Pattern Recognition 42 (2009) 1351 – 1360

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Improving the characterization of the alternative hypothesis via minimum verification error training with applications to speaker verification

Yi-Hsiang Chao[a,b,1], Wei-Ho Tsai[c,*], Hsin-Min Wang[a,2], Ruei-Chuan Chang[b,3]

[a] Institute of Information Science, Academia Sinica, Taipei 115, Taiwan
[b] Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan
[c] Department of Electronic Engineering & Graduate Institute of Computer and Communication Engineering, National Taipei University of Technology, Taipei 10608, Taiwan

ARTICLE INFO

Article history:
Received 18 August 2007
Received in revised form 7 August 2008
Accepted 19 October 2008

Keywords:
Genetic algorithm
Hypothesis testing
Log-likelihood ratio
Minimum verification error training
Speaker verification

ABSTRACT

Speaker verification is usually formulated as a statistical hypothesis testing problem and solved by a log-likelihood ratio (LLR) 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 paper, 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. Our experiment results show that the proposed framework outperforms conventional LLR-based approaches.

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 measurement $L(\cdot)$ for a given observation X:

$$\begin{aligned} H_0 &: L(X) \geqslant \theta \\ H_1 &: L(X) < \theta, \end{aligned} \tag{1}$$

where θ is the decision threshold. A number of measurements have been investigated for various applications, with the log-likelihood ratio (LLR) measure combined with parametric modeling being the most popular. Specifically, each hypothesis is represented by a set of probability-related parameters through a training process, and the probability of generating a given observation X is then evaluated for the parameter set of each hypothesis. The LLR test is expressed as

$$L(X) = \log \frac{p(X|H_0)}{p(X|H_1)} \begin{cases} \geqslant \theta & \text{accept } H_0 \\ < \theta & \text{accept } H_1 \text{ (i.e., reject } H_0\text{)}, \end{cases} \tag{2}$$

where $p(X|H_i)$, $i = 0$ or 1, is the probability of observation X under hypothesis H_i. The hypotheses H_0 and H_1 can be represented by parametric models such as Gaussian mixture models (GMMs) [1,2], which are usually denoted as λ (the null hypothesis model or target model) and $\bar{\lambda}$ (the alternative hypothesis model or anti-model).

However, in most applications, the alternative hypothesis model is usually ill-defined and difficult to characterize a priori. For example, in speaker verification [3–7], 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

[*] Corresponding author. Tel.: +886 2 27712171x2257; fax: +886 2 27317120.
E-mail addresses: yschao@iis.sinica.edu.tw (Y.-H. Chao), whtsai@ntut.edu.tw (W.-H. Tsai), whm@iis.sinica.edu.tw (H.-M. Wang), rc@cc.nctu.edu.tw (R.-C. Chang).
[1] Tel.: +886 2 27883799x1568; fax: +886 2 27824814.
[2] Tel.: +886 2 27883799x1714; fax: +886 2 27824814.
[3] Tel.: +886 3 5712121x31723; fax: +886 3 5721490.

0031-3203/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.10.015

speaker, and hence lacks explicit data for modeling. Thus, a number of approaches have been proposed to better characterize H_1. One popular approach pools all the speech data from a large number of background speakers and trains a single speaker-independent model Ω, called the world model or the universal background model (UBM) [2]. During a test, the LLR measure that an unknown utterance U was spoken by the claimed speaker can be evaluated by

$$L_{\mathrm{UBM}}(U) = \log p(U|\lambda) - \log p(U|\Omega), \tag{3}$$

where λ is the target speaker model trained using speech from the claimed speaker. The larger the value of $L_{\mathrm{UBM}}(U)$, the more likely it is that the utterance U was spoken by the claimed speaker. Due to the good generalization ability of the UBM, $L_{\mathrm{UBM}}(U)$ (usually called the GMM–UBM method [2]) 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 models $\{\lambda_1, \lambda_2, \ldots, \lambda_B\}$ using speech from several representative speakers, called a cohort [9], which simulates potential impostors. This leads to the following possible LLR measures, where the alternative hypothesis can be characterized by:

(i) the likelihood of the most competitive cohort model [10], i.e.,

$$L_{\mathrm{Max}}(U) = \log p(U|\lambda) - \max_{1 \leqslant i \leqslant B} \log p(U|\lambda_i), \tag{4}$$

(ii) the arithmetic mean of the likelihoods of the B cohort models [1], i.e.,

$$L_{\mathrm{Ari}}(U) = \log p(U|\lambda) - \log \left\{ \frac{1}{B} \sum_{i=1}^{B} p(U|\lambda_i) \right\}, \tag{5}$$

(iii) the geometric mean of the likelihoods of the B cohort models [10], i.e.,

$$L_{\mathrm{Geo}}(U) = \log p(U|\lambda) - \frac{1}{B} \sum_{i=1}^{B} \log p(U|\lambda_i). \tag{6}$$

In a well-known score normalization method called *T-norm* [12,13], $L_{\mathrm{Geo}}(U)$ is divided by the standard deviation of the log-likelihoods of the B cohort models.

The approaches in Eqs. (3)–(6), which have been proposed to characterize $H_{1,}$ can be expressed collectively in the following general form [2]:

$$p(U|\bar{\lambda}) = \Psi(p(U|\lambda_1), p(U|\lambda_2) \ldots, p(U|\lambda_N)), \tag{7}$$

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 yields $L_{\mathrm{Max}}(U)$, while the arithmetic mean yields $L_{\mathrm{Ari}}(U)$, and the geometric mean yields $L_{\mathrm{Geo}}(U)$. When $\Psi(\cdot)$ is an identity function, $N = 1$, and $\lambda_1 = \Omega$, Eq. (7) yields $L_{\mathrm{UBM}}(U)$.

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. To handle this problem more effectively, it is necessary to design a trainable mechanism for $\Psi(\cdot)$. We therefore 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 $L_{\mathrm{Geo}}(U)$ or the arithmetic mean in $L_{\mathrm{Ari}}(U)$, 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 [14,15], 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 [21–24], 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 [14–20], 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) [14], 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 LLR measures based on a genetic algorithm (GA) [25]. 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) [29] demonstrate that the proposed methods outperform conventional LLR-based approaches.

The remainder of this paper is organized as follows. Section 2 presents the proposed methods for characterizing the alternative hypothesis. Sections 3 and 4 describe, respectively, the gradient-based MVE training and the EMVE training used to optimize our methods. Section 5 contains the experiment results. Then, in Section 6, we present our conclusions.

2. 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 LLR measure based on WAC or WGC can be viewed as a generalized and trainable version of $L_{\mathrm{UBM}}(U)$, $L_{\mathrm{Max}}(U)$, $L_{\mathrm{Ari}}(U)$ or $L_{\mathrm{Geo}}(U)$.

2.1. The weighted arithmetic combination (WAC)

First, we define the function $\Psi(\cdot)$ in Eq. (7) based on the weighted arithmetic combination as

$$p(U|\bar{\lambda}) = \Psi(p(U|\lambda_1), \ldots, p(U|\lambda_N)) = \sum_{i=1}^{N} w_i p(U|\lambda_i), \tag{8}$$

where w_i is the weight of the likelihood $p(U|\lambda_i)$ 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. (8) 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.

2.2. The weighted geometric combination (WGC)

Alternatively, we can define the function $\Psi(\cdot)$ in Eq. (7) from the perspective of the weighted geometric combination as

$$p(U|\bar{\lambda}) = \Psi(p(U|\lambda_1), \ldots, p(U|\lambda_N)) = \prod_{i=1}^{N} p(U|\lambda_i)^{w_i}. \tag{9}$$

Similar to the weighted arithmetic combination, Eq. (9) 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 $p(U|\bar{\lambda}) \rightarrow 0$. 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., $p(U|\lambda_i) \rightarrow 0$. However, if a weight is attached to each background model, $\Psi(\cdot)$ defined in Eq. (9) should be less sensitive to a tiny value of the likelihood; hence, it should be more robust and reliable than the heuristic geometric mean.

2.3. Relation to conventional LLR measures

We observe that Eqs. (8) and (9) 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 Eqs. (8) and (9) will degenerate to a maximum function if we set $w_{i^*} = 1$, where $i^* = \text{argmax}_{1 \leqslant i \leqslant N} p(U|\lambda_i)$, and $w_i = 0$, $\forall i \neq i^*$. Furthermore, the LLR measure based on Eq. (8) or (9) will degenerate to $L_{\text{UBM}}(U)$ in Eq. (3) if only a UBM Ω is used as the background model. Thus, both WAC- and WGC-based LLR measures can be viewed as generalized and trainable versions of $L_{\text{UBM}}(U)$, $L_{\text{Max}}(U)$, $L_{\text{Ari}}(U)$ or $L_{\text{Geo}}(U)$.

In the WAC method, we refer to the alternative hypothesis model $\bar{\lambda}$ defined in Eq. (8) 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 [31], while the outer mixture weights are estimated using MVE training or evolutionary MVE (EMVE) training, which we discuss in Section 3 and Section 4, respectively. In other words, GMM2 integrates the Bayesian learning and discriminative training algorithms. The objective is to optimize the LLR measure by considering the null hypothesis and the alternative hypothesis jointly.

2.4. 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 [1,9–13] 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 LLR measures.

2.4.1. 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. (8) and WGC in Eq. (9). Depending on the definition of a cohort, we consider two commonly-used methods [1]. One selects the B closest speaker models {$\lambda_{\text{cst}1}$, $\lambda_{\text{cst}2}$, …, $\lambda_{\text{cst}B}$} 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 [1]:

$$d(\lambda_i, \lambda_j) = \log \frac{p(U_i|\lambda_i)}{p(U_i|\lambda_j)} + \log \frac{p(U_j|\lambda_j)}{p(U_j|\lambda_i)}, \tag{10}$$

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, {Ω, $\lambda_{\text{cst}1}$, $\lambda_{\text{cst}2}, \ldots, \lambda_{\text{cst}B}$} or {$\Omega$, $\lambda_{\text{cst}1}, \ldots, \lambda_{\text{cst}B/2}$, $\lambda_{\text{fst}1}$, …, $\lambda_{\text{fst}B/2}$}, for Eq. (7).

2.4.2. Combining multiple types of anti-models

As shown in Eqs. (3)–(6), various types of anti-models have been studied for conventional LLR measures. However, none of the LLR 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 LLR measures motivate us to combine them into a unified LLR measure because of the complementary information that each anti-model can contribute.

Consider K different LLR measures $L_i(U)$, each with an anti-model $\bar{\lambda}_i$, $i = 1,2,\ldots,K$. If we treat each anti-model $\bar{\lambda}_i$ as a background model, Eq. (7) can be rewritten as,

$$p(U|\bar{\lambda}) = \Psi(p(U|\bar{\lambda}_1), p(U|\bar{\lambda}_2) \ldots, p(U|\bar{\lambda}_K)). \tag{11}$$

Using WAC or WGC to realize Eq. (11), we can form a trainable version of the conventional LLR measures in Eqs. (3)–(6), where each anti-model $\bar{\lambda}_i$, $i = 1,\ldots,4$, is computed, respectively, by

$$p(U|\bar{\lambda}_1) = p(U|\Omega), \tag{12}$$

$$p(U|\bar{\lambda}_2) = \max_{1 \leqslant i \leqslant B} p(U|\lambda_i), \tag{13}$$

$$p(U|\bar{\lambda}_3) = \frac{1}{B} \sum_{i=1}^{B} p(U|\lambda_i), \tag{14}$$

and

$$p(U|\bar{\lambda}_4) = \left(\prod_{i=1}^{B} p(U|\lambda_i) \right)^{1/B}. \tag{15}$$

As a result, for Eq. (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.

3. Gradient-based MVE 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 MVE training [14,15].

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 $g_i(U)$, $i = 0,1,\ldots,M-1$. The misclassification measure in the MCE method [21] is defined as

$$d_i(U) = -g_i(U) + \log\left[\frac{1}{M-1}\sum_{j,j\neq i}\exp[g_j(U)\eta]\right]^{1/\eta}, \tag{16}$$

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

$$g_i(U) = \begin{cases}\log p(U|\lambda) & \text{if } i = 0\\ \log p(U|\bar{\lambda}) & \text{if } i = 1,\end{cases} \tag{17}$$

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

$$d(U) = \begin{cases} d_0(U) = -L(U) & \text{if } U \in H_0\\ d_1(U) = L(U) & \text{if } U \in H_1,\end{cases} \tag{18}$$

where $L(U)$ is the LLR in Eq. (2). We further express $L(U)$ as the following equivalent test

$$L(U) = \log p(U|\lambda) - \log p(U|\bar{\lambda}) - \theta \begin{cases} \geqslant 0 & \text{accept } H_0\\ < 0 & \text{accept } H_1,\end{cases} \tag{19}$$

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

$$s(d(U)) = \frac{1}{1+\exp(-a \cdot d(U))} \tag{20}$$

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

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

$$\ell_i = \frac{1}{N_i}\sum_{U\in H_i} s(d(U)), \tag{21}$$

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

$$D = x_0\ell_0 + x_1\ell_1, \tag{22}$$

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 Eqs. (8) and (9) such that Eq. (22) can be minimized. This can be achieved by using the gradient descent algorithm [14]. 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

$$w_i = \frac{\exp(\alpha_i)}{\sum_{j=1}^{N}\exp(\alpha_j)} \tag{23}$$

which is similar to the strategy used in [21]. Parameter α_i is iteratively optimized using

$$\alpha_i^{(t+1)} = \alpha_i^{(t)} - \varepsilon\frac{\partial D}{\partial \alpha_i} \tag{24}$$

where ε is the step size, and

$$\begin{aligned}\frac{\partial D}{\partial \alpha_i} &= x_0\frac{\partial \ell_0}{\partial \alpha_i} + x_1\frac{\partial \ell_1}{\partial \alpha_i}\\ &= x_0\frac{\partial \ell_0}{\partial s}\cdot\frac{\partial s}{\partial d}\cdot\frac{\partial d}{\partial L}\cdot\frac{\partial L}{\partial \alpha_i} + x_1\frac{\partial \ell_1}{\partial s}\cdot\frac{\partial s}{\partial d}\cdot\frac{\partial d}{\partial L}\cdot\frac{\partial L}{\partial \alpha_i}\\ &= x_0\cdot\frac{1}{N_0}\sum_{U\in H_0}\left\{a\cdot s(-L(U))[1-s(-L(U))]\cdot\left(-\frac{\partial L}{\partial \alpha_i}\right)\right\}\\ &\quad + x_1\cdot\frac{1}{N_1}\sum_{U\in H_1}\left\{a\cdot s(L(U))[1-s(L(U))]\cdot\frac{\partial L}{\partial \alpha_i}\right\},\end{aligned} \tag{25}$$

where

$$\frac{\partial L}{\partial \alpha_i} = \sum_{j=1}^{N}\left(\frac{\partial L}{\partial w_j}\cdot\frac{\partial w_j}{\partial \alpha_i}\right) = w_i\left(\frac{\partial L}{\partial w_i} - \sum_{j=1}^{N} w_j\frac{\partial L}{\partial w_j}\right). \tag{26}$$

If WAC is used, then

$$\frac{\partial L}{\partial w_i} = \frac{-\partial}{\partial w_i}\log\left(\sum_{j=1}^{N} w_j p(U|\lambda_j)\right) = \frac{-p(U|\lambda_i)}{\sum_{j=1}^{N} w_j p(U|\lambda_j)}. \tag{27}$$

If WGC is used, then

$$\frac{\partial L}{\partial w_i} = \frac{-\partial}{\partial w_i}\left(\sum_{j=1}^{N} w_j\log p(U|\lambda_j)\right) = -\log p(U|\lambda_i). \tag{28}$$

The threshold θ in Eq. (19) can be estimated using

$$\theta^{(t+1)} = \theta^{(t)} - \varepsilon\frac{\partial D}{\partial \theta}, \tag{29}$$

where

$$\begin{aligned}\frac{\partial D}{\partial \theta} &= x_0\frac{\partial \ell_0}{\partial s}\cdot\frac{\partial s}{\partial d}\cdot\frac{\partial d}{\partial L}\cdot\frac{\partial L}{\partial \theta} + x_1\frac{\partial \ell_1}{\partial s}\cdot\frac{\partial s}{\partial d}\cdot\frac{\partial d}{\partial L}\cdot\frac{\partial L}{\partial \theta}\\ &= x_0\cdot\frac{1}{N_0}\sum_{U\in H_0} a\cdot s(-L(U))[1-s(-L(U))]\\ &\quad - x_1\cdot\frac{1}{N_1}\sum_{U\in H_1} a\cdot s(L(U))[1-s(L(U))].\end{aligned} \tag{30}$$

In our implementation, the overall expected loss is set as

$$D = C_{Miss}\times\ell_0\times P_{Target} + C_{FalseAlarm}\times\ell_1\times(1-P_{Target}). \tag{31}$$

Eq. (31) simulates the detection cost function (DCF) [8]

$$\begin{aligned}C_{DET} = {}& C_{Miss}\times P_{Miss}\times P_{Target} + C_{FalseAlarm}\times P_{FalseAlarm}\\ &\times(1-P_{Target}),\end{aligned} \tag{32}$$

where C_{Miss} denotes the cost of the miss (false rejection) error; $C_{FalseAlarm}$ denotes the cost of the false alarm (false acceptance) error; $P_{Miss} \approx l_0$ is the miss (false rejection) probability; $P_{FalseAlarm} \approx l_1$ is the false alarm (false acceptance) probability; and P_{Target} is the *a priori* probability of the target speaker.

4. Evolutionary MVE 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 LLR 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 [25]. As shown

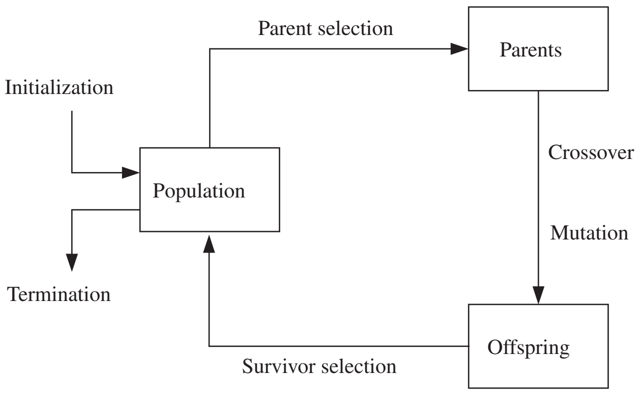

Fig. 1. The general scheme of a GA.

in Fig. 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. (22), 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.

(1) *Encoding*: Each chromosome is a string $\{\alpha_1, \alpha_2, \ldots, \alpha_N, \theta\}$ of length N+1, which is the concatenation of all intermediate parameters α_i in Eq. (23) and the threshold θ in Eq. (19). 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* [25].

(3) *Crossover*: We use the N-point crossover [25] 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) [26–28], 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, $i = 1, 2, \ldots, N$ as follows:

$$\alpha_i^{new} = \alpha_i^{old} - \varepsilon \frac{\partial D}{\partial \alpha_i}, \tag{33}$$

where α_i^{new} and α_i^{old} are, respectively, the parameter α_i in a chromosome after and before mutation; ε is the step size ; and $\partial D/\partial \alpha_i$ is computed by Eq. (25). Similarly, the GDO for the threshold θ is computed by

$$\theta^{new} = \theta^{old} - \varepsilon \frac{\partial D}{\partial \theta}, \tag{34}$$

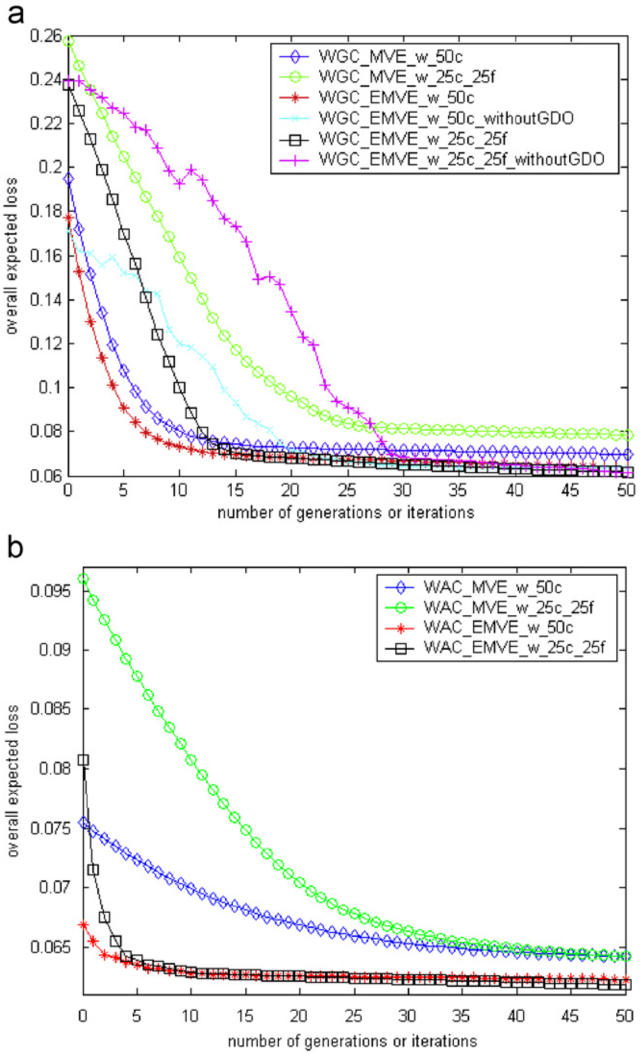

Fig. 2. The learning curves of gradient-based MVE and EMVE for the "Evaluation" subset in Configuration II. (a) WGC methods and (b) WAC methods.

where θ^{new} and θ^{old} are, respectively, the threshold θ in a chromosome after and before mutation; and $\partial D/\partial \theta$ is computed by Eq. (30).

(5) *Survivor selection*: We adopt the generational model [25] 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), 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.

5. Experiments

We evaluated the proposed approaches via speaker verification experiments conducted on speech data extracted from the Extended M2VTS Database (XM2VTSDB) [29]. The first set of experiments followed Configuration II of XM2VTSDB, as defined in [30]. 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) [6–8].

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. (32), 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) [32]. All the experiments were conducted on a 3.2 GHz Intel Pentium IV computer with 1.5 GB of RAM, running Windows XP.

5.1. Evaluation based on Configuration II

In accordance with Configuration II of XM2VTSDB, the database was divided into three subsets: "Training", "Evaluation[4] ", and "Test". We used the "Training" subset to build each target speaker's model and the background models. The "Evaluation" subset was used to optimize the weights w_i in Eq. (8) or (9), along with the threshold θ. Then, the speaker verification performance was evaluated on the "Test" subset. As shown in Table 1, a total of 293 speakers[5] 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".

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 [31] and their first time derivatives.

We used 12 (2×2×3) utterances/client from sessions 1 and 2 to train each client model, 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 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 1
Configuration II of XM2VTSDB [30]

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			

Table 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	√	√	√

5.1.1. Experiment results

First, we compared the learning ability of gradient-based MVE training and EMVE training in the proposed WGC- and WAC-based LLR 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 LLR 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(a) and (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, 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 LLR systems as our baselines:

(a) $L_{UBM}(U)$ ("Lubm"),
(b) $L_{Max}(U)$ with the 50 closest cohort models ("Lmax_50c"),
(c) $L_{Geo}(U)$ with the 50 closest cohort models ("Lgeo_50c"),
(d) $L_{Geo}(U)$ with the 25 closest cohort models and the 25 farthest cohort models ("Lgeo_25c_25f"),
(e) $L_{Ari}(U)$ with the 50 closest cohort models ("Lari_50c"), and

[4] This is usually called the "Development" set by the speech recognition community. We use "Evaluation" in accordance with the configuration of XM2VTSDB.

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

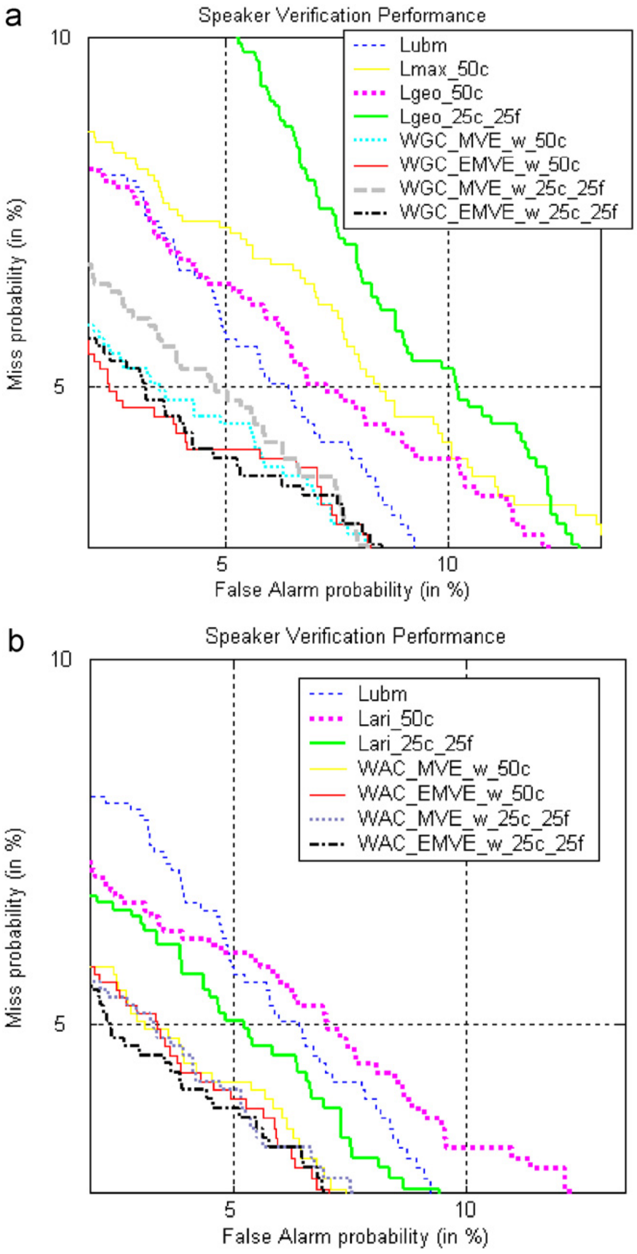

Fig. 3. DET curves for the "Test" subset in Configuration II. (a) Geometric mean versus WGC and (b) arithmetic mean versus WAC.

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

Fig. 3 shows the detection error tradeoff (DET) curves [34] obtained by evaluating the above systems using the "Test" subset, where Fig. 3(a) compares the WGC-based approach and the geometric mean approach, while Fig. 3(b) compares the WAC-based approach and the arithmetic mean approach. From the figure, we observe that all the WGC-based LLR systems outperform the baseline LLR systems "Lubm", "Lmax_50c", "Lgeo_50c", and "Lgeo_25c_25f", while all the WAC-based LLR systems outperform the baseline LLR systems "Lubm", "Lari_50c", and "Lari_25c_25f". From Fig. 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., $p(U|\lambda_i) \rightarrow 0$. In contrast, the results show that the WGC-based LLR systems sidestep this problem with the aid of the weighted strategy. Figs. 3(a) and (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".

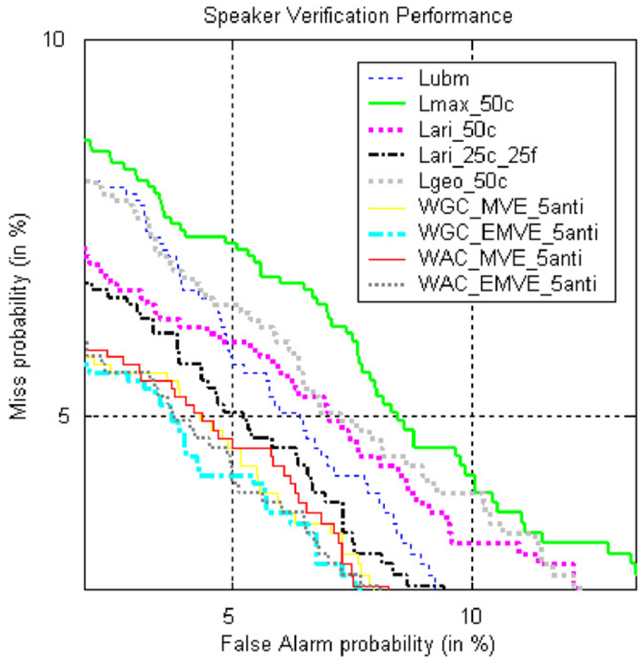

Fig. 4. Hybrid anti-model systems versus all baselines: DET curves for the "Test" subset in Configuration II.

In addition to the above systems, we also evaluated the WAC- and WGC-based LLR measures using the hybrid anti-model defined in Eq. (11). 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. 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 LLR system with a single anti-model.

5.1.2. Discussion

Table 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 LLR measures were optimized automatically using the "Evaluation" subset, and then applied to the "Test" subset.

Several conclusions can be drawn from Table 3. First, all the proposed WAC- and WGC-based LLR systems with either the hybrid anti-model or the background model set (the world model plus a cohort) outperform all the baseline LLR 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 LLR 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".

5.2. *Evaluation based on the NIST SRE-like configuration*

To conform to NIST SRE [6–8], we conducted another series of experiments on XM2VTSDB, which was re-configured as shown Table 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.

Table 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

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. In addition, we implemented two current state-of-the-art systems in the text-independent speaker verification task, namely T-norm [12] and "Lubm_MAP". "Lubm_MAP" is based on the UBM-MAP adaptation method [2]; each client model with 512 mixture Gaussian components was adapted from the UBM via the maximum a posteriori (MAP) estimation [33] 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.

5.2.1. *Experiment results*

As in Section 5.1, we implemented four WGC-based LLR systems: "WGC_MVE_w_50c", "WGC_EMVE_w_50c", "WGC_MVE_w_25c_25f", and "WGC_EMVE_w_25c_25f"; four WAC-based LLR 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 LLR 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 5.1 show that the performance of the proposed WGC- and WAC-based LLR systems using EMVE training is better than that of the systems using gradient-based MVE training, Fig. 5 only compares the performance of the proposed WGC- and WAC-based LLR systems using EMVE training with two state-of-the-art systems and two best baseline systems in Section 5.1, 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 LLR 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 5 summarizes the experiment results for all systems in terms of the DCF. For each baseline system, the decision threshold

Table 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				

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 5, it is clear that all the proposed WGC- and WAC-based LLR systems using either gradient-based MVE training or EMVE training outperform all the conventional LLR 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.

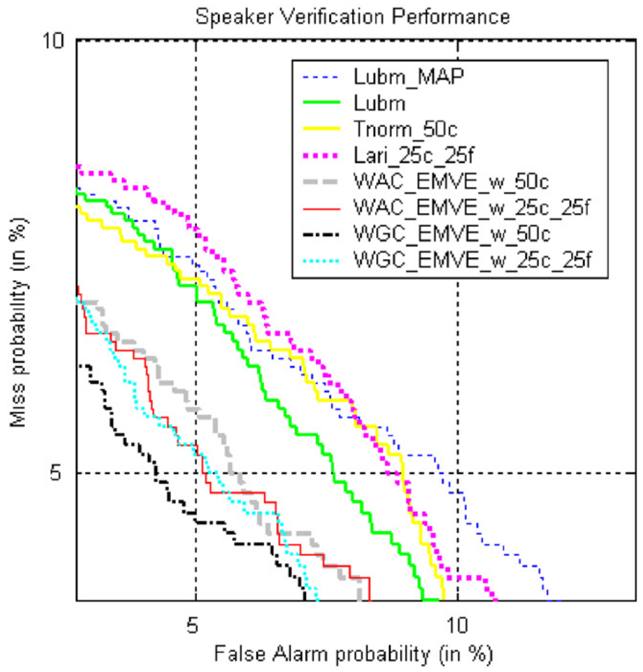

Fig. 5. DET curves for the "Test" subset in the NIST SRE-like configuration.

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 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 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 5 shows the verification time for an input test utterance. The average duration of the test utterances is around 1.5 s. As expected, "Lubm_MAP" is the fastest method, since only one background model (i.e., UBM) is involved and the fast scoring scheme [2] 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.

6. Conclusion

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 parameters associated with WAC or WGC are then optimized using 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 demonstrate that the proposed systems achieve higher verification accuracy than conventional LLR-based approaches. Although they need more training time than conventional LLR-based

Table 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 (s)	Verification time for an input test utterance (s)
Lubm_MAP	0.0704	0.0601		5.79	0.08
Lubm	0.0575	0.0573		7.87	0.12
Tnorm_50c	0.0607	0.0569		27.46	0.75
Lmax_50c	0.0732	0.0734		27.46	0.75
Lari_50c	0.0653	0.0600		27.46	0.75
Lari_25c_25f	0.0611	0.0588		27.46	0.75
Lgeo_50c	0.0758	0.0692		27.46	0.75
WGC_MVE_w_50c	0.0578	0.0529	3 h 06 min 22.31 s	27.46	0.86
WGC_EMVE_w_50c	0.0479	0.0495	3 h 22 min 15.38 s	27.46	0.86
WGC_MVE_w_25c_25f	0.0610	0.0570	3 h 06 min 22.31 s	27.46	0.86
WGC_EMVE_w_25c_25f	0.0485	0.0509	3 h 22 min 15.40 s	27.46	0.86
WAC_MVE_w_50c	0.0575	0.0546	3 h 06 min 25.09 s	27.46	0.86
WAC_EMVE_w_50c	0.0556	0.0533	3 h 24 min 50.14 s	27.46	0.86
WAC_MVE_w_25c_25f	0.0564	0.0549	3 h 06 min 25.09 s	27.46	0.86
WAC_EMVE_w_25c_25f	0.0543	0.0527	3 h 24 min 50.15 s	27.46	0.86
WGC_MVE_5anti	0.0583	0.0541	3 h 06 min 15.58 s	27.46	0.86
WGC_EMVE_5anti	0.0576	0.0514	3 h 09 min 54.53 s	27.46	0.86
WAC_MVE_5anti	0.0610	0.0556	3 h 06 min 15.72 s	27.46	0.86
WAC_EMVE_5anti	0.0587	0.0566	3 h 10 min 15.70 s	27.46	0.86

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. It is worth noting that although we only consider the speaker verification problem in this paper, the proposed framework is not limited to this application. It can be applied to other types of data and hypothesis testing problems.

Acknowledgment

This work was funded in part by the National Science Council, Taiwan, under Grant no.: NSC95-2221-E-001-034.

References

[1] D.A. Reynolds, Speaker identification and verification using Gaussian mixture speaker models, Speech Commun. 17 (1–2) (1995) 91–108.

[2] D.A. Reynolds, T.F. Quatieri, R.B. Dunn, Speaker verification using adapted Gaussian mixture models, Digital Signal Process. 10 (1) (2000) 19–41.

[3] F. Bimbot, J.F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, D.A. Reynolds, A tutorial on text-independent speaker verification, EURASIP J. Appl. Signal Process. 4 (2004) 430–451.

[4] M. Faundez-Zanuy, E. Monte-Moreno, State-of-the-art in speaker recognition, IEEE Aerosp. Electron. Syst. Mag. 20 (5) (2005) 7–12.

[5] B.G.B. Fauve, D. Matrouf, N. Scheffer, J.F. Bonastre, J.S.D. Mason, State-of-the-art performance in text-independent speaker verification through open-source software, IEEE Trans. Audio Speech Lang. Process. 15 (7) (2007) 1960–1968.

[6] M.A. Przybocki, A.F. Martin, A.N. Le, NIST speaker recognition evaluations utilizing the mixer corpora—2004, 2005, 2006, IEEE Trans. Audio Speech Lang. Process. 15 (7) (2007) 1951–1959.

[7] D.A. van Leeuwen, A.F. Martin, M.A. Przybocki, J.S. Bouten, NIST and NFI-TNO evaluations of automatic speaker recognition, Comput. Speech Lang. 20 (2006) 128–158.

[8] ⟨http://www.nist.gov/speech/tests/spk/index.htm⟩.

[9] A.E. Rosenberg, J. DeLong, C.H. Lee, B.H. Juang, F.K. Soong, The use of cohort normalized scores for speaker verification, in: Proceedings of the International Conference on Spoken Language Processing, Banff, Alberta, Canada, 1992, pp. 599–602.

[10] C.S. Liu, H.C. Wang, C.H. Lee, Speaker verification using normalized log-likelihood score, IEEE Trans. Speech Audio Process. 4 (1) (1996) 56–60.

[11] A. Higgins, L. Bahler, J. Porter, Speaker verification using randomized phrase prompting, Digital Signal Process. 1 (2) (1991) 89–106.

[12] R. Auckenthaler, M. Carey, H. Lloyd-Thomas, Score normalization for text-independent speaker verification system, Digital Signal Process. 10 (1) (2000) 42–54.

[13] D.E. Sturim, D.A. Reynolds, Speaker adaptive cohort selection for Tnorm in text-independent speaker verification, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Philadelphia, PA, USA, 2005, pp. 741–744.

[14] W. Chou, B.H. Juang, Pattern Recognition in Speech and Language Processing, CRC Press, Boca Raton, FL, 2003.

[15] A.E. Rosenberg, O. Siohan, S. Parthasarathy, Speaker verification using minimum verification error training, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle, WA, USA, 1998, pp. 105–108.

[16] R.A. Sukkar, A.R. Setlur, M.G. Rahim, C.H. Lee, Utterance verification of keyword strings using word-based minimum verification error (WB-MVE) training, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Atlanta, GA, USA, 1996, pp. 518–521.

[17] M.G. Rahim, C.H. Lee, String based minimum verification error (SB-MVE) training for flexible speech recognition, Comput. Speech Lang. 11 (2) (1997) 147–160.

[18] R.A. Sukkar, Subword-based minimum verification error (SB-MVE) training for task independent utterance verification, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle, WA, USA, 1998, pp. 229–232.

[19] H.K.J. Kuo, C.H. Lee, I. Zitouni, E. Fosler-Lussiert, Minimum verification error training for topic verification, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2003, pp. I-380–I-383.

[20] M.H. Siu, B. Mak, W.H. Au, Minimization of utterance verification error rate as a constrained optimization problem, IEEE Signal Process. Lett. 13 (12) (2006) 760–763.

[21] B.H. Juang, W. Chou, C.H. Lee, Minimum classification error rate methods for speech recognition, IEEE Trans. Speech Audio Process. 5 (3) (1997) 257–265.

[22] O. Siohan, A.E. Rosenberg, S. Parthasarathy, Speaker identification using minimum classification error training, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle, WA, USA, 1998, pp. 109–112.

[23] E. McDermott, T.J. Hazen, J. Le Roux, A. Nakamura, S. Katagiri, Discriminative training for large-vocabulary speech recognition using minimum classification error, IEEE Trans. Audio Speech Lang. Process. 15 (1) (2007) 203–223.

[24] C. Ma, E. Chang, Comparison of discriminative training methods for speaker verification, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2003, pp. I-192–I-195.

[25] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer, Berlin, 2003.

[26] K. Krishna, M.N. Murty, Genetic *K*-means algorithm, IEEE Trans. Syst. Man Cybern. Part B 29 (3) (1999) 433–439.

[27] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, S.J. Brown, FGKA: a fast genetic *K*-means clustering algorithm, in: Proceedings of the ACM Symposium on Applied Computing, 2004, Nicosia, Cyprus, pp. 622–623.

[28] S.S. Cheng, Y.H. Chao, H.M. Wang, H.C. Fu, A prototypes embedded genetic algorithm for *K*-means clustering, in: Proceedings of the 18th International Conference on Pattern Recognition, 2006, Hong Kong, China, pp. 724–727.

[29] K. Messer, J. Matas, J. Kittler, J. Luettin, G. Maitre, XM2VTSDB: The extended M2VTS database, in: Proceedings of the Second International Conference on Audio- and Video-based Biometric Person Authentication, Washington DC, USA, 1999, pp. 72–77.

[30] J. Luettin, G. Maitre, Evaluation protocol for the extended M2VTS database (XM2VTSDB), IDIAP-COM 98-05, IDIAP, 1998.

[31] X. Huang, A. Acero, H.W. Hon, Spoken Language Processing, Prentice Hall, New Jersey, 2001.

[32] J. Lindberg, J. Koolwaaij, H.P. Hutter, D. Genoud, J.B. Pierrot, M. Blomberg, F. Bimbot, Techniques for a priori decision threshold estimation in speaker verification, in: Proceedings RLA2C, Avignon, 1998, pp. 89–92.

[33] J.L. Gauvain, C.H. Lee, Maximum a posteriori estimation for multivariate Gaussian mixture observation of Markov chains, IEEE Trans. Speech Audio Process. 2 (2) (1994) 291–298.

[34] A. Martin, G. Doddington, T. Kamm, M. Ordowski, M. Przybocki, The DET curve in assessment of detection task performance, in: Proceedings of European Conference on Speech Communication and Technology, Rhodes, Greece, 1997, pp. 1895–1898.

About the Author—YI-HSIANG CHAO received the B.S. degree in Information Management from Tatung University, Taipei, Taiwan, in 1999 and the M.S. degree in Computer Science from National Chiao-Tung University, Hsinchu, Taiwan, in 2001. He is currently a Ph.D. candidate in the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. In 2001, he joined the Spoken Language Group, Chinese Information Processing Laboratory, Institute of Information Science, Academia Sinica, Taipei, Taiwan, as a Research Assistant. His research interests include pattern recognition, speech processing, and neural networks.

About the Author—WEI-HO TSAI received the B.S. degree in Electrical Engineering from National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C., in 1995. He received the M.S. and Ph.D. degrees in Communication Engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1997 and 2001, respectively. From 2001 to 2003, he was with Philips Research East Asia, Taipei, Taiwan, where he worked on speech processing problems in embedded systems. From 2003 to 2005, he served as Postdoctoral Fellow at the Institute of Information Science, Academia Sinica, Taipei, Taiwan. He is currently an Assistant Professor at the Department of Electronic Engineering and Graduate Institute of Computer and Communication Engineering, National Taipei University of Technology, Taiwan. His research interests include spoken language processing and music information retrieval.

About the Author—HSIN-MIN WANG received the B.S. and Ph.D. degrees in Electrical Engineering from National Taiwan University, Taipei, Taiwan, in 1989 and 1995, respectively. In October 1995, he joined the Institute of Information Science, Academia Sinica, Taipei, Taiwan, as a Postdoctoral Fellow. He was promoted to Assistant Research Fellow and then Associate Research Fellow in 1996 and 2002, respectively. He was an adjunct associate professor with National Taipei University of Technology and National Chengchi University. He was a board member and chair of academic council of ACLCLP. He currently serves as secretary-general of ACLCLP and as an editorial board member of International Journal of Computational Linguistics and Chinese Language Processing. His major research interests include speech processing, natural language processing, spoken dialogue processing, multimedia information retrieval, and pattern recognition. Dr. Wang was a recipient of the Chinese Institute of Engineers (CIE) Technical Paper Award in 1995. He is a life member of ACLCLP and IICM and a member of ISCA.

About the Author—RUEI-CHUAN CHANG received the B.S. degree in 1979, the M.S. degree in 1981, and his Ph.D. degree in 1984, all in Computer Science from National Chiao Tung University, Hsinchu, Taiwan. In August 1983, he joined the Department of Computer and Information Science at National Chiao Tung University as a Lecturer. Now he is a Professor of the Department of Computer and Information Science. He is also an Associate Research Fellow at the Institute of Information Science, Academia Sinica, Taipei, Taiwan.