0.0514	3hr 09min 54.53sec	27.46sec	0.86sec
WAC_MVE_5anti	0.0610	0.0556	3hr 06min 15.72sec	27.46sec	0.86sec
WAC_EMVE_5anti	0.0587	0.0566	3hr 10min 15.70sec	27.46sec	0.86sec

Chapter 3

Improving the Characterization of the Alternative Hypothesis Using Kernel Discriminant Analysis

In this chapter, we further propose improving the characterization of the alternative hypothesis by designing two decision functions based on WAC and WGC. We can regard the proposed decision functions as nonlinear discriminant classifiers. The parameters associated with the classifiers are then optimized using two kernel discriminant analysis techniques, namely, the Kernel Fisher Discriminant (KFD) [Mika 1999, 2002] and Support Vector Machine (SVM) [Burges 1998]. The proposed approaches have two advantages over existing methods. The first is that they embed a trainable mechanism in the decision functions. The second is that they convert variable-length utterances into fixed-dimension characteristic vectors, which are easily processed by kernel discriminant analysis.

In recent years, a number of SVM-based speaker verification techniques have been developed [Campbell 2006, 2007; Bengio 2001; Wan 2005]. One of the main issues with using SVMs for speaker verification is that the number of training samples represented by frames is usually too large to handle efficiently. For this reason, the concept of a sequence kernel [Campbell 2006, 2007; Bengio 2001; Wan 2005] was proposed to compare speech utterances at the sequence level instead of the frame level. However, constructing a proper sequence kernel for utterance-based SVMs is an issue that requires further investigation. In this work, as the proposed WGC and WAC methods convert variable-length utterances into fixed-dimension characteristic vectors, the derived kernel processes play the same role as the sequence kernel method, but they have the advantage of not having to specifically design the kernel functions.

In addition, most existing SVM-based speaker verification approaches only use a single background model, i.e., the world model, instead of multiple background models, to characterize the alternative hypothesis. For example, Bengio et al. [Bengio 2001] proposed the following decision function:

$$L_{\text{Bengio}}(U) = a_1 \log p(U \mid \lambda) - a_2 \log p(U \mid \Omega) + a_3, \tag{3.1}$$

where a_1, a_2, and a_3 are adjustable parameters estimated using SVM. The input to SVM comprises the two-dimensional vector $[\log p(U \mid \lambda)\ \ \text{-} \log p(U \mid \Omega)]'$. An extended version of Eq. (3.1) using the Fisher kernel and the LR score-space kernel for SVM was investigated in [Wan 2005]. In contrast, our framework integrates more available information from multiple background models into a characteristic vector as the input to SVM, which makes it easier to distinguish one hypothesis from another. The results of speaker verification experiments conducted on both the XM2VTSDB and the ISCSLP2006-SRE database show that the proposed kernel-based methods outperform all of the conventional approaches.

The remainder of this chapter is organized as follows. Section 3.1 introduces the design of the decision function used in our methods. Section 3.2 presents the kernel discriminant analysis techniques that we use to find the weight vector. Sections 3.3 describe the concepts related to the characteristic vector. Then, in Section 3.4, we detail the experiment results.

3.1. The Proposed Decision Functions

To handle the speaker-verification problem more effectively, it is necessary to devise a decision function with a trainable mechanism, such that one hypothesis can be optimally separated from another. To this end, we formulate the characterization of the alternative hypothesis as a problem of optimally combining the discriminative information derived from a set of pre-trained background models, and design the decision function based on two perspectives: a weighted geometric combination (WGC) and a weighted arithmetic combination (WAC) of the likelihoods of the background models.

We begin by rewriting the function $\Psi(\cdot)$ in Eq. (2.2) in terms of WGC as

$$\Psi(p(U\,|\,\lambda_1),\ldots,p(U\,|\,\lambda_N)) = \left(\prod_{i=1}^{N} p(U\,|\,\lambda_i)^{w_i}\right)^{1/(w_1+w_2+\ldots+w_N)}. \tag{3.2}$$

By substituting Eq. (3.2) into Eq. (1.7), and taking the logarithmic form, we obtain

$$\begin{aligned} L_{\mathrm{WGC}}(U) &= \log\left(\prod_{i=1}^{N}\left(\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_i)}\right)^{w_i}\right)^{1/(w_1+w_2+\ldots+w_N)} \\ &= \frac{1}{w_1+w_2+\ldots+w_N}\sum_{i=1}^{N} w_1 \log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_i)} \begin{cases} \ge \log\theta & \text{accept} \\ < \log\theta & \text{reject}, \end{cases} \\ &= \mathbf{w}'\mathbf{x} \begin{cases} \ge \theta_1 & \text{accept} \\ < \theta_1 & \text{reject}, \end{cases} \end{aligned} \tag{3.3}$$

where $\mathbf{w} = [w_1\ w_2 \ldots w_N]'$ is an $N \times 1$ weight vector, the new threshold $\theta_1 = (w_1 + w_2 + \ldots + w_N)\log\theta$, and $\mathbf{x}$ is an $N \times 1$ vector in the space R^N expressed as

$$\mathbf{x} = [\log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_1)}\ \log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_2)}\ \ldots\ \log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_N)}]'. \tag{3.4}$$

The implicit idea in Eq. (3.4) is that the input utterance U can be represented by a characteristic vector $\mathbf{x}$.

Alternatively, we can also rewrite the function $\Psi(\cdot)$ in Eq. (2.1) in terms of WAC as

$$\Psi(p(U \mid \lambda_1),\ldots, p(U \mid \lambda_N)) = \frac{1}{w_1 + w_2 + \ldots + w_N} \sum_{i=1}^{N} w_i \, p(U \mid \lambda_i). \tag{3.5}$$

By substituting Eq. (3.5) into Eq. (1.7) and reversing Eq. (1.7), we obtain

$$\begin{aligned} L_{\text{WAC}}(U) &= \frac{1}{L(U)} = \frac{1}{w_1 + w_2 + \ldots + w_N} \sum_{i=1}^{N} w_i \frac{p(U \mid \lambda_i)}{p(U \mid \lambda)} \begin{cases} \le 1/\theta & \text{accept} \\ > 1/\theta & \text{reject}, \end{cases} \\ &= \mathbf{w}'\mathbf{x} \begin{cases} \le \theta_2 & \text{accept} \\ > \theta_2 & \text{reject}, \end{cases} \end{aligned} \tag{3.6}$$

where $\mathbf{w} = [w_1 \; w_2 \ldots w_N]'$ is an N × 1 weight vector, the new threshold $\theta_2 = (w_1 + w_2 + \ldots + w_N)/\theta$, and $\mathbf{x}$ is an N × 1 characteristic vector in the space R^N , expressed by

$$\mathbf{x} = [\frac{p(U \mid \lambda_1)}{p(U \mid \lambda)} \; \frac{p(U \mid \lambda_2)}{p(U \mid \lambda)} \ldots \frac{p(U \mid \lambda_N)}{p(U \mid \lambda)}]'. \tag{3.7}$$

3.2. Kernel Discriminant Analysis

The process of representing an utterance U as a characteristic vector $\mathbf{x}$ in Eq. (3.4) or Eq. (3.7) can be regarded as $\mathbf{x} = \Phi(U)$, where $\Phi(\cdot)$[1] is a nonlinear mapping function. If we replace the threshold θ_1 in Eq. (3.3) or θ_2 in Eq. (3.6) with a bias w_0, the decision functions in Eqs. (3.3) and (3.6) can be rewritten as

$$L(U) = \mathbf{w}'\Phi(U) + w_0, \tag{3.8}$$

where $L(U)$ forms a nonlinear discriminant classifier for U. The classifier translates the goal of solving an LR test problem into one of optimizing $\mathbf{w}$ and w_0, such that the utterances of target speakers and non-target speakers can be separated. To realize this classifier, we need three distinct data sets: one for generating each target speaker's model, one for generating the background models, and one for optimizing **w** and w_0. Since the bias w_0 plays the same role as the decision threshold θ of the LR test, which can be determined through a tradeoff between the false acceptance and the false rejection rates, our main goal here is to find **w**.

To solve the weight vector **w**, we propose using two kernel-based discriminant techniques, namely the Kernel Fisher Discriminant (KFD) and Support Vector Machine (SVM), because of their ability to separate samples of target speakers from those of non-target speakers efficiently.

3.2.1. Kernel Fisher Discriminant (KFD)

Suppose that we have N_i training utterances $\{U_1^i,..,U_{n_i}^i\}$ for hypothesis H_i, $i = 0$ or 1. The goal of KFD is to locate the weight vector **w** that maximizes the between-class scatter, while minimizing the within-class scatter. According to [Mika 1999], the solution of **w** must lie in the span of all mapped training utterances; therefore, we can represent **w** as

$$\mathbf{w} = \sum_{j=1}^{J} \gamma_j \Phi(U_j), \tag{3.9}$$

where $\{U_j, 1 \le j \le J\} = \{U_1^0, U_2^0,..,U_{n_0}^0\} \cup \{U_1^1, U_2^1,..,U_{n_1}^1\}$, $J = n_0 + n_1$, and γ_j is the combination coefficient. Substituting Eq. (3.9) into Eq. (3.8), we obtain

$$L(U) = \sum_{j=1}^{J} \gamma_j \Phi(U_j)' \Phi(U) + w_0 = \sum_{j=1}^{J} \gamma_j k(U_j, U) + w_0 , \tag{3.10}$$

where the inner product of two vectors $\Phi(U_j)$ and $\Phi(U)$ is expressed by a kernel function $k(U_j$,

[1] More precisely, $\Phi(U)$ should be denoted by $\Phi(U; \lambda; \lambda_1, \lambda_2, ..., \lambda_N)$.

U). Such a kernel function is also called the sequence kernel [Campbell 2006], because it takes two utterance sequences, U_j and U, as inputs. The goal therefore changes from finding $\mathbf{w}$ to finding $\boldsymbol{\gamma} = [\gamma_1\ \gamma_2 \ldots \gamma_J]'$, which maximizes

$$\Gamma(\boldsymbol{\alpha}) = \frac{\boldsymbol{\gamma}'\mathbf{M}\boldsymbol{\gamma}}{\boldsymbol{\gamma}'\mathbf{N}\boldsymbol{\gamma}}. \tag{3.11}$$

M and **N** are computed by

$$\mathbf{M} = (\boldsymbol{\eta}_0 - \boldsymbol{\eta}_1)(\boldsymbol{\eta}_0 - \boldsymbol{\eta}_1)' \tag{3.12}$$

and

$$\mathbf{N} = \sum_{i=0,1} \mathbf{K}_i(\mathbf{I}_{n_i} - \mathbf{1}_{n_i})\mathbf{K}_i{}', \tag{3.13}$$

respectively, where $\boldsymbol{\eta}_i$ is an J×1 vector with element $(\eta_i)_s = (1/n_i)\sum_{j=1}^{n_i} k(U_s, U_j^i)$; $\mathbf{K}_i$ is an J×n_i matrix with element $(\mathrm{K}_i)_{sj} = k(U_s, U_j^i)$; $\mathbf{I}_{n_i}$ is an n_i×n_i identity matrix; and $\mathbf{1}_{n_i}$ is an n_i×n_i matrix in which all elements are equal to $1/n_i$. Following [Mika 2002], the solution to $\boldsymbol{\gamma}$, which maximizes $\Gamma(\boldsymbol{\gamma})$ defined in Eq. (3.11), is taken as the leading eigenvector of $\mathbf{N}^{-1}\mathbf{M}$.

3.2.2. Support Vector Machine (SVM)

The weight vector $\mathbf{w}$ can also be solved with SVM. In this case, the goal is to find a separating hyperplane that maximizes the margin between the classes. Following [Burges 1998], $\mathbf{w}$ can be expressed as

$$\mathbf{w} = \sum_{j=1}^{J} y_j \beta_j \Phi(U_j), \tag{3.14}$$

which yields

$$L(U)=\sum_{j=1}^{J} y_j \beta_j k(U_j, U)+w_0, \tag{3.15}$$

where each training utterance U_j, j = 1, 2,…, J, is labeled by either $y_j = 1$ (a null hypothesis) or $y_j = -1$ (an alternative hypothesis). The optimal coefficients $\boldsymbol{\beta}=[\beta_1\ \beta_2 \ldots \beta_J]'$ can be determined by maximizing the objective function

$$Q(\boldsymbol{\beta})=\sum_{j=1}^{J}\beta_j-\frac{1}{2}\sum_{i=1}^{J}\sum_{j=1}^{J} y_i y_j \beta_i \beta_j k(U_i, U_j), \tag{3.16}$$

subject to the constraints $\sum_{j=1}^{J} y_j \beta_j = 0$ and $0 \le \beta_j \le C_\beta$, $\forall j$, where C_β is a penalty parameter [Burges 1998]. This process can be performed with quadratic programming techniques [Vapnik 1998]. Note that most elements of β are equal to zero, and training samples associated with non-zero β_j are called *support vectors*. A few support vectors play a key role in deciding the optimal margin between classes in SVM.

3.2.3. Mercer Kernels

The effectiveness of the above KFD or SVM approaches depends essentially on how the kernel function $k(\cdot)$ is designed. A kernel function must be symmetric, positive definite, and conform to Mercer's condition [Herbrich 2002]. There are a number of kernel functions [Herbrich 2002]. However, since we have converted speech utterances into characteristic vectors, the kernel function takes the form

$$k(U_1, U_2)=\Phi(U_1)'\Phi(U_2)=\mathbf{x}_1'\mathbf{x}_2=k_1(\mathbf{x}_1, \mathbf{x}_2). \tag{3.17}$$

Eq. (3.17) indicates that the sequence kernel function with two input utterances, U_1 and U_2, forms a dot product kernel with two input characteristic vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$. Alternatively, if we use the closure property of Mercer kernels [Herbrich 2002] to form a kernel function

$$\hat{k}(U_1,U_2) = \exp\left(-\frac{k(U_1,U_1) + k(U_2,U_2) - 2k(U_1,U_2)}{2\sigma^2} \right), \tag{3.18}$$

where σ is a tunable parameter, then $\hat{k}(U_1,U_2)$ is equivalent to the following Radial Basis Function (RBF) kernel with two inputs $\mathbf{x}_1$ and $\mathbf{x}_2$:

$$k_2(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(\frac{-\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2} \right). \tag{3.19}$$

3.3. Concepts Related to the Characteristic Vector

In this section, we compare the proposed classifiers with several approaches related to the characteristic vector. It is worth noting that the major advantage of our classifiers lies in a trainable mechanism, which tries to optimally exploit useful information from background models, rather than make an ad hoc modification or use a combination of existing approaches.

3.3.1. Direct Fusion of Multiple LRs

The most intuitive way to improve the conventional LR-based speaker verification method would be to fuse multiple LR measures directly. Similar to the fusion approaches in [Ben-Yacoub 1999; Cheng 2005], we define a fusion-based LR as

$$L_{\text{Fusion}}(U) = w_{\text{UBM}}L_{\text{UBM}}(U) + w_{\text{Max}}L_{\text{Max}}(U) + w_{\text{Ari}}L_{\text{Ari}}(U) + w_{\text{Geo}}L_{\text{Geo}}(U) = \mathbf{w}'\mathbf{x} \begin{cases} \geq \theta & \text{accept} \\ < \theta & \text{reject,} \end{cases} \tag{3.20}$$

where $\mathbf{w} = [w_{\text{UBM}}\ w_{\text{Max}}\ w_{\text{Ari}}\ w_{\text{Geo}}]'$, and

$$\mathbf{x} = [L_{\text{UBM}}(U)\ \ L_{\text{Max}}(U)\ \ L_{\text{Ari}}(U)\ \ L_{\text{Geo}}(U)]'. \tag{3.21}$$

As with WGC and WAC, the weight vector $\mathbf{w}$ can be trained using KFD or SVM. A preliminary result reported in [Chao 2006] shows that, compared to approaches that use a single LR, such a fusion scheme improves speaker verification performance noticeably. However, we found that direct fusion is often dominated by one particular LR, or it is limited by some inferior LRs.

3.3.2. Relation to the Anchor Modeling Approach

The concept of our methods is similar to that of the anchor modeling approach [Sturim 2001; Mami 2006] used in speaker indexing and speaker identification applications. The objective of the anchor modeling approach is to construct a speaker space based on a set of pre-trained representative models $\{A_1, A_2, \ldots, A_N\}$, called *anchor models.* Then, any speech utterance U can be projected into the space, and represented as a characteristic vector **x** [Sturim 2001],

$$\mathrm{x} = [p(U\,|A_1)\;\; p(U\,|A_2)\;\; \ldots\;\; p(U\,|A_N)]'. \tag{3.22}$$

The speaker of an unknown utterance U can be identified by computing the distance between the characteristic vector **x** and the typical vectors of the target speakers. The characteristic vector defined in Eq. (3.22) is similar to the characteristic vector used in this study. However, to find the location of a target speaker in the speaker space, the anchor modeling approach only considers the projection of the speech utterance from the target speaker, which is different from the proposed discriminative framework. More specifically, the decision functions based on WGC and WAC characterize a target speaker by locating the boundary that optimally separates the characteristic vectors of a target speaker from those of non-target speakers; hence, the proposed methods are expected to be more effective than the anchor modeling approach.

3.4. Experiments and Analysis

We conducted the speaker-verification experiments on two databases: the XM2VTSDB and the ISCSLP2006 speaker recognition evaluation (ISCSLP2006-SRE) database [Zheng 2006].

3.4.1. Evaluation on the XM2VTSDB

The first set of experiments was conducted on XM2VTSDB following Configuration II. We built the world model with 256 Gaussian mixture components. The cohort size B was set to 20. The remaining experiment setup was same as that in Section 2.4.1. Because a kernel-based technique can be intractable when a large number of training samples are involved, we also reduced the number of evaluation-impostor samples from 119,400 to 2,250 for estimating **w**.

A. Weighted Geometric Combination versus Geometric Mean

The first experiment evaluated the proposed weighted geometric combination of background models, i.e., $L_{\mathrm{WGC}}(U)$ defined in Eq. (3.3). The set of background models was comprised of (i) the world model and the 20 closest cohort models (“w_20c”), or (ii) the world model and the 10 closest cohort models, plus the 10 farthest cohort models (“w_10c_10f”). The weight vector was optimized by kernel-based discrimination solutions (KFD or SVM). We derived the following eight WGC-based systems:

a) KFD with $k_1(\cdot)$ defined in Eq. (3.17) and “w_20c” (“WGC_dot_KFD_w_20c”),

b) KFD with $k_1(\cdot)$ defined in Eq. (3.17) and “w_10c_10f” (“WGC_dot_KFD_w_10c_10f”),

c) SVM with $k_1(\cdot)$ defined in Eq. (3.17) and “w_20c” (“WGC_dot_SVM_w_20c”),

d) SVM with $k_1(\cdot)$ defined in Eq. (3.17) and “w_10c_10f” (“WGC_dot_SVM_w_10c_10f”),

e) KFD with $k_2(\cdot)$ defined in Eq. (3.19) and “w_20c” (“WGC_RBF_KFD_w_20c”),

f) KFD with $k_2(\cdot)$ defined in Eq. (3.19) and “w_10c_10f” (“WGC_RBF_KFD_w_10c_10f”),

g) SVM with $k_2(\cdot)$ defined in Eq. (3.19) and “w_20c” (“WGC_RBF_SVM_w_20c”), and

h) SVM with $k_2(\cdot)$ defined in Eq. (3.19) and “w_10c_10f” (“WGC_RBF_SVM_w_10c_10f”).

Both SVM and KFD used an RBF kernel function $k_2(\cdot)$ with $\sigma = 5$. We used the SSVM tool [Lee 2001] to implement the SVM experiments, where the parameter C_β of SVM was set to 1.

For the performance comparison, we used three systems as our baselines:

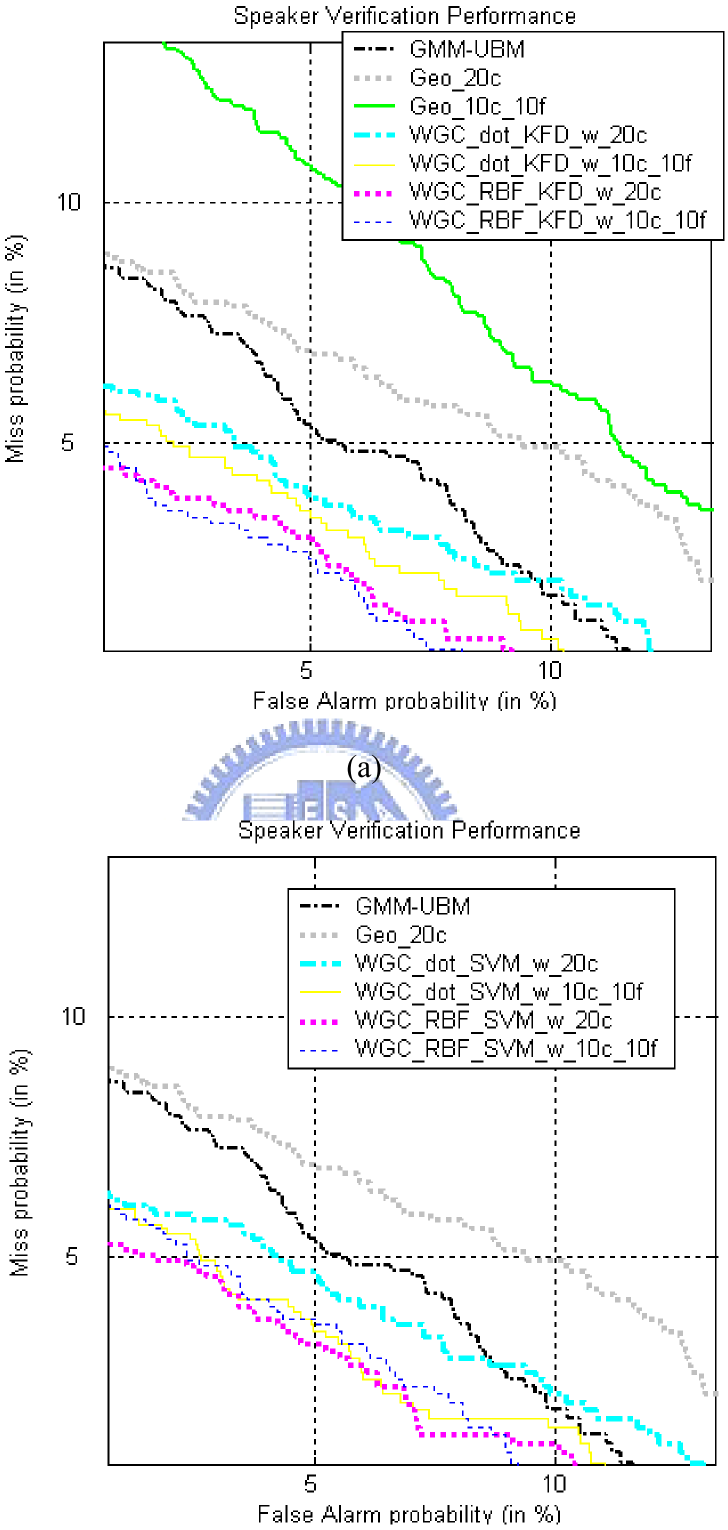
a) $L_{\text{UBM}}(U)$ (“GMM-UBM”),

b) $L_{\text{Geo}}(U)$ with the 20 closest cohort models (“Geo_20c”), and

c) $L_{\text{Geo}}(U)$ with the 10 closest cohort models plus the 10 farthest cohort models (“Geo_10c_10f”).

Fig. 3.1 shows the speaker verification results of the above systems evaluated on the XM2VTSDB “Test” subset in terms of Detection Error Tradeoff (DET) curves [Martin 1997]. Figures 3.1(a) and 3.1(b) compare the DET curves derived by KFD-based systems and SVM-based systems, respectively.

From Fig. 3.1, we observe that all the WGC-based systems with kernel functions $k_1(\cdot)$ or $k_2(\cdot)$ outperform the baseline systems “GMM-UBM”, “Geo_20c”, and “Geo_10c_10f”. We also observe that “Geo_10c_10f” in Fig. 3.1(a) yields the poorest performance. In addition, both Fig. 3.1(a) and Fig. 3.1(b) show that the WGC-based systems with $k_2(\cdot)$ outperform the WGC-based systems with $k_1(\cdot)$. Thus, in the subsequent experiments, we focused on investigating the performance achieved by the kernel-based discrimination solutions using the kernel function $k_2(\cdot)$.



(a)

(b)

Fig. 3.1. Geometric Mean versus WGC: DET curves for the “Test” subset in XM2VTSDB.

B. Weighted Arithmetic Combination versus Arithmetic Mean

The second experiment evaluated the proposed weighted arithmetic combination of background models, i.e., $L_{\mathrm{WAC}}(U)$ defined in Eq. (3.6). We implemented the WAC-based systems using the kernel-based discrimination solution in four ways:

a) KFD with “w_20c” (“WAC_RBF_KFD_w_20c”),

b) KFD with “w_10c_10f’ (“WAC_RBF_KFD_w_10c_10f”),

c) SVM with “w_20c” (“WAC_RBF_SVM_w_20c”), and

d) SVM with “w_10c_10f’ (“WAC_RBF_SVM_w_10c_10f’).

In the above cases, SVM and KFD used an RBF kernel function $k_2(\cdot)$ with $\sigma = 60$. For the performance comparison, we used three systems as our baselines:

a) $L_{\mathrm{UBM}}(U)$ (“GMM-UBM”),

b) $L_{\mathrm{Ari}}(U)$ with the 20 closest cohort models (“Ari_20c”), and

c) $L_{\mathrm{Ari}}(U)$ with the 10 closest cohort models plus the 10 farthest cohort models (“Ari_10c_10f’).

Fig. 3.2 shows the results of the above systems evaluated on the XM2VTSDB “Test” subset in terms of DET curves. Clearly, all the WAC-based systems based on either KFD or SVM outperform the baseline systems “GMM-UBM”, “Ari_20c”, and “Ari_10c_10f’. We also observe that the performances of SVM and KFD are similar.

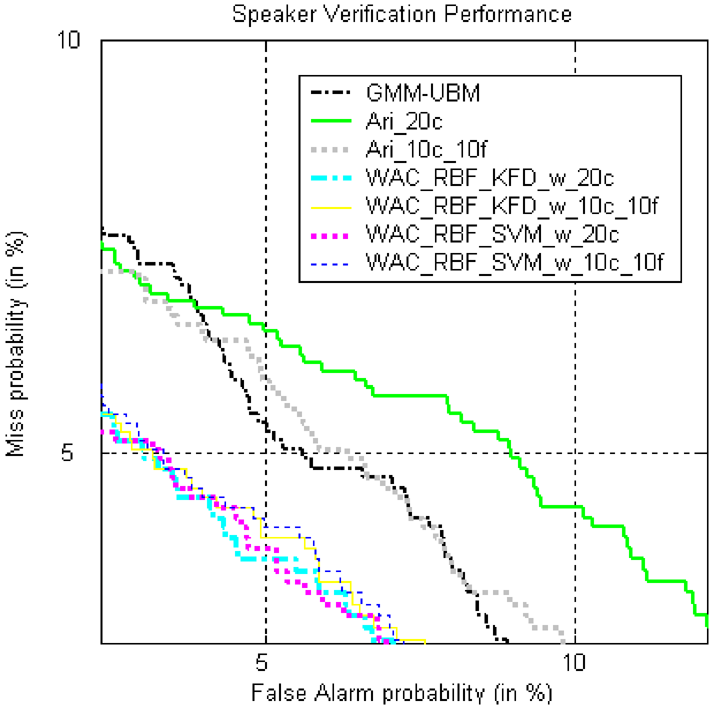


Fig. 3.2. Arithmetic Mean versus WAC: DET curves for the "Test" subset in XM2VTSDB.

C. Discussion

An analysis of the experiment results based on the DCF with $C_{Miss} = 1$, $C_{Fa} = 1$, and $P_{Target} = 0.5$ is given in Table 3.1. In addition to the above systems, we evaluated four related systems:

a) $L_{\mathrm{Max}}(U)$ with the 20 closest cohort models ("Max_20c");

b) $L_{\mathrm{Bengio}}(U)$ using an RBF kernel function with σ = 10 ("GMM-UBM/SVM");

c) $L_{\mathrm{Fusion}}(U)$ with a fusion of five baseline LR measures, namely, "GMM-UBM", "Max_20c", "Ari_20c", "Ari_10c_10f", and "Geo_20c", by KFD ("Fusion_KFD"); and

d) $L_{\mathrm{Fusion}}(U)$ with a fusion of five baseline LR measures, namely, "GMM-UBM", "Max_20c", "Ari_20c", "Ari_10c_10f", and "Geo_20c", by SVM ("Fusion_SVM").

In the fusion systems, KFD and SVM used an RBF kernel function with σ = 5. For each approach, the decision threshold was carefully tuned to minimize the DCF using the “Evaluation” subset, and then applied to the “Test” subset.

Table 3.1. DCFs for the “Evaluation” and “Test” subsets in the XM2VTS database

System	min DCF for “Evaluation”	actual DCF for “Test”
GMM-UBM	0.0633	0.0519
Max_20c	0.0776	0.0635
Ari_20c	0.0676	0.0535
Ari_10c_10f	0.0589	0.0515
Geo_20c	0.0734	0.0583
GMM-UBM/SVM	0.0590	0.0508
Fusion_KFD	0.0496	0.0475
Fusion_SVM	0.0505	0.0469
WGC_RBF_KFD_w_20c	0.0247	0.0357
WGC_RBF_KFD_w_10c_10f	0.0232	0.0389
WGC_RBF_SVM_w_20c	0.0320	0.0414
WGC_RBF_SVM_w_10c_10f	0.0310	0.0417
WAC_RBF_KFD_w_20c	0.0462	0.0443
WAC_RBF_KFD_w_10c_10f	0.0469	0.0445
WAC_RBF_SVM_w_20c	0.0460	0.0454
WAC_RBF_SVM_w_10c_10f	0.0479	0.0450

Several conclusions can be drawn from Table 3.1. First, the two direct fusion systems, “Fusion_KFD” and “Fusion_SVM”, as well as “GMM-UBM/SVM”, outperform the baseline LR systems. Second, the proposed WGC- and WAC-based systems not only outperform all the baseline LR systems, “GMM-UBM”, “Max_20c”, “Ari_20c”, “Ari_10c_10f”, and “Geo_20c”, they are also better than the fusion systems and the “GMM-UBM/SVM” system. The WGC- and WAC-based SVM systems are better than the “GMM-UBM/SVM” system because they consider multiple background models (including the world model), whereas the “GMM-UBM/SVM” system only considers the world model. Third, the WGC-based systems slightly outperform the WAC-based systems. Fourth, both KFD and SVM perform well in terms of finding nonlinear discrimination solutions. From the actual DCF for the “Test” subset, we observe that “WGC_RBF_KFD_w_20c” achieved a 30.68% relative improvement compared to “Ari_10c_10f” – the best baseline LR system. Table 3.2 compares the correlation of correct and incorrect decisions between “WGC_RBF_KFD_w_20c” and “Ari_10c_10f” for the actual DCF [Van Leeuwen 2006]. Based on McNemar’s test [Gillick 1989] with a significance level = 0.001, we can conclude that “WGC_RBF_KFD_w_20c” performs significantly better than “Ari_10c_10f”, since the resulting *P*-value < 0.001.

Table 3.2. Comparison of errors made by “WGC_RBF_KFD_w_20c” and “Ari_10c_10f,” where P and N denote the number of positive (target speaker) trials and the number of negative (impostor) trials, respectively. There are 1,194 P and 329,544 N in total.

Trial counts		Ari_10c_10f	
		Correct	Incorrect
WGC_RBF_KFD_w_20c	Correct	1,107P + 315,200N	32P + 6,019N
	Incorrect	5P + 3,056N	50P + 5,269N

3.4.2. Evaluation on the ISCSLP2006-SRE Database

We also evaluated the proposed methods on a text-independent single-channel speaker verification task conforming to the ISCSLP2006 Speaker Recognition Evaluation (ISCSLP2006-SRE) Plan [Chinese Corpus Consortium 2006]. Unlike the XM2VTSDB task, the ISCSLP2006-SRE database was divided into two subsets: a “Development Data Set” and an “Evaluation Data Set”. The “Development Data Set” contained 300 speakers. Each speaker made two utterances, each of which was cut into one long segment, which was longer than 30 seconds, and several short segments. In the experiments, we collected each speaker’s two long segments to build a UBM with 1,024 Gaussian mixture components, and used the two long segments per speaker to train each speaker’s 1024-mixture GMM through UBM-MAP adaptation. For each speaker, *B* speakers’ GMMs were chosen from the other 299 speakers as the cohort models. The remaining short segments of all the speakers were used to estimate θ, $\mathbf{w}$, and w_0. In the implementation, each short segment served as a positive sample for its associated speaker, but acted as a negative sample for each of the 20 randomly-selected speakers from the remaining 299 speakers. This yielded 1,551 positive samples and 31,020 (1,551×20) negative samples for estimating θ or w_0. Moreover, we used 1,551 positive samples and 1,551 randomly-selected negative samples to estimate $\mathbf{w}$ in the proposed systems.

The "Evaluation Data Set" contained 800 target speakers that did not overlap with the speakers in the "Development Data Set". Each target speaker made one long training utterance, ranging in duration from 21 to 85 seconds, with an average length of 37.06 seconds. This was used to generate the speaker's 1024-mixture GMM through UBM-MAP adaptation. For each target speaker, B speakers' GMMs were chosen from the 300 speakers in the "Development Data Set" as the cohort models. In addition, there were 5,933 test utterances (trials) in the "Evaluation Data Set", each of which ranged in duration from 5 seconds to 54 seconds, with an average length of 15.66 seconds. Each test utterance was associated with the claimed speaker's ID, and the task involved judging whether it was true or false. The answer sheet was released after the evaluation finished.

The acoustic feature extraction process was same as that applied in the XM2VTSDB task.

A. Experiment results

The GMM-UBM and T-norm systems are the current state-of-the-art approaches for the text-independent speaker verification task. Thus, in this part, we focus on the performance improvement of our methods over these two baseline systems. As with the GMM-UBM system, we used the fast scoring method [Reynolds 2000] for likelihood ratio computation in the proposed methods. Both the target speaker model λ and the B cohort models were adapted from the UBM Ω. Because the mixture indices were retained after UBM-MAP adaptation, each element of the characteristic vector $\mathbf{x}$ was computed approximately by only considering the C mixture components corresponding to the top C scoring mixtures in the UBM [Reynolds 2000]. In our experiments, C was set to 5, and B was set to 20.

The experiment results of the XM2VTSDB task showed that there was no significant performance difference between the two cohort selection methods used to construct the characteristic vector **x.** Thus, in the following experiments, we only used one type of characteristic vector, i.e., the vector associated with the UBM and the 20 closest cohort models ("w_20c"), to compute WGC- and WAC-based decision functions. This yielded the following four systems:

a) $L_{\text{WGC}}(U)$ using SVM with $k_2(\cdot)$ and "w_20c" ("WGC_RBF_SVM_w_20c"),

b) $L_{\text{WGC}}(U)$ using KFD with $k_2(\cdot)$ and "w_20c" ("WGC_RBF_KFD_w_20c"),

c) $L_{\text{WAC}}(U)$ using SVM with $k_2(\cdot)$ and "w_20c" ("WAC_RBF_SVM_w_20c"), and

d) $L_{\text{WAC}}(U)$ using KFD with $k_2(\cdot)$ and "w_20c" ("WAC_RBF_KFD_w_20c").

We compared the proposed systems with the GMM-UBM system, the T-norm system with the 50 closest cohort models ("Tnorm_50c"), and Bengio et al.'s system ("GMM-UBM/SVM"). The kernel parameters for SVM and KFD were same as those used in the XM2VTSDB task. Following the ISCSLP2006-SRE Plan, the performance was measured by the DCF with $C_{Miss} = 10$, $C_{Fa} = 1$, and $P_{Target} = 0.05$. In each system, the decision threshold was tuned to minimize the DCF using the (1,551 + 31,020) samples in the "Development Data Set", and then applied to the "Evaluation Data Set". Table 3.3 summarizes the minimum DCFs and the actual DCFs derived from 5,933 trials in the "Evaluation Data Set", and Fig. 3.3 shows the experiment results for all systems in terms of

DET curves. It is clear that all the proposed systems outperform “GMM-UBM”, “Tnorm_50c”, and “GMM-UBM/SVM.” The actual DCFs in Table 3.3 show that “WGC_RBF_KFD_w_20c” achieved a 52.72% relative improvement over “Tnorm_50c”. Table 3.4 compares the correlation of correct and incorrect decisions between “WGC_RBF_KFD_w_20c” and “Tnorm_50c” for the actual DCF. Based on McNemar’s test with a significance level = 0.001, we can conclude that “WGC_RBF_KFD_w_20c” performs significantly better than “Tnorm_50c”, since the resulting *P*-value < 0.001.

Table 3.3. Minimum DCFs and actual DCFs for the ISCSLP2006-SRE “Evaluation Data Set”

	Minimum DCFs	Actual DCFs
GMM-UBM	0.0184	0.0228
Tnorm_50c	0.0151	0.0184
GMM-UBM/SVM	0.0143	0.0146
WGC_RBF_KFD_w_20c	0.0081	0.0087
WAC_RBF_KFD_w_20c	0.0087	0.0112
WGC_RBF_SVM_w_20c	0.0091	0.0105
WAC_RBF_SVM_w_20c	0.0093	0.0105

Table 3.4. Comparison of errors made by “WGC_RBF_KFD_w_20c” and “ Tnorm_50c”, where P and N denote the number of positive (target speaker) trials and the number of negative (impostor) trials, respectively. There are 347 P and 5,586 N in total.

Trial counts		Tnorm_50c	
		Correct	Incorrect
WGC_RBF_KFD_w_20c	Correct	342P + 5,508N	2P + 52N
	Incorrect	0P + 12N	3P + 14N

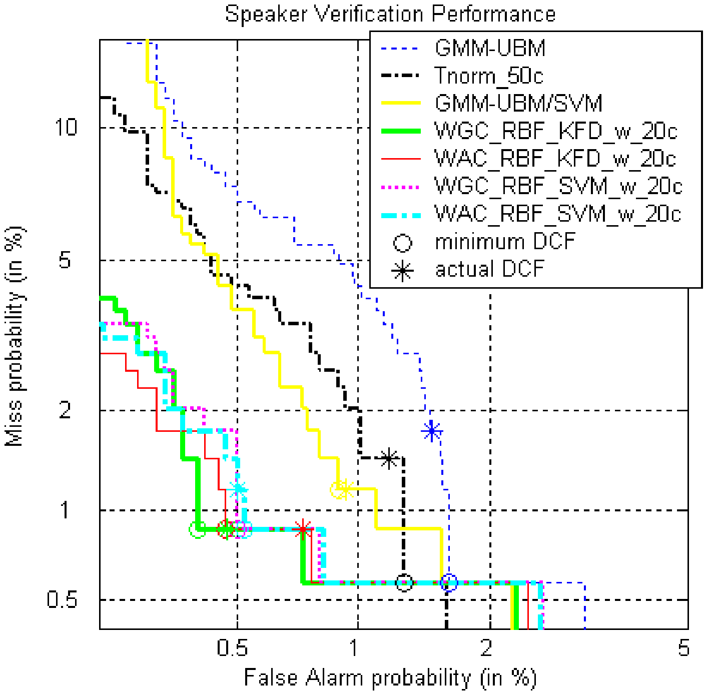


Fig. 3.3. Baseline systems versus WAC and WGC: DET curves for the ISCSLP2006-SRE "Evaluation Data Set". The stars and circles indicate the actual and minimum DCFs, respectively.

Chapter 4

Improving GMM-UBM Speaker Verification Using Discriminative Feedback Adaptation

In this chapter, we focus on the discussion of the current state-of-the-art GMM-UBM approach [Reynolds 2000] for text-independent speaker verification that uses the UBM-MAP technique to generate the target model λ and the anti-model $\bar{\lambda}$. This approach pools all speech data from a large number of background speakers to form a universal background model (UBM) as $\bar{\lambda}$ via the expectation-maximization (EM) algorithm. It then adapts the UBM to λ via the maximum a posteriori (MAP) estimation technique. GMM-UBM is effective because its generalization ability allows λ to handle acoustic patterns not covered by the limited training data of the target speaker. However, since λ and $\bar{\lambda}$ are trained according to separate criteria, the optimization procedure can not distinguish a target speaker from background speakers optimally. In particular, since GMM-UBM uses a common UBM $\bar{\lambda}$ for all target speakers, it tends to be weak in rejecting impostors' voices that are similar to the target speaker's voice. Moreover, as λ is derived from $\bar{\lambda}$, both models may correspond to a similar probability distribution.

One possible way to improve the performance of GMM-UBM is to use discriminative training methods, such as the minimum classification error (MCE) method [Juang 1997] and the maximum mutual information (MMI) method [Ma 2003]. In [Rosenberg 1998], a minimum verification error (MVE) training method is developed by adapting MCE training to the binary classification problem, in which the parameters of λ and $\bar{\lambda}$ are estimated using the generalized probabilistic descent (GPD) approach [Chou 2003]. However, as the MVE training method requires a large number of positive and negative samples to estimate a model's parameters, it tends to over-train the model if the amount of training data is insufficient. In addition, it is difficult to select the optimal stopping point in GPD-based training.

To resolve the limitation of MVE training, we propose a framework called discriminative feedback adaptation (DFA), which improves the discrimination ability of GMM-UBM while preserving its generalization ability. The rationale behind DFA is that only mis-verified training samples are considered in the discriminative training process, rather than all the training samples used in the conventional MVE method. More specifically, DFA regards the UBM and the target speaker model obtained by the GMM-UBM approach as initial models, and then reinforces the discriminability between the models by using the mis-verified training samples. Since the reinforcement is based on model adaptation rather than training from scratch, it does not destroy the generalization ability of the two models, even if they are updated iteratively until convergence. However, recognizing that a small number of mis-verified training samples may not be able to adapt a large number of model parameters, to implement DFA, we propose two adaptation techniques: a linear regression-based minimum verification squared-error (LR-MVSE) adaptation method and an eigenspace-based minimum verification squared-error (E-MVSE) adaptation method. LR-MVSE is motivated by the minimum classification error linear regression (MCELR) techniques [Chengalvarayan 1998; Wu 2002; He 2003], which have been studied in the context of automatic speech recognition; while E-MVSE is motivated by the MCE/eigenvoice technique [Valente 2003], which has been studied in the context of speaker identification.

The remainder of this chapter is organized as follows. In Section 4.1, we introduce the proposed DFA framework. Sections 4.2 and 4.3 describe, respectively, the proposed LR-MVSE and E-MVSE adaptation techniques used to implement DFA. Section 4.4 presents simplified versions of LR-MVSE and E-MVSE. Then, in Section 4.5, we detail the experiment results.

4.1. Discriminative Feedback Adaptation

Fig. 4.1 shows a block diagram of the proposed discriminative feedback adaptation (DFA) framework, which is divided into two phases. The first phase, indicated by the dotted line, utilizes the conventional GMM-UBM approach. The initial target speaker model and the UBM obtained in the first phase serve as the initial models for DFA in the second phase. The basic strategy of DFA is to reinforce the discriminability between the initial target speaker model and the UBM for ambiguous data that is mis-verified by the GMM-UBM approach. The reinforcement strategy is based on two concepts. First, since the GMM-UBM approach uses a single anti-model, UBM, for all target speakers, it tends to be weak in rejecting impostors' voices that are similar to the target speaker's voice. To resolve this problem, DFA tries to generate a discriminative anti-model exclusively for each target speaker by using the negative samples from the cohort [Rosenberg 1992] of each target speaker to adapt both λ and $\bar{\lambda}$. Since the models may affect each other, the DFA framework also uses the positive samples to avoid increasing the miss probability while reducing the false alarm probability. The resulting λ and

$\bar{\lambda}$ are then updated iteratively. Second, since the DFA framework only uses mis-verified training samples as adaptation data in each iteration, it actually fine-tunes the model's parameters based on a small amount of adaptation data. It thus preserves the generalization ability of the GMM-UBM approach while reinforcing the discrimination between H_0 and H_1. To implement the above concepts, we developed the following algorithms.

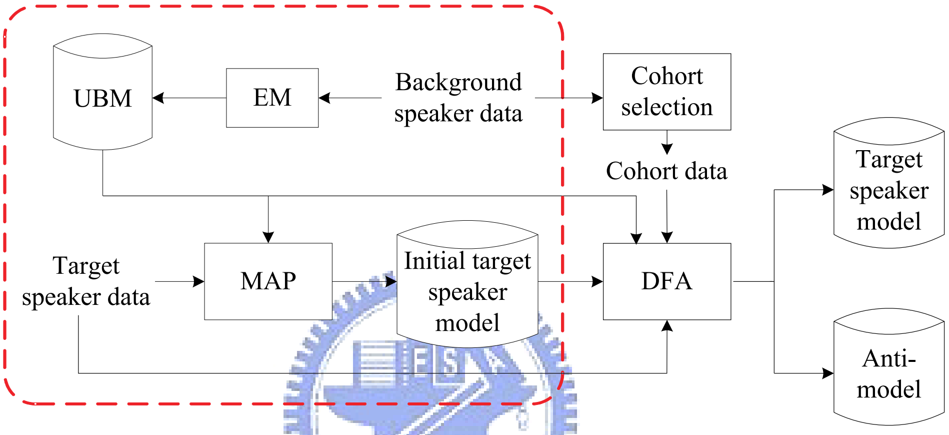


Fig. 4.1. The proposed discriminative feedback adaptation framework.

4.1.1. Minimum Verification Squared-Error (MVSE) adaptation strategy

We modify the minimum verification error (MVE) training method [Rosenberg 1998] to fit our requirement that only mis-verified training samples should be considered. This is called the minimum verification squared-error (MVSE) adaptation strategy. The goal of DFA is to minimize the overall expected loss *D*, defined as

$$D = x_0 \ell_0 + x_1 \ell_1, \tag{4.1}$$

where x_0 and x_1 reflect which type of error is of more concern in a practical application; and ℓ_i is a loss function that describes the average false rejection loss (i = 0) or false acceptance loss

(i = 1), defined as

$$\ell_i = \frac{1}{N_i} \sum_{U \in H_i} s(d(U)), \tag{4.2}$$

where N_0 and N_1 are the numbers of training utterances from the target speaker and the cohort, respectively; and $d(U)$ is a mis-verification measure defined as

$$d(U) = \begin{cases} -L(U) & \text{if } U \in H_0 \\ L(U) & \text{if } U \in H_1 , \end{cases} \tag{4.3}$$

where $L(U)$ is the logarithmic LR defined as

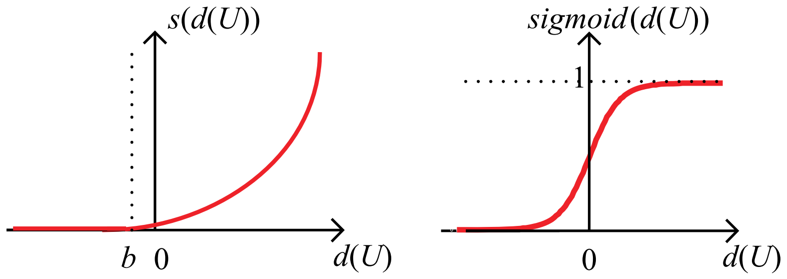
$$L(U) = \log p(U \mid \lambda) - \log p(U \mid \bar{\lambda}), \tag{4.4}$$

where λ is the target speaker model; and $\bar{\lambda}$ is the anti-model.

To reflect the requirement that only mis-verified training utterances should be considered, we define a new function $s(\cdot)$ instead of the sigmoid function used in the function ℓ_i, which represents the verification error as an adjustable quantity as follows:

$$s(d(U)) = \begin{cases} a(d(U)-b)^2 & \text{if } d(U) > b \\ 0 & \text{if } d(U) \le b, \end{cases} \tag{4.5}$$

where a is a scalar and b is a bias for controlling the convergence speed of DFA. The input utterance U is considered incorrectly verified if $d(U) > b$. Therefore, $s(d(U))$ is a response squared-error value. Fig. 4.2 contrasts the curve of the s function with that of the well-known sigmoid function. If $d(U) \le b$, the response value $s(d(U)) = 0$, i.e., the utterance U is verified correctly; hence, it will not be used for model adaptation. If $d(U) > b$, the steeper slope of the s function for a larger value of $d(U)$ results in a larger gradient to update the model's parameters. In contrast, as the value of $d(U)$ increases, the sigmoid function used in MVE [Rosenberg 1998] will become flat, and the obtained gradient will approximate zero. As a result, the mis-verified utterance U will not contribute to model adaptation. Another difference between the proposed DFA framework and the conventional MVE training method is that the latter always updates the model's parameters if the value of the sigmoid function is not 0 or 1; thus, it may over-train the well-trained models obtained from the GMM-UBM method with the correctly-verified input training utterances.



(a) *s* function

(b) sigmoid function

Fig. 4.2. The *s* function compared to the sigmoid function.

4.1.2. Fast scoring for DFA

To speed up DFA, we use a fast scoring approach [Reynolds 2000] to compute the logarithmic LR. Given an utterance $U = \{o_1, \ldots, o_T\}$, the computation of the logarithmic LR for a GMM with *M* Gaussian mixture components can be written as

$$\begin{aligned} L(U) &= \frac{1}{T}\sum_{t=1}^{T}\left(\log\sum_{m=1}^{M} p_m p(o_t \mid \mathbf{g}_m) - \log\sum_{m=1}^{M} p_m p(o_t \mid \bar{\mathbf{g}}_m)\right) \\ &\approx \frac{1}{T}\sum_{t=1}^{T}\left(\log\sum_{i=1}^{C} p_{C_i(t)} p(o_t \mid \mathbf{g}_{C_i(t)}) - \log\sum_{i=1}^{C} p_{C_i(t)} p(o_t \mid \bar{\mathbf{g}}_{C_i(t)})\right), \end{aligned} \tag{4.6}$$

where $\mathbf{g}_m$ and $\bar{\mathbf{g}}_m$ are the *m*-th Gaussian mixture components of the target speaker model and the anti-model, respectively; and p_m is the mixture weight, *m* = 1,…, *M*. Note that the target speaker model has the same mixture weights as the anti-model. For each frame o_t, we determine the top C scoring mixture indices, $C_i(t)$, i = 1,…, C, in the UBM, where $C << M$; hence, it requires $M + C$ Gaussian computations in the first iteration, and 2C Gaussian computations per iteration thereafter. In this study, the value of C is set at 5 [Reynolds 2000].

4.2. Linear regression-based MVSE (LR-MVSE) adaptation

Recognizing that a small amount of adaptation data selected from the mis-verified training samples may not be able to adapt a large number of model parameters, we propose using a linear regression method to implement MVSE adaptation. We call it linear regression-based MVSE (LR-MVSE) adaptation. Our strategy is motivated by the minimum classification error linear regression (MCELR) techniques [Chengalvarayan 1998; Wu 2002; He 2003], which have been studied in the context of automatic speech recognition. We assume that the initial target speaker model $\lambda^{(0)}$ and anti-model $\bar{\lambda}^{(0)}$ have M Gaussian mixtures $\mathbf{g}_m^{(0)} \sim N(\boldsymbol{\mu}_m^{(0)}, \boldsymbol{\Sigma}_m)$ and $\bar{\mathbf{g}}_m^{(0)} \sim N(\bar{\boldsymbol{\mu}}_m^{(0)}, \boldsymbol{\Sigma}_m)$, respectively, where $\boldsymbol{\mu}_m^{(0)}$ and $\bar{\boldsymbol{\mu}}_m^{(0)}$ are r–dimensional mean vectors obtained with the GMM-UBM method; and $\boldsymbol{\Sigma}_m$ is an r×r covariance matrix of the UBM, m = 1,…, M. Note that, in this study, we only adapt the mean vectors of GMMs. After adaptation, the new mean vectors of the target speaker model and the anti-model take the following respective forms:

$$\boldsymbol{\mu}_m = \mathbf{W}\boldsymbol{\xi}_m^{(0)} \tag{4.7}$$

and

$$\bar{\boldsymbol{\mu}}_m = \bar{\mathbf{W}}\bar{\boldsymbol{\xi}}_m^{(0)}, \tag{4.8}$$

where $\mathbf{W}$ and $\bar{\mathbf{W}}$ are r×(r+1) transformation matrices; and $\boldsymbol{\xi}_m^{(0)} = [1\ \boldsymbol{\mu}_{\mathrm{m}}^{(0)\prime}]'$ and

$\bar{\boldsymbol{\xi}}_m^{(0)} = [1\ \bar{\boldsymbol{\mu}}_{\mathrm{m}}^{(0)\prime}]'$. Given initial transformation matrices $\mathbf{W}^{(0)} = \overline{\mathbf{W}}^{(0)} = [\mathbf{0}\ \mathbf{I}]$, where $\mathbf{0}$ is an $r\times 1$ zero vector and $\mathbf{I}$ is an $r\times r$ identity matrix, the parameters $\mathbf{W}$ and $\overline{\mathbf{W}}$ can be iteratively optimized using

$$\mathbf{W}^{(k+1)} = \mathbf{W}^{(k)} - \delta \frac{\partial D}{\partial \mathbf{W}^{(k)}} \tag{4.9}$$

and

$$\overline{\mathbf{W}}^{(k+1)} = \overline{\mathbf{W}}^{(k)} - \delta \frac{\partial D}{\partial \overline{\mathbf{W}}^{(k)}}, \tag{4.10}$$

respectively, where the superscript "(k)" denotes the k-th iteration, and δ is the step size. In addition,

$$\begin{aligned}\frac{\partial D}{\partial \mathbf{W}^{(k)}} &= x_0 \frac{\partial \ell_0}{\partial s(d(U))} \cdot \frac{\partial s(d(U))}{\partial d(U)} \cdot \frac{\partial d(U)}{\partial L(U)} \cdot \frac{\partial L(U)}{\partial \mathbf{W}^{(k)}} + x_1 \frac{\partial \ell_1}{\partial s(d(U))} \cdot \frac{\partial s(d(U))}{\partial d(U)} \cdot \frac{\partial d(U)}{\partial L(U)} \cdot \frac{\partial L(U)}{\partial \mathbf{W}^{(k)}} \\ &= x_0 \cdot \frac{1}{N_0} \sum_{U\in H_0, -L(U)>b} \left\{ 2a \cdot (-L(U) - b) \cdot \left(-\frac{\partial L(U)}{\partial \mathbf{W}^{(k)}} \right) \right\} \\ &\quad + x_1 \cdot \frac{1}{N_1} \sum_{U\in H_1, L(U)>b} \left\{ 2a \cdot (L(U) - b) \cdot \frac{\partial L(U)}{\partial \mathbf{W}^{(k)}} \right\}\end{aligned} \tag{4.11}$$

and

$$\begin{aligned}\frac{\partial D}{\partial \overline{\mathbf{W}}^{(k)}} &= x_0 \frac{\partial \ell_0}{\partial s(d(U))} \cdot \frac{\partial s(d(U))}{\partial d(U)} \cdot \frac{\partial d(U)}{\partial L(U)} \cdot \frac{\partial L(U)}{\partial \overline{\mathbf{W}}^{(k)}} + x_1 \frac{\partial \ell_1}{\partial s(d(U))} \cdot \frac{\partial s(d(U))}{\partial d(U)} \cdot \frac{\partial d(U)}{\partial L(U)} \cdot \frac{\partial L(U)}{\partial \overline{\mathbf{W}}^{(k)}} \\ &= x_0 \cdot \frac{1}{N_0} \sum_{U\in H_0, -L(U)>b} \left\{ 2a \cdot (-L(U) - b) \cdot \left(-\frac{\partial L(U)}{\partial \overline{\mathbf{W}}^{(k)}} \right) \right\} \\ &\quad + x_1 \cdot \frac{1}{N_1} \sum_{U\in H_1, L(U)>b} \left\{ 2a \cdot (L(U) - b) \cdot \frac{\partial L(U)}{\partial \overline{\mathbf{W}}^{(k)}} \right\},\end{aligned} \tag{4.12}$$

where

$$\frac{\partial L(U)}{\partial \mathbf{W}^{(k)}} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{p(o_t \mid \lambda^{(k)})} \left(\sum_{i=1}^{C} p_{C_i(t)} \frac{\partial p(o_t \mid \mathbf{g}_{C_i(t)}^{(k)})}{\partial \mathbf{W}^{(k)}} \right) \tag{4.13}$$

and

$$\frac{\partial L(U)}{\partial \overline{\mathbf{W}}^{(k)}} = \frac{1}{T}\sum_{t=1}^{T}\frac{-1}{p(o_t \mid \overline{\lambda}^{(k)})}\left(\sum_{i=1}^{C} p_{C_i(t)}\frac{\partial p(o_t \mid \overline{\mathbf{g}}^{(k)}_{C_i(t)})}{\partial \overline{\mathbf{W}}^{(k)}}\right), \tag{4.14}$$

where the target speaker model $\lambda^{(k)}$ with mixtures $\mathbf{g}_m^{(k)}$ and the anti-model $\overline{\lambda}^{(k)}$ with mixtures $\overline{\mathbf{g}}_m^{(k)}$, $m = 1,\ldots, M$, are obtained by LR-MVSE adaptation in k iterations, and

$$\frac{\partial p(o_t \mid \mathbf{g}^{(k)}_{C_i(t)})}{\partial \mathbf{W}^{(k)}} = p(o_t \mid \mathbf{g}^{(k)}_{C_i(t)})\Sigma^{-1}_{C_i(t)}\left(o_t - \mathbf{W}^{(k)}\boldsymbol{\xi}^{(0)}_{C_i(t)}\right)\boldsymbol{\xi}^{(0)}_{C_i(t)}{}' \tag{4.15}$$

and

$$\frac{\partial p(o_t \mid \overline{\mathbf{g}}^{(k)}_{C_i(t)})}{\partial \overline{\mathbf{W}}^{(k)}} = p(o_t \mid \overline{\mathbf{g}}^{(k)}_{C_i(t)})\Sigma^{-1}_{C_i(t)}\left(o_t - \overline{\mathbf{W}}^{(k)}\overline{\boldsymbol{\xi}}^{(0)}_{C_i(t)}\right)\overline{\boldsymbol{\xi}}^{(0)}_{C_i(t)}{}'. \tag{4.16}$$

If we assume that all covariance matrices $\boldsymbol{\Sigma}_m$ of the UBM, $m = 1,\ldots, M$, are diagonal, Eqs. (4.15) and (4.16) can be rewritten as

$$\frac{\partial p(o_t \mid \mathbf{g}^{(k)}_{C_i(t)})}{\partial \mathbf{W}^{(k)}(r_1, r_2)} = \frac{p(o_t \mid \mathbf{g}^{(k)}_{C_i(t)})}{\sigma^2_{C_i(t)}(r_1)}\left(o_t(r_1) - \sum_{j=1}^{r+1}\mathbf{W}^{(k)}(r_1, j)\xi^{(0)}_{C_i(t)}(j)\right)\xi^{(0)}_{C_i(t)}(r_2) \tag{4.17}$$

and

$$\frac{\partial p(o_t \mid \overline{\mathbf{g}}^{(k)}_{C_i(t)})}{\partial \overline{\mathbf{W}}^{(k)}(r_1, r_2)} = \frac{p(o_t \mid \overline{\mathbf{g}}^{(k)}_{C_i(t)})}{\sigma^2_{C_i(t)}(r_1)}\left(o_t(r_1) - \sum_{j=1}^{r+1}\overline{\mathbf{W}}^{(k)}(r_1, j)\overline{\xi}^{(0)}_{C_i(t)}(j)\right)\overline{\xi}^{(0)}_{C_i(t)}(r_2), \tag{4.18}$$

respectively, where $\sigma^2_m(r_1)$ is the r_1-th diagonal element of $\boldsymbol{\Sigma}_m$; $o_t(r_1)$ is the r_1-th element of o_t; $\xi^{(0)}_m(r_2)$ and $\overline{\xi}^{(0)}_m(r_2)$ are, respectively, the r_2-th elements of $\boldsymbol{\xi}^{(0)}_m$ and $\overline{\boldsymbol{\xi}}^{(0)}_m$; and $\mathbf{W}^{(k)}(r_1, r_2)$ and $\overline{\mathbf{W}}^{(k)}(r_1, r_2)$ are, respectively, the r_1-th row and r_2-th column elements of $\mathbf{W}^{(k)}$ and $\overline{\mathbf{W}}^{(k)}$, $r_1 = 1,\ldots, r$, and $r_2 = 1,\ldots, (r+1)$.

4.3. Eigenspace-based MVSE (E-MVSE) adaptation

Alternatively, we can use the eigenspace method to implement MVSE adaptation. We call it eigenspace-based MVSE (E-MVSE) adaptation. E-MVSE is motivated by the MCE/eigenvoice technique [Valente 2003], which has been studied in the context of speaker identification. In this case, we also assume that only the mean vectors of GMMs are adapted. Let $\mathbf{u}^{(0)}$ and $\bar{\mathbf{u}}^{(0)}$ be $(rM)\times 1$ supervectors [Kuhn 2000; Thyes 2000] obtained by concatenating all the mean vectors of the initial target speaker model $\lambda^{(0)}$ and anti-model (a clone of the UBM) $\bar{\lambda}^{(0)}$, where

$$\mathbf{u}^{(0)} = [\boldsymbol{\mu}_1^{(0)\prime}\ \boldsymbol{\mu}_2^{(0)\prime} \ldots \boldsymbol{\mu}_M^{(0)\prime}]' \tag{4.19}$$

and

$$\bar{\mathbf{u}}^{(0)} = [\bar{\boldsymbol{\mu}}_1^{(0)\prime}\ \bar{\boldsymbol{\mu}}_2^{(0)\prime} \ldots \bar{\boldsymbol{\mu}}_M^{(0)\prime}]'. \tag{4.20}$$

Following the eigenvoice approach, we use the principal component analysis (PCA) technique [Duda 2001] to construct a speaker eigenspace $\mathbf{E}$ = span{$\mathbf{e}_1$, $\mathbf{e}_2$,…, $\mathbf{e}_Z$} based on R supervectors derived from R pre-trained background speaker GMMs, where $Z \leq R-1$. According to the orthogonality principle [Strang 2005], we can decompose $\mathbf{u}^{(0)}$ and $\bar{\mathbf{u}}^{(0)}$ into

$$\mathbf{u}^{(0)} = \boldsymbol{\eta} + \sum_{z=1}^{Z} f_z^{(0)}\mathbf{e}_z + f_{Z+1}^{(0)}\mathbf{e}^{\perp} \tag{4.21}$$

and

$$\bar{\mathbf{u}}^{(0)} = \boldsymbol{\eta} + \sum_{z=1}^{Z} \bar{f}_z^{(0)}\mathbf{e}_z + \bar{f}_{Z+1}^{(0)}\bar{\mathbf{e}}^{\perp}, \tag{4.22}$$

respectively, where $\boldsymbol{\eta}$ is the sample mean vector of R supervectors. The second terms in Eqs. (4.21) and (4.22) represent the results of projecting $(\mathbf{u}^{(0)} - \boldsymbol{\eta})$ and $(\bar{\mathbf{u}}^{(0)} - \boldsymbol{\eta})$ onto the eigenspace **E**. Note that, in most cases, $(\mathbf{u}^{(0)} - \boldsymbol{\eta})$ and $(\bar{\mathbf{u}}^{(0)} - \boldsymbol{\eta}) \notin \mathbf{E}$, since the initial target speaker model and anti-model are not included in the background speaker model set. The coordinates, $f_z^{(0)}$ and $\bar{f}_z^{(0)}$, $z = 1,.., Z$, are computed by

$$f_z^{(0)} = \mathbf{e}_z'(\mathbf{u}^{(0)} - \boldsymbol{\eta}) \tag{4.23}$$

and

$$\bar{f}_z^{(0)} = \mathbf{e}_z'(\bar{\mathbf{u}}^{(0)} - \boldsymbol{\eta}), \tag{4.24}$$

respectively. The third terms in Eqs. (4.21) and (4.22) represent the residuals after the projection. If the residuals are not zero, we can define $f_{Z+1}^{(0)}$ and $\bar{f}_{Z+1}^{(0)}$ as

$$f_{Z+1}^{(0)} = \left\| \mathbf{u}^{(0)} - \boldsymbol{\eta} - \sum_{z=1}^{Z} f_z^{(0)} \mathbf{e}_z \right\| \tag{4.25}$$

and

$$\bar{f}_{Z+1}^{(0)} = \left\| \bar{\mathbf{u}}^{(0)} - \boldsymbol{\eta} - \sum_{z=1}^{Z} \bar{f}_z^{(0)} \mathbf{e}_z \right\|, \tag{4.26}$$

and define $\mathbf{e}^{\perp}$ and $\bar{\mathbf{e}}^{\perp}$ as

$$\mathbf{e}^{\perp} = \frac{\mathbf{u}^{(0)} - \boldsymbol{\eta} - \sum_{z=1}^{Z} f_z^{(0)} \mathbf{e}_z}{\left\| \mathbf{u}^{(0)} - \boldsymbol{\eta} - \sum_{z=1}^{Z} f_z^{(0)} \mathbf{e}_z \right\|} \tag{4.27}$$

and

$$\bar{\mathbf{e}}^{\perp} = \frac{\bar{\mathbf{u}}^{(0)} - \boldsymbol{\eta} - \sum_{z=1}^{Z} \bar{f}_z^{(0)} \mathbf{e}_z}{\left\| \bar{\mathbf{u}}^{(0)} - \boldsymbol{\eta} - \sum_{z=1}^{Z} \bar{f}_z^{(0)} \mathbf{e}_z \right\|}. \tag{4.28}$$

Since both $\mathbf{e}^{\perp}$ and $\bar{\mathbf{e}}^{\perp}$ are orthogonal to **E**, $\mathbf{u}^{(0)}$ and $\bar{\mathbf{u}}^{(0)}$ can be represented, respectively, by the initial coordinates $[f_1^{(0)}\ \ f_2^{(0)} \ldots f_Z^{(0)}\ \ f_{Z+1}^{(0)}]'$ and $[\bar{f}_1^{(0)}\ \ \bar{f}_2^{(0)} \ldots \bar{f}_Z^{(0)}\ \ \bar{f}_{Z+1}^{(0)}]'$ in a target speaker space $\mathbf{E}_\lambda$ with an orthonormal basis {$\mathbf{e}_1$, $\mathbf{e}_2$,…, $\mathbf{e}_Z$, $\mathbf{e}^\perp$} and an anti-model space $\mathbf{E}_{\bar{\lambda}}$ with an orthonormal basis {$\mathbf{e}_1$, $\mathbf{e}_2$,…, $\mathbf{e}_Z$, $\bar{\mathbf{e}}^\perp$}. If $f_{Z+1}^{(0)} = 0$ and $\bar{f}_{Z+1}^{(0)} = 0$, both $\mathbf{e}^\perp$ and $\bar{\mathbf{e}}^\perp$ are zero vectors, and the (Z+1)-th weight is not included in a coordinate vector $\in \mathbf{E}_\lambda = \mathbf{E}_{\bar{\lambda}} = \mathbf{E}$ with a basis {$\mathbf{e}_1$, $\mathbf{e}_2$,…, $\mathbf{e}_Z$}. Our goal is to find the best coordinates $[f_1\ \ f_2 \ldots f_Z\ \ f_{Z+1}]'$ in $\mathbf{E}_\lambda$ and $[\bar{f}_1\ \ \bar{f}_2 \ldots \bar{f}_Z\ \ \bar{f}_{Z+1}]'$ in $\mathbf{E}_{\bar{\lambda}}$ such that the reconstructed models can optimally distinguish the target speaker's voice from the non-target speakers' voices. The reconstructed mean vectors of the target speaker model and the anti-model take the following respective forms:

$$\boldsymbol{\mu}_m = \boldsymbol{\eta}_m + \sum_{z=1}^{Z} f_z \mathbf{e}_{z,m} + f_{Z+1}\mathbf{e}_m^\perp \tag{4.29}$$

and

$$\bar{\boldsymbol{\mu}}_m = \boldsymbol{\eta}_m + \sum_{z=1}^{Z} \bar{f}_z \mathbf{e}_{z,m} + \bar{f}_{Z+1}\bar{\mathbf{e}}_m^\perp, \tag{4.30}$$

where $\boldsymbol{\eta}_m$, $\mathbf{e}_{z,m}$, $\mathbf{e}_m^\perp$, and $\bar{\mathbf{e}}_m^\perp$ represent the m-th subvectors of $\boldsymbol{\eta}$, $\mathbf{e}_z$, $\mathbf{e}^\perp$, and $\bar{\mathbf{e}}^\perp$, respectively, and correspond to the mean vector of the m-th Gaussian mixture component of the target speaker model and the anti-model, m = 1,…, M. The coordinates, f_z and $\bar{f}_z$, z = 1,.., Z+1, can be iteratively optimized using

$$f_z^{(k+1)} = f_z^{(k)} - \delta \frac{\partial D}{\partial f_z^{(k)}} \tag{4.31}$$

and

$$\bar{f}_z^{(k+1)} = \bar{f}_z^{(k)} - \delta \frac{\partial D}{\partial \bar{f}_z^{(k)}}, \tag{4.32}$$

respectively, where δ is the step size. In addition,

$$\begin{aligned}\frac{\partial D}{\partial f_z^{(k)}} &= x_0 \frac{\partial \ell_0}{\partial s(d(U))}\cdot\frac{\partial s(d(U))}{\partial d(U)}\cdot\frac{\partial d(U)}{\partial L(U)}\cdot\frac{\partial L(U)}{\partial f_z^{(k)}} + x_1 \frac{\partial \ell_1}{\partial s(d(U))}\cdot\frac{\partial s(d(U))}{\partial d(U)}\cdot\frac{\partial d(U)}{\partial L(U)}\cdot\frac{\partial L(U)}{\partial f_z^{(k)}} \\ &= x_0 \cdot \frac{1}{N_0} \sum_{U\in H_0,\, -L(U)>b} \left\{ 2a\cdot(-L(U)-b)\cdot\left(-\frac{\partial L(U)}{\partial f_z^{(k)}}\right)\right\} \\ &\quad + x_1 \cdot \frac{1}{N_1} \sum_{U\in H_1,\, L(U)>b} \left\{ 2a\cdot(L(U)-b)\cdot\frac{\partial L(U)}{\partial f_z^{(k)}}\right\}\end{aligned} \tag{4.33}$$

and

$$\begin{aligned}\frac{\partial D}{\partial \bar{f}_z^{(k)}} &= x_0 \frac{\partial \ell_0}{\partial s(d(U))}\cdot\frac{\partial s(d(U))}{\partial d(U)}\cdot\frac{\partial d(U)}{\partial L(U)}\cdot\frac{\partial L(U)}{\partial \bar{f}_z^{(k)}} + x_1 \frac{\partial \ell_1}{\partial s(d(U))}\cdot\frac{\partial s(d(U))}{\partial d(U)}\cdot\frac{\partial d(U)}{\partial L(U)}\cdot\frac{\partial L(U)}{\partial \bar{f}_z^{(k)}} \\ &= x_0 \cdot \frac{1}{N_0} \sum_{U\in H_0,\, -L(U)>b} \left\{ 2a\cdot(-L(U)-b)\cdot\left(-\frac{\partial L(U)}{\partial \bar{f}_z^{(k)}}\right)\right\} \\ &\quad + x_1 \cdot \frac{1}{N_1} \sum_{U\in H_1,\, L(U)>b} \left\{ 2a\cdot(L(U)-b)\cdot\frac{\partial L(U)}{\partial \bar{f}_z^{(k)}}\right\},\end{aligned} \tag{4.34}$$

where

$$\frac{\partial L(U)}{\partial f_z^{(k)}} = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{p(o_t \mid \lambda^{(k)})}\left(\sum_{i=1}^{C} \alpha_{C_i(t)} \frac{\partial p(o_t \mid \mathbf{g}_{C_i(t)}^{(k)})}{\partial f_z^{(k)}}\right) \tag{4.35}$$

and

$$\frac{\partial L(U)}{\partial \bar{f}_z^{(k)}} = \frac{1}{T}\sum_{t=1}^{T} \frac{-1}{p(o_t \mid \bar{\lambda}^{(k)})}\left(\sum_{i=1}^{C} \alpha_{C_i(t)} \frac{\partial p(o_t \mid \bar{\mathbf{g}}_{C_i(t)}^{(k)})}{\partial \bar{f}_z^{(k)}}\right), \tag{4.36}$$

where the Gaussian mixture components, $\mathbf{g}_m^{(k)}$ and $\bar{\mathbf{g}}_m^{(k)}$, $m = 1,\ldots, M$, of $\lambda^{(k)}$ and $\bar{\lambda}^{(k)}$ are the results of the *k*-th iteration, and

$$\frac{\partial p(o_t \mid \mathbf{g}_{C_i(t)}^{(k)})}{\partial f_z^{(k)}} = \begin{cases} p(o_t \mid \mathbf{g}_{C_i(t)}^{(k)})\left(o_t - \boldsymbol{\mu}_{C_i(t)}^{(k)}\right)'\Sigma_{C_i(t)}^{-1}\mathbf{e}_{C_i(t)}^{\perp} & \text{if } z = Z+1 \\ p(o_t \mid \mathbf{g}_{C_i(t)}^{(k)})\left(o_t - \boldsymbol{\mu}_{C_i(t)}^{(k)}\right)'\Sigma_{C_i(t)}^{-1}\mathbf{e}_{z,C_i(t)} & \text{otherwise} \end{cases} \tag{4.37}$$

and

$$\frac{\partial p(o_t \mid \bar{\mathbf{g}}_{C_i(t)}^{(k)})}{\partial \bar{f}_z^{(k)}} = \begin{cases} p(o_t \mid \bar{\mathbf{g}}_{C_i(t)}^{(k)})\left(o_t - \bar{\boldsymbol{\mu}}_{C_i(t)}^{(k)}\right)'\Sigma_{C_i(t)}^{-1}\bar{\mathbf{e}}_{C_i(t)}^{\perp} & \text{if } z = Z+1 \\ p(o_t \mid \bar{\mathbf{g}}_{C_i(t)}^{(k)})\left(o_t - \bar{\boldsymbol{\mu}}_{C_i(t)}^{(k)}\right)'\Sigma_{C_i(t)}^{-1}\mathbf{e}_{z,C_i(t)} & \text{otherwise,} \end{cases} \tag{4.38}$$

where

$$\boldsymbol{\mu}_{C_i(t)}^{(k)} = \boldsymbol{\eta}_{C_i(t)} + \sum_{j=1}^{Z} f_j^{(k)} \mathbf{e}_{j,C_i(t)} + f_{Z+1}^{(k)} \mathbf{e}_{C_i(t)}^{\perp} \tag{4.39}$$

and

$$\overline{\boldsymbol{\mu}}_{C_i(t)}^{(k)} = \boldsymbol{\eta}_{C_i(t)} + \sum_{j=1}^{Z} \bar{f}_j^{(k)} \mathbf{e}_{j,C_i(t)} + \bar{f}_{Z+1}^{(k)} \overline{\mathbf{e}}_{C_i(t)}^{\perp}. \tag{4.40}$$

If we assume that all covariance matrices $\boldsymbol{\Sigma}_m$ of the UBM, m = 1,…, M, are diagonal, Eqs. (4.37) - (4.40) can be rewritten, respectively, as

$$\frac{\partial p(o_t \mid \mathbf{g}_{C_i(t)}^{(k)})}{\partial f_z^{(k)}} = \begin{cases} p(o_t \mid \mathbf{g}_{C_i(t)}^{(k)}) \displaystyle\sum_{r_1=1}^{r} \frac{\left(o_t(r_1) - \boldsymbol{\mu}_{C_i(t)}^{(k)}(r_1)\right) \mathbf{e}_{C_i(t)}^{\perp}(r_1)}{\sigma_{C_i(t)}^2(r_1)} & \text{if } z = Z+1 \\ p(o_t \mid \mathbf{g}_{C_i(t)}^{(k)}) \displaystyle\sum_{r_1=1}^{r} \frac{\left(o_t(r_1) - \boldsymbol{\mu}_{C_i(t)}^{(k)}(r_1)\right) \mathbf{e}_{z,C_i(t)}(r_1)}{\sigma_{C_i(t)}^2(r_1)} & \text{otherwise,} \end{cases} \tag{4.41}$$

$$\frac{\partial p(o_t \mid \overline{\mathbf{g}}_{C_i(t)}^{(k)})}{\partial \bar{f}_z^{(k)}} = \begin{cases} p(o_t \mid \overline{\mathbf{g}}_{C_i(t)}^{(k)}) \displaystyle\sum_{r_1=1}^{r} \frac{\left(o_t(r_1) - \overline{\boldsymbol{\mu}}_{C_i(t)}^{(k)}(r_1)\right) \overline{\mathbf{e}}_{C_i(t)}^{\perp}(r_1)}{\sigma_{C_i(t)}^2(r_1)} & \text{if } z = Z+1 \\ p(o_t \mid \overline{\mathbf{g}}_{C_i(t)}^{(k)}) \displaystyle\sum_{r_1=1}^{r} \frac{\left(o_t(r_1) - \overline{\boldsymbol{\mu}}_{C_i(t)}^{(k)}(r_1)\right) \mathbf{e}_{z,C_i(t)}(r_1)}{\sigma_{C_i(t)}^2(r_1)} & \text{otherwise,} \end{cases} \tag{4.42}$$

$$\boldsymbol{\mu}_{C_i(t)}^{(k)}(r_1) = \boldsymbol{\eta}_{C_i(t)}(r_1) + \sum_{j=1}^{Z} f_j^{(k)} \mathbf{e}_{j,C_i(t)}(r_1) + f_{Z+1}^{(k)} \mathbf{e}_{C_i(t)}^{\perp}(r_1), \tag{4.43}$$

and

$$\overline{\boldsymbol{\mu}}_{C_i(t)}^{(k)}(r_1) = \boldsymbol{\eta}_{C_i(t)}(r_1) + \sum_{j=1}^{Z} \bar{f}_j^{(k)} \mathbf{e}_{j,C_i(t)}(r_1) + \bar{f}_{Z+1}^{(k)} \overline{\mathbf{e}}_{C_i(t)}^{\perp}(r_1), \tag{4.44}$$

where $\boldsymbol{\eta}_m(r_1)$, $\mathbf{e}_{z,m}(r_1)$, $\mathbf{e}_m^{\perp}(r_1)$, and $\overline{\mathbf{e}}_m^{\perp}(r_1)$, m = 1,…, M, r_1 = 1,…, r, represent the r_1-th elements of the m-th subvectors $\boldsymbol{\eta}_m$, $\mathbf{e}_{z,m}$, $\mathbf{e}_m^{\perp}$, and $\overline{\mathbf{e}}_m^{\perp}$, respectively.

4.4. Simplified Versions of LR-MVSE and E-MVSE

As far as reliability is concerned, a target speaker model trained with the GMM-UBM approach may be effective in characterizing the target speaker's voice. In contrast, a UBM generated from a number of background speakers may not be able to represent the imposters with respect to each specific target speaker. In other words, it may not be able to distinguish between imposters and the target speaker. Thus, it is more important to reinforce discriminability in the UBM than in the target speaker model. Moreover, in our experience, the training samples of target speakers are seldom mis-verified; i.e., nearly all the mis-verified training samples are from the cohort. Accordingly, to adapt the UBM to the target speaker dependent anti-model, it might be sufficient to use only negative training samples in our DFA framework. In this case, the training goal can be simplified to one of minimizing the average false acceptance (false alarm) loss ℓ_1. For LR-MVSE adaptation, the parameter $\overline{\mathbf{W}}$ is iteratively optimized using

$$\overline{\mathbf{W}}^{(k+1)} = \overline{\mathbf{W}}^{(k)} - \delta \frac{\partial \ell_1}{\partial \overline{\mathbf{W}}^{(k)}}, \tag{4.45}$$

where

$$\frac{\partial \ell_1}{\partial \overline{\mathbf{W}}^{(k)}} = \frac{\partial \ell_1}{\partial s(d(U))} \cdot \frac{\partial s(d(U))}{\partial d(U)} \cdot \frac{\partial d(U)}{\partial L(U)} \cdot \frac{\partial L(U)}{\partial \overline{\mathbf{W}}^{(k)}} = \frac{1}{N_1} \sum_{U \in H_1,\, L(U) > b} \left\{ 2a \cdot (L(U) - b) \cdot \frac{\partial L(U)}{\partial \overline{\mathbf{W}}^{(k)}} \right\}, \tag{4.46}$$

and $\frac{\partial L(U)}{\partial \overline{\mathbf{W}}^{(k)}}$ is computed by Eq. (4.14). For E-MVSE adaptation, the coordinates $\bar{f}_z$, z = 1,.., $Z+1$, are iteratively optimized using

$$\bar{f}_z^{(k+1)} = \bar{f}_z^{(k)} - \delta \frac{\partial \ell_1}{\partial \bar{f}_z^{(k)}}, \tag{4.47}$$

where

$$\frac{\partial \ell_1}{\partial \bar{f}_z^{(k)}} = \frac{\partial \ell_1}{\partial s(d(U))} \cdot \frac{\partial s(d(U))}{\partial d(U)} \cdot \frac{\partial d(U)}{\partial L(U)} \cdot \frac{\partial L(U)}{\partial \bar{f}_z^{(k)}} = \frac{1}{N_1} \sum_{U \in H_1,\, L(U) > b} \left\{ 2a \cdot (L(U) - b) \cdot \frac{\partial L(U)}{\partial \bar{f}_z^{(k)}} \right\}, \qquad (4.48)$$

and $\frac{\partial L(U)}{\partial \bar{f}_z^{(k)}}$ is computed by Eq. (4.36). When $N_0 \approx N_1$, the training times of the simplified versions of LR-MVSE and E-MVSE are about one-quarter of the training times of the respective original versions.

4.5. Experiments and Analysis

A. Experiment setup

In our experiments, we used the NIST 2001 cellular speaker recognition evaluation (NIST2001-SRE) database, and divided it into two subsets: an evaluation set and a development set. The evaluation set contained 74 male and 100 female speakers. On average, each speaker had approximately 2 minutes of training utterances and 10 test segments. The development set contained 38 males and 22 females as background speakers that did not overlap with the speakers in the evaluation set. To scale up the number of background speakers, we also included 139 male and 191 female speakers extracted from the NIST2002-SRE corpus. Thus, we collected the training utterances of 177 male and 213 female background speakers to build two gender-dependent UBMs, each containing 1,024 mixture components. To train each target speaker's GMM, we only adapted the mean vectors from the speaker's corresponding gender-dependent UBM in the GMM-UBM method. Then, for each male or female target speaker, we chose the *B* closest speakers from the 177 male or 213 female background speakers, respectively, as a cohort based on the degree of closeness measured in terms of the pairwise distance defined in Eq. (2.3). For each cohort speaker, we extracted *J* 3-second speech segments from his/her training utterances as negative samples of a target speaker. Thus, each target speaker had $J \times B$ negative samples in total. All the 3-second segments extracted from each target speaker's training utterances served as positive samples in LR-MVSE or E-MVSE adaptation.

To remove silence/noise frames, we processed all the speech data with a Voice Activity Detector (VAD). Then, using a 32-ms Hamming-windowed frame with 10-ms shifts, we converted each utterance into a stream of 30-dimensional feature vectors, each consisting of 15 Mel-scale frequency cepstral coefficients (MFCCs) and their first time derivatives. To compensate for channel mismatch effects, we applied feature warping [Pelecanos 2001] after MFCC extraction.

In the experiments, a and b in the s function defined in Eq. (4.1) were set at 3 and 0.01, respectively. For E-MVSE adaptation, we generated two gender-dependent Z-dimensional eigenspaces using the GMMs of the 177 male and 213 female background speakers, respectively, with Z set to 70 or 140. The LR-MVSE and E-MVSE adaptation procedures were trained until they almost converged, i.e., until the number of mis-verified training samples approximated zero. For the overall expected loss D, x_0 and x_1 were set as $C_{Miss} \times P_{Target}$ and $C_{FalseAlarm} \times (1 - P_{Target})$, respectively, according to the NIST Detection Cost Function (DCF) in Eq. (2.25). Following the NIST2001-SRE protocol, C_{Miss}, $C_{FalseAlarm}$, and P_{Target} were set at 10, 1, and 0.01, respectively.

B. Experiment results

To evaluate the performance of the DFA framework, we used the Detection Error Tradeoff (DET) curve and the NIST DCF; the latter reflects the performance at a single operating point on the former. We implemented the proposed DFA framework in three ways:

a) LR-MVSE adaptation ("MAP + LR-MVSE"),

b) E-MVSE adaptation with the first 70 eigenvectors (“MAP + E-MVSE70”), and

c) E-MVSE adaptation with the first 140 eigenvectors (“MAP + E-MVSE140”).

For the performance comparison, we used two baseline systems:

a) GMM-UBM (“MAP”) and

b) conventional MVE (MCE) training with the sigmoid function (“MAP + MVE”).

The target speaker GMM and the UBM obtained from the GMM-UBM method served as the initial models for the proposed DFA-related methods and the conventional MVE method.

Fig. 4.3 plots the minimum DCFs against the total number of negative training samples per target speaker for each adaptation method. The experiments involved 2,038 target speaker trials and 20,380 impostor trials of the evaluation set. We considered different numbers of negative samples, but not different numbers of positive samples because the same target speaker data had been used to train the initial target speaker model in the GMM-UBM method. From the figure, we observe that “MAP + E-MVSE70” achieves the lowest minDCF in cases where the adaptation data only includes 6 or 12 negative training samples per target speaker; while “MAP + LR-MVSE” achieves the lowest minDCF in cases where the adaptation data includes 36 or 60 negative training samples per target speaker. As expected, a small amount of adaptation data favors the methods in which a smaller number of model parameters must be estimated. Note that the larger the number of negative training samples used, the lower the minDCF that can be achieved.

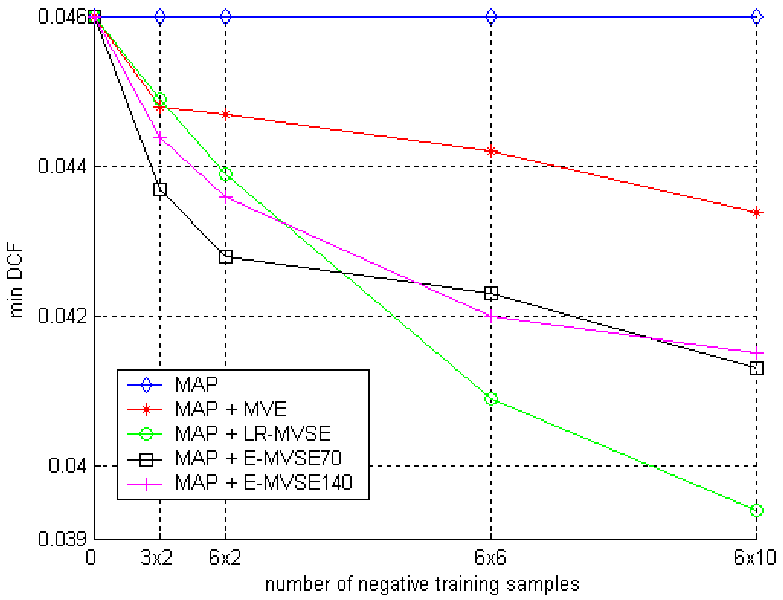


Fig. 4.3. The minimum DCFs versus the number (*J*×*B*) of 3-second negative training samples per target speaker.

Fig. 4.4 shows the DET curves obtained by evaluating the above systems for the case with 60 negative training samples per target speaker. It is clear that the performances of the three proposed methods, "MAP + LR-MVSE", "MAP + E-MVSE70", and "MAP + E-MVSE140", are comparable; and they all outperform the conventional methods "MAP" and "MAP + MVE". Interestingly, the performance of "MAP + MVE" is not always better than that of "MAP". This is because MVE tends to over-train the models obtained from the GMM-UBM method, and it is difficult to select the optimal stopping point in MVE training.

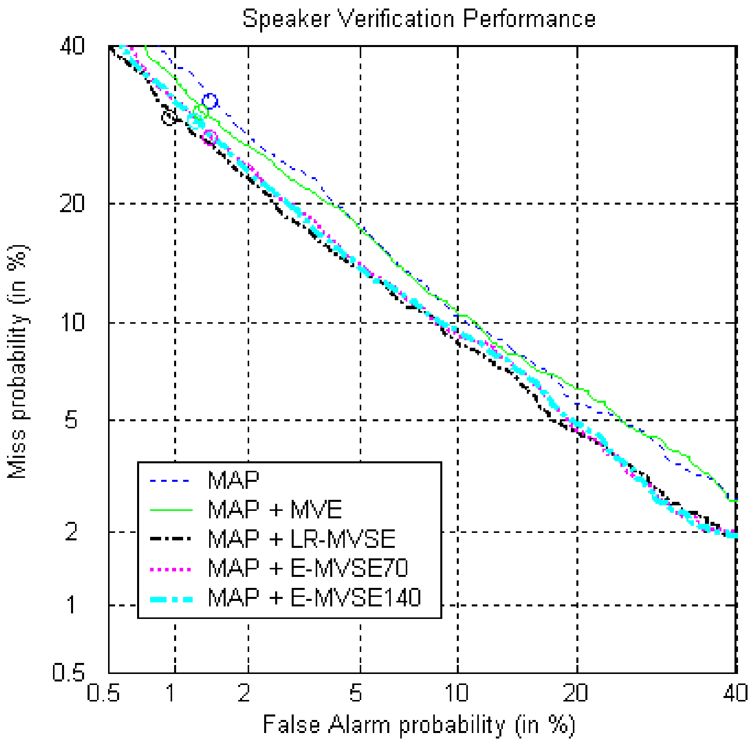
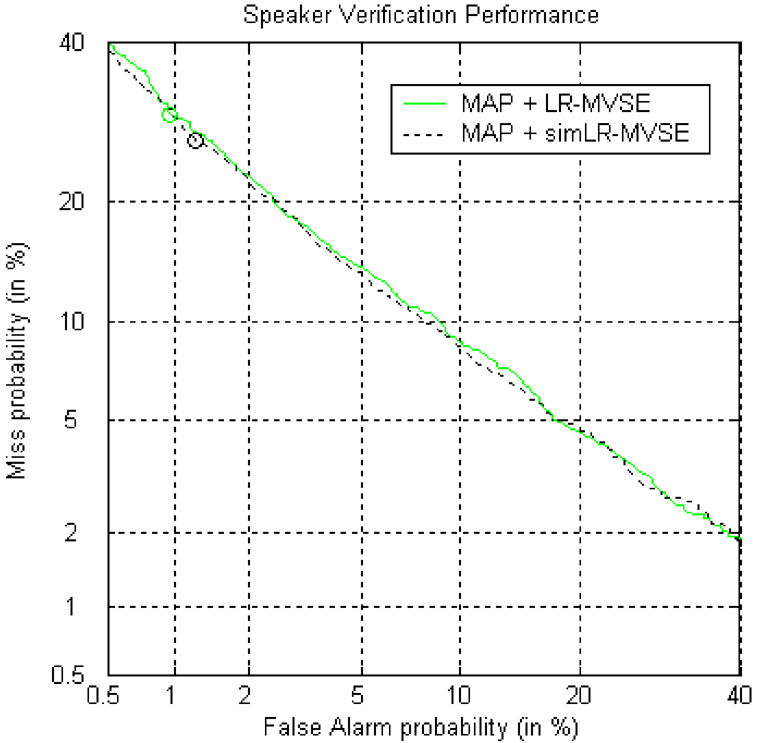


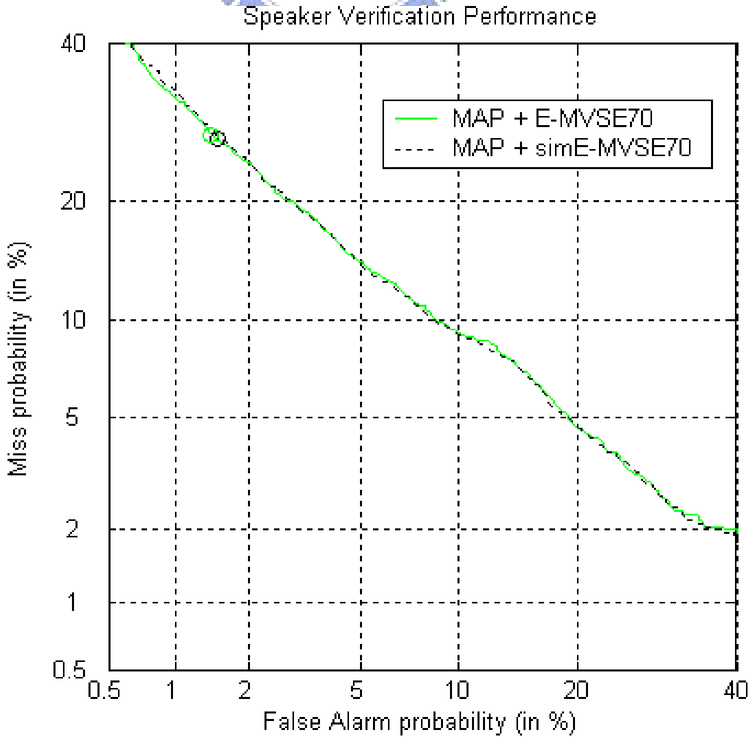
Fig. 4.4. Experiment results in DET curves. The circles indicate the minimum DCFs.

In the above experiments, we found that nearly all the mis-verified training samples in each adaptation iteration were negative training samples. Thus, we further compared the simplified versions of the LR-MVSE and E-MVSE methods with the respective original versions. Fig. 4.5 shows the DET curves for the case of 60 negative training samples per target speaker. It is clear that the simplified versions perform comparably to the respective original versions. This confirms our assumption that reinforcing the discriminability in the UBM is more beneficial than reinforcing the discriminability in the target speaker model.

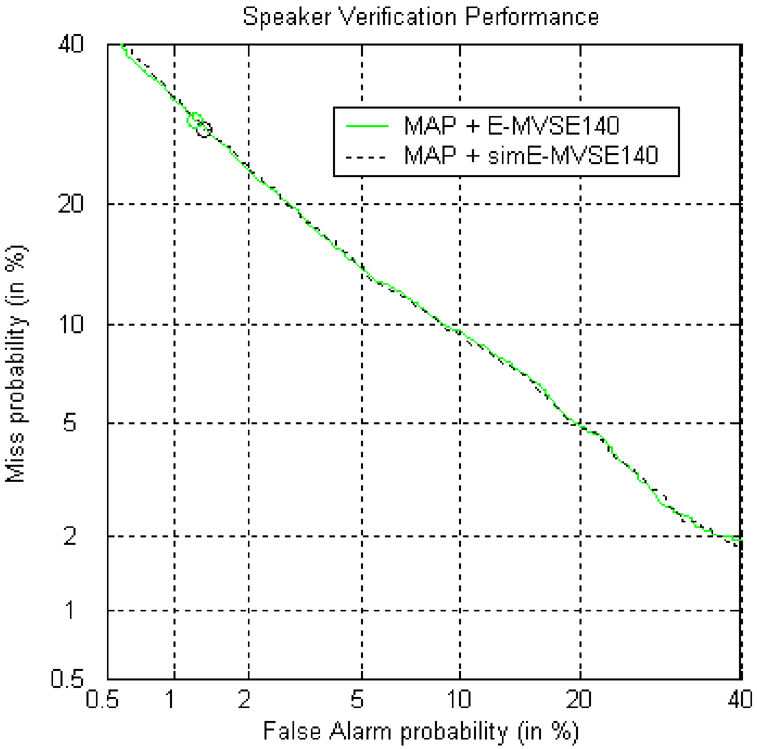
Table 4.1 summarizes the minimum DCFs of each system shown in Figs. 4.4 and 4.5. We observe that “MAP + LR-MVSE” achieves a 14.35% relative DCF reduction over the baseline GMM-UBM system (“MAP”) and a 9.22% relative DCF reduction over the “MAP + MVE” method. In fact, “MAP + simLR-MVSE” even performs slightly better than the original version “MAP + LR-MVSE”.



(a) LR-MVSE vs. the simplified version of LR-MVSE (simLR-MVSE)



(b) E-MVSE70 vs. the simplified version of E-MVSE70 (simE-MVSE70)



(c) E-MVSE140 vs. the simplified version of E-MVSE140 (simE-MVSE140)

Fig. 4.5. The DET curves of the LR-MVSE and E-MVSE systems and their simplified versions. The circles indicate the minimum DCFs.

Table 4.1. Summary of the minimum DCFs in Figs. 4.4 and 4.5.

Methods	minDCF
MAP	0.0460
MAP + MVE	0.0434
MAP + LR-MVSE	0.0394
MAP + E-MVSE70	0.0413
MAP + E-MVSE140	0.0415
MAP + simLR-MVSE	0.0390
MAP + simE-MVSE70	0.0420
MAP + simE-MVSE140	0.0416

Chapter 5

Conclusions

In this dissertation, we have proposed a framework to improve the characterization of the alternative hypothesis for speaker verification. The framework is built on either a weighted arithmetic combination (WAC) or a weighted geometric combination (WGC) of useful information extracted from a set of pre-trained background models. The proposed combinations are more effective and robust than the simple geometric mean and arithmetic mean used in conventional approaches. The parameters associated with WAC or WGC are then optimized using the minimum verification error (MVE) criterion, such that both the false acceptance probability and the false rejection probability are minimized. In addition to applying the conventional gradient-based MVE training method to this problem, we also proposed an evolutionary MVE (EMVE) training scheme to further reduce the verification errors. The results of our speaker verification experiments conducted on the Extended M2VTS Database (XM2VTSDB) demonstrate that the proposed systems along with the MVE or EMVE training achieve higher verification accuracy than conventional LR-based approaches. Although they need more training time than conventional LR-based approaches in the offline training phase, the increase of the training time for enrolling a new target speaker or the verification time for an input test utterance is negligible. The proposed systems are still capable of supporting a real-time response.

Alternatively, we have also presented two novel WGC- and WAC-based decision functions for solving the speaker-verification problem. The new decision functions are treated as nonlinear discriminant classifiers that can be solved by using kernel-based techniques, such as the Kernel Fisher Discriminant and Support Vector Machine, to optimally separate samples of the null hypothesis from those of the alternative hypothesis. The proposed approaches have two advantages over existing methods. The first is that they embed a trainable mechanism in the decision functions. The second is that they convert variable-length utterances into fixed-dimension characteristic vectors, which are easily processed by kernel discriminant analysis. The results of experiments on two speaker verification tasks, the XM2VTSDB and ISCSLP2006-SRE tasks, show notable improvements in performance over classical approaches. It is worth noting that although we only consider the speaker verification problem in this dissertation, the above proposed approach is not limited to this application. It can be applied to other types of data and hypothesis testing problems.

Finally, we have proposed a discriminative feedback adaptation (DFA) framework to improve the state of the art GMM-UBM speaker verification approach. The framework not only preserves the generalization ability of the GMM-UBM approach, but also reinforces the discrimination between H_0 and H_1. Our method is based on the minimum verification squared-error (MVSE) adaptation strategy, which is modified from the MVE training method so that only mis-verified training utterances are considered. Because a small number of mis-verified training samples may not be able to adapt a large number of model parameters, to implement DFA, we developed two adaptation techniques: the linear regression-based minimum verification squared-error (LR-MVSE) method and the eigenspace-based minimum verification squared-error (E-MVSE) method. In addition, we use a fast LR scoring approach and the simplified version of LR-MVSE or E-MVSE to improve the efficiency and effectiveness of the DFA framework. The results of experiments conducted on the

NIST2001-SRE database show that the proposed DFA framework can substantially improve the performance of the conventional GMM-UBM approach.

Bibliography

Auckenthaler, R., M. Carey, and H. Lloyd-Thomas, "Score Normalization for Text-Independent Speaker Verification System", *Digital Signal Processing*, vol. 10, no. 1, pp. 42-54, 2000.

Bengio, S. and J. Mariéthoz, "Learning the Decision Function for Speaker Verification", in *Proc. ICASSP*, Salt Lake City, USA, 2001, pp. 425-428.

Ben-Yacoub, S., "Multi-modal Data Fusion for Person Authentication Using SVM", in *Proc. AVBPA*, Washington DC, USA, 1999, pp. 25-30.

Bimbot, F., J. F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D. A. Reynolds, "A Tutorial on Text-Independent Speaker Verification", *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 430-451, 2004.

Burges, C., "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, vol.2, pp. 121-167, 1998.

Campbell, W. M., J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support Vector Machines for Speaker and Language Recognition", *Computer Speech and Language*, vol. 20, no. 2-3, pp. 210-229, 2006.

Campbell, W. M., D. E. Sturim, and D. A. Reynolds, "Support Vector Machine Using GMM Supervectors for Speaker Verification", *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308-311, 2006.

Campbell, W. M., J. P. Campbell, T. P. Gleason, D. A. Reynolds, and W. Shen, "Speaker Verification Using Support Vector Machines and High-Level Features", *IEEE Trans. Audio , Speech and Language Processing*, vol. 15, no. 7, pp. 2085-2094, 2007.

Chao, Y. H., W. H. Tsai, H. M. Wang, and R. C. Chang, "A Kernel-based Discrimination Framework for Solving Hypothesis Testing Problems with Application to Speaker Verification", in *Proc. ICPR.*, Hong Kong, China, 2006, pp. 229-232.

Cheng, S. S., Y. H. Chao, H. M. Wang, and H. C. Fu, "A Prototypes Embedded Genetic Algorithm for K-means Clustering", in *Proc. ICPR2006*.

Cheng, H. T., Y. H. Chao, S. L. Yen, C. S. Chen, H. M. Wang, and Y. P. Hung, "An Efficient Approach to Multi-Modal Person Identity Verification by Fusing Face and Voice Information", in *Proc. ICME*, Amsterdam, The Netherlands, July 2005.

Chengalvarayan, R., "Speaker Adaptation Using Discriminative Linear Regression on Time-Varying Mean Parameters in Trended HMM", *IEEE Signal Processing Letters*, vol. 5, no. 3, pp. 63-65, 1998.

Chinese Corpus Consortium (CCC), "Evaluation Plan for ISCSLP'2006 Special Session on Speaker Recognition", 2006.

Chou, W. and B. H. Juang, *Pattern Recognition in Speech and Language Processing*, CRC Press, 2003.

Duda, R. O., P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd. ed., John Wiley & Sons, New York, 2001.

Eiben, A. E. and J. E. Smith, *Introduction to Evolutionary Computing*, Springer, Berlin, 2003.

Faundez-Zanuy, M. and E. Monte-Moreno, "State-of-the-Art in Speaker Recognition", *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 5, pp.7-12, 2005.

Fauve, B. G. B., D. Matrouf, N. Scheffer, J. F. Bonastre, and J. S. D. Mason, "State-of-the-Art Performance in Text-Independent Speaker Verification Through Open-Source Software", *IEEE Trans. Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 1960-1968, 2007.

Gauvain, J. L. and C. H. Lee, “Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observation of Markov Chains”, *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 2, pp. 291-298, 1994.

Gillick, L. and S. J. Cox, “Some Statistical Issues in the Comparison of Speech Recognition Algorithms”, in *Proc. ICASSP*, Glasgow, UK, 1989, pp. 532-535.

He, X. D. and W. Chou, “Minimum Classification Error Linear Regression for Acoustic Model Adaptation of Continuous Density HMMs”, in *Proc. ICASSP2003*.

He, X. D. and W. Chou, “Minimum Classification Error (MCE) Model Adaptation of Continuous Density HMMs”, in *Proc. Eurospeech2003*.

Herbrich, R., *Learning Kernel Classifiers: Theory and Algorithms*, MIT Press, Cambridge, 2002.

Higgins, A., L. Bahler, and J. Porter, “Speaker Verification Using Randomized Phrase Prompting”, *Digital Signal Processing*, vol. 1, no. 2, pp. 89-106, 1991.

Huang, X., A. Acero, and H. W. Hon, *Spoken Language Processing*, Prentics Hall, New Jersey, 2001.

Juang, B. H., W. Chou, and C. H. Lee, “Minimum Classification Error Rate Methods for Speech Recognition”, *IEEE Trans. Speech and Audio Processing*, vol. 5, pp. 257-265, 1997.

Krishna, K. and M. N. Murty, “Genetic K-Means Algorithm”, *IEEE Trans. Systems, Man, and Cybernetics – Part B*, vol. 29, no. 3, pp. 433-439, June 1999.

Kuhn, R., J. C. Junqua, P. Nguyen, and N. Niedzielski, “Rapid Speaker Adaptation in Eigenvoice Space”, *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 6, pp. 695-707, 2000.

Kuo, H. K. J., C. H. Lee, I. Zitouni, and E. Fosler-Lussiert, “Minimum Verification Error Training for Topic Verification”, in *Proc. ICASSP2003*.

Lee, Y. J. and O. L. Mangasarian, “SSVM: Smooth Support Vector Machine for Classification”, *Computational Optimization and Applications*, vol. 20, no. 1, pp. 5-22, 2001.

Lindberg, J., J. Koolwaaij, H. P. Hutter, D. Genoud, J. B. Pierrot, M. Blomberg, and F. Bimbot, “Techniques for A Priori Decision Threshold Estimation in Speaker Verification”, in *Proc. RLA2C*, pp. 89-92, 1998.

Liu, C. S., H. C. Wang, and C. H. Lee, “Speaker Verification Using Normalized Log-Likelihood Score”, *IEEE Trans. Speech and Audio Processing*, vol. 4, no. 1, pp. 56-60, 1996.

Lu, Y., S. Lu, F. Fotouhi, Y. Deng, and S. J. Brown, “FGKA: A Fast Genetic K-means Clustering Algorithm”, in *Proc. ACM Symposium on Applied Computing*, 2004.

Luettin, J. and G. Maitre, *Evaluation Protocol for the Extended M2VTS Database (XM2VTSDB)*, IDIAP-COM 98-05, IDIAP, 1998.

Ma, C. and E. Chang, “Comparison of Discriminative Training Methods for Speaker Verification”, *Proc. ICASSP2003*.

Mami, Y. and D. Charlet, “Speaker Recognition by Location in the Space of Reference Speakers”, *Speech Communication*, vol. 48, pp. 127-141, 2006.

Martin, A., G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, “The DET Curve in Assessment of Detection Task Performance”, in *Proc. Eurospeech1997*.

McDermott, E., T. J. Hazen, J. Le Roux, A. Nakamura, and S. Katagiri, “Discriminative Training for Large-Vocabulary Speech Recognition Using Minimum Classification Error”, *IEEE Trans. Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 203-223, 2007.

Messer, K., J. Matas, J. Kittler, J. Luettin, and G. Maitre, “XM2VTSDB: The Extended M2VTS Database”, in *Proc. AVBPA1999*.

Mika, S., G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller, “Fisher Discriminant Analysis with Kernels”, in *Proc. Neural Networks for Signal Processing IX*, Madison, WI, USA, 1999, pp. 41-48.

Mika, S., “Kernel Fisher Discriminants”, Ph.D thesis, University of Technology, Berlin, 2002.

Pelecanos, J. and S. Sridharan, “Feature Warping for Robust Speaker Verification”, in *Proc. Odyssey2001*.

Przybocki, M. A., A. F. Martin, and A. N. Le, “NIST Speaker Recognition Evaluations Utilizing the Mixer Corpora—2004, 2005, 2006”, *IEEE Trans. Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 1951-1959, 2007.

Rahim, M. G. and C. H. Lee, “String based Minimum Verification Error (SB-MVE) Training for Flexible Speech Recognition”, *Computer Speech and Language*, vol. 11, no. 2, pp. 147-160, 1997.

Reynolds, D. A., “Speaker Identification and Verification Using Gaussian Mixture Speaker Models”, *Speech Communication*, vol.17, no. 1-2, pp. 91-108, 1995.

Reynolds, D. A., T. F. Quatieri, and R. B. Dunn, “Speaker Verification Using Adapted Gaussian Mixture Models”, *Digital Signal Processing*, vol. 10, no. 1, pp. 19-41, 2000.

Rosenberg, A. E., J. DeLong, C. H. Lee, B. H. Juang and F. K. Soong, “The Use of Cohort Normalized Scores for Speaker Verification”, in *Proc. ICSLP1992*.

Rosenberg, A. E., O. Siohan, and S. Parthasarathy, “Speaker Verification Using Minimum Verification Error Training”, in *Proc. ICASSP1998.*

Siohan, O., A. E. Rosenberg, and S. Parthasarathy, “Speaker Identification Using Minimum Classification Error Training”, *Proc. ICASSP1998.*

Siu, M. H., B. Mak, and W. H. Au, “Minimization of Utterance Verification Error Rate as a Constrained Optimization Problem”, *IEEE Signal Processing Letters*, vol. 13, no. 12, pp. 760-763, 2006.

Strang, G., *Linear Algebra and Its Applications*, 4th. ed., Brooks/Cole, 2005.

Sturim, D. E., D. A. Reynolds, E. Singer, and J. P. Campbell, "Speaker Indexing in Large Audio Databases Using Anchor Models", in *Proc. ICASSP*, Salt Lake City, USA, 2001, vol.1, pp. 429-432.

Sturim, D. E. and D. A. Reynolds, "Speaker Adaptive Cohort Selection for Tnorm in Text-Independent Speaker Verification", in *Proc. ICASSP2005*.

Sukkar, R. A., A. R. Setlur, M. G. Rahim, and C. H. Lee, "Utterance Verification of Keyword Strings Using Word-Based Minimum Verification Error (WB-MVE) Training", in *Proc. ICASSP1996*.

Sukkar, R. A., "Subword-based Minimum Verification Error (SB-MVE) Training for Task Independent Utterance Verification", in *Proc. ICASSP1998*.

Thyes, O., R. Kuhn, P. Nguyen, and J.-C. Junqua, "Speaker Identification and Verification Using Eigenvoices", in *Proc. ICSLP2000*.

Valente, F. and C. Wellekens, "Minimum Classification Error/Eigenvoices Training for Speaker Identification", in *Proc. ICASSP2003*.

Van Leeuwen, D. A., A. F. Martin, M. A. Przybocki, and J. S. Bouten, "NIST and NFI-TNO Evaluations of Automatic Speaker Recognition", *Computer Speech and Language*, vol. 20, pp. 128-158, 2006.

Vapnik, V., *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.

Wan, V. and S. Renals, "Speaker Verification Using Sequence Discriminant Support Vector Machines", *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 2, pp. 203-210, 2005.

Wu, J. and Q. Huo, "Supervised Adaptation of MCE-Trained CDHMMs Using Minimum Classification Error Linear Regression", in *Proc. ICASSP2002*.

Zheng, T. F., Z. Song, L. Zhang, M. Brasser, W. Wu, and J. Deng, “CCC Speaker Recognition Evaluation 2006: Overview, Methods, Data, Results and Perspective”, in *Proc. ISCSLP*, Kent Ridge, Singapore, Dec. 2006.