Computational Linguistics and Chinese Language Processing
Vol. 12, No. 3, September 2007, pp. 255-272

A Novel Characterization of the Alternative Hypothesis Using Kernel Discriminant Analysis for LLR-Based Speaker Verification

Yi-Hsiang Chao*+ , Hsin-Min Wang* and Ruei-Chuan Chang*+

Abstract

In a log-likelihood ratio (LLR)-based speaker verification system, the alternative hypothesis is usually difficult to characterize a priori, since the model should cover the space of all possible impostors. In this paper, we propose a new LLR measure in an attempt to characterize the alternative hypothesis in a more effective and robust way than conventional methods. This LLR measure can be further formulated as a non-linear discriminant classifier and solved by kernel-based techniques, such as the Kernel Fisher Discriminant (KFD) and Support Vector Machine (SVM). The results of experiments on two speaker verification tasks show that the proposed methods outperform classical LLR-based approaches.

Keywords: Kernel Fisher Discriminant, Log-likelihood Ratio, Speaker Verification, Support Vector Machine.

1. Introduction

In essence, the speaker verification task is a hypothesis testing problem. Given an input utterance U, the goal is to determine whether U was spoken by the hypothesized speaker or not. The log-likelihood ratio (LLR)-based detector [Reynolds 1995] is one of the state-of-the-art approaches for speaker verification. Consider the following hypotheses:

H_0: U is from the hypothesized speaker,

H_1: U is not from the hypothesized speaker.

The LLR test is expressed as:

* Institute of Information Science, Academia Sinica, Taipei, Taiwan
+ Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
E-mail: {yschao,whm}@iis.sinica.edu.tw; rc@cc.nctu.edu.tw

[Received April 3, 2007; Revised August 9, 2007; Accepted August 12, 2007]

$$L(U) = \log \frac{p(U \mid H_0)}{p(U \mid H_1)} \begin{cases} \geq \theta & \text{accept } H_0 \\ < \theta & \text{accept } H_1 \text{ (i.e., reject } H_0 \text{)} , \end{cases} \tag{1}$$

where $p(U \mid H_i),\ i = 0,\ 1,$ is the likelihood of hypothesis H_i given the utterance U, and θ is the threshold. H_0 and H_1 are, respectively, called the null hypothesis and the alternative hypothesis. Mathematically, H_0 and H_1 can be represented by parametric models denoted as λ and $\bar{\lambda}$, respectively; $\bar{\lambda}$ is often called an anti-model. Though H_0 can be modeled straightforwardly using speech utterances from the hypothesized speaker, H_1 does not involve any specific speaker, thus lacks explicit data for modeling. Many approaches have been proposed to characterize H_1, and various LLR measures have been developed. We can formulate these measures in the following general form [Reynolds 2000]:

$$L(U) = \log \frac{p(U \mid \lambda)}{p(U \mid \bar{\lambda})} = \log \frac{p(U \mid \lambda)}{\Psi(p(U \mid \lambda_1), p(U \mid \lambda_2), \ldots, p(U \mid \lambda_N))}, \tag{2}$$

where $\Psi(\cdot)$ is some function of the likelihood values from a set of so-called background models $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$. For example, the background model set can be obtained from N representative speakers, called a cohort [Rosenberg 1992], which simulates potential impostors. If $\Psi(\cdot)$ is an average function [Reynolds 1995], the LLR can be written as:

$$L_1(U) = \log p(U \mid \lambda) - \log \left\{ \frac{1}{N} \sum_{i=1}^{N} p(U \mid \lambda_i) \right\}. \tag{3}$$

Alternatively, the average function can be replaced by various functions, such as the maximum [Liu 1996], *i.e.*:

$$L_2(U) = \log p(U \mid \lambda) - \max_{1 \leq i \leq N} \log p(U \mid \lambda_i), \tag{4}$$

or the geometric mean [Liu 1996], *i.e.*,

$$L_3(U) = \log p(U \mid \lambda) - \frac{1}{N} \sum_{i=1}^{N} \log p(U \mid \lambda_i). \tag{5}$$

A special case arises when $\Psi(\cdot)$ is an identity function and N = 1. In this instance, a single background model is usually trained by pooling all the available data, which is generally irrelevant to the clients, from a large number of speakers. This is called the world model or the Universal Background Model (UBM) [Reynolds 2000]. The LLR in this case becomes:

$$L_4(U) = \log p(U \mid \lambda) - \log p(U \mid \Omega), \tag{6}$$

where Ω denotes the world model.

However, none of the LLR measures developed so far has proven to be absolutely superior to any other, since the selection of Ψ(·) is usually application and training data dependent. In particular, the use of a simple function, such as the average, maximum, or geometric mean, is a heuristic that does not include any optimization process. The issues of selection, size, and combination of background models motivate us to design a more comprehensive function, Ψ(·), to improve the characterization of the alternative hypothesis. In this paper, we first propose a new LLR measure in an attempt to characterize H_1 by integrating all the background models in a more effective and robust way than conventional methods. Then, we formulate this new LLR measure as a non-linear discriminant classifier and apply kernel-based techniques, including the Kernel Fisher Discriminant (KFD) [Mika 1999] and Support Vector Machine (SVM) [Burges 1998], to optimally separate the LLR samples of the null hypothesis from those of the alternative hypothesis.

SVM-based techniques have been successfully applied to many classification and regression tasks, including speaker verification. Unlike our work, existing approaches [Bengio 2001; Wan 2005] only use a single background model, *i.e.*, the world model, to represent the alternative hypothesis, instead of integrating multiple background models to characterize the alternative hypothesis. For example, Bengio *et al.* [Bengio 2001] proposed a decision function:

$$L_5(U) = a_1 \log p(U \mid \lambda) - a_2 \log p(U \mid \Omega) + b, \tag{7}$$

where a_1, a_2, and b are adjustable parameters estimated using SVM. An extended version of Eq. (7) with the Fisher kernel and the LR score-space kernel for SVM was investigated in Wan [Wan 2005].

The results of speaker verification experiments conducted on both the XM2VTS database [Messer 1999] and the ISCSLP2006-SRE database [Chinese Corpus Consortium 2006] show that the proposed methods outperform classical LLR-based approaches. The remainder of this paper is organized as follows. Section 2 describes the design of the new LLR measure in our approach. Sections 3 and 4 introduce the kernel discriminant analysis used in this work and the formation of the characteristic vector by background model selection, respectively. Section 5 contains our experiment results. Finally, in Section 6, we present our conclusions.

2. New LLR Measure Design

2.1 Analysis of the Alternative Hypothesis

First of all, we redesign the function Ψ(·) in Eq. (2) as:

$$p(U \mid \bar{\lambda}) = \Psi(\mathbf{u}) = (p(U \mid \lambda_1)^{w_1} \cdot p(U \mid \lambda_2)^{w_2} \cdot \ldots \cdot p(U \mid \lambda_N)^{w_N})^{1/(w_1+w_2+\ldots+w_N)}, \tag{8}$$

where $\mathbf{u}=[p(U\,|\,\lambda_1), p(U\,|\,\lambda_2),..., p(U\,|\,\lambda_N)]^T$ is an $N\times1$ vector and w_i is the weight of the likelihood $p(U \mid \lambda_i)$, i = 1,2,..., N. This function gives N background models different weights according to their individual contribution to the alternative hypothesis. It is clear that Eq. (8) is equivalent to a geometric mean function when $w_i = 1$, i = 1,2,..., N. If some background model λ_i contrasts with an input utterance U, the likelihood $p(U \mid \lambda_i)$ may be extremely small, thus causing the geometric mean to approximate zero. In contrast, by assigning a favorable weight to each background model, the function $\Psi(\cdot)$ defined in Eq. (8) may be less affected by any specific background model with an extremely small likelihood. Therefore, the resulting score for the alternative hypothesis obtained by Eq. (8) will be more robust and reliable than that obtained by a geometric mean function. It is also clear that Eq. (8) will reduce to a maximum function when $w_{i^*} = 1$, $i^* = \arg\max_{1\le i\le N} \log p(U \mid \lambda_i)$; and $w_i = 0$, $\forall i \neq i^*$.

By substituting Eq. (8) into Eq. (2), we obtain:

$$\begin{aligned}
L_6(U) &= \log\frac{p(U\,|\,\lambda)}{p(U\,|\,\bar{\lambda})} \\
&= \log\frac{p(U\,|\,\lambda)}{(p(U\,|\,\lambda_1)^{w_1}\cdot p(U\,|\,\lambda_2)^{w_2}\cdot...\cdot p(U\,|\,\lambda_N)^{w_N})^{1/(w_1+w_2+...+w_N)}} \\
&= \log\left(\left(\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_1)}\right)^{w_1}\cdot\left(\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_2)}\right)^{w_2}\cdot...\cdot\left(\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_N)}\right)^{w_N}\right)^{1/(w_1+w_2+...+w_N)} \\
&= \frac{1}{w_1+w_2+...+w_N}\left(w_1\log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_1)}+w_2\log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_2)}+...+w_N\log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_N)}\right) \\
&= \frac{1}{w_1+w_2+...+w_N}\mathbf{w}^T\mathbf{x}\ \begin{cases}\ge\theta & \text{accept}\\ <\theta & \text{reject}\end{cases} \\
&= \mathbf{w}^T\mathbf{x}\ \begin{cases}\ge\theta' & \text{accept}\\ <\theta' & \text{reject,}\end{cases}
\end{aligned}\tag{9}$$

where $\mathbf{w}=[w_1, w_2..., w_N]^T$ is an $N\times1$ weight vector, the new threshold $\theta' = (w_1+w_2+...+w_N)\theta$, and $\mathbf{x}$ is an $N \times 1$ vector in the space R^N, expressed by

$$\mathbf{x}=[\log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_1)},\ \log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_2)},...,\log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_N)}]^T. \tag{10}$$

The implicit idea in Eq. (10) is that the speech utterance U can be represented by a characteristic vector $\mathbf{x}$.

If we replace the threshold θ' in Eq. (9) with a bias b, the equation can be rewritten as:

$$L(U)\ = \mathbf{w}^T\mathbf{x}\ + b = f(\mathbf{x}), \tag{11}$$

where $f(\mathbf{x})$ forms a so-called linear discriminant classifier. This classifier translates the goal of solving an LLR measure into the optimization of $\mathbf{w}$ and b, such that the utterances of clients and impostors can be separated. To realize this classifier, three distinct data sets are needed: one for generating each client's model, one for generating the background models, and one for optimizing $\mathbf{w}$ and b. Since the bias b plays the same role as the decision threshold of the conventional LLR measure, which can be determined through a trade-off between false acceptance and false rejection, the main goal here is to find $\mathbf{w}$. Existing linear discriminant analysis techniques, such as Fisher's Linear Discriminant (FLD) [Duda 2001] or Linear SVM [Burges 1998], can be applied to implement Eq. (11).

2.2 Linear Discriminant Analysis

Fisher's Linear Discriminant (FLD) is one of the popular linear discriminant classifiers [Duda 2001]. Suppose the i-th class has n_i data samples, $\mathbf{X}_i = \{\mathbf{x}_1^i,..,\mathbf{x}_{n_i}^i\}$, i = 1, 2. The goal of FLD is to seek a direction $\mathbf{w}$ in the space R^N such that the following Fisher's criterion function $J(\mathbf{w})$ is maximized:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}, \tag{12}$$

where $\mathbf{S}_b$ and $\mathbf{S}_w$ are, respectively, the between-class scatter matrix and the within-class scatter matrix defined as

$$\mathbf{S}_b = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \tag{13}$$

and

$$\mathbf{S}_w = \sum_{i=1,2} \sum_{\mathbf{x} \in \mathbf{X}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T, \tag{14}$$

where $\mathbf{m}_i$ is the mean vector of the i-th class computed by

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{s=1}^{n_i} \mathbf{x}_s^i. \tag{15}$$

According to Duda [Duda 2001], the solution for $\mathbf{w}$, which maximizes $J(\mathbf{w})$ defined in Eq. (12), is the leading eigenvector of $\mathbf{S}_w^{-1}\mathbf{S}_b$.

3. Kernel Discriminant Analysis

Intuitively, $f(\mathbf{x})$ in Eq. (11) can be solved via linear discriminant training algorithms [Duda 2001], such as FLD or Linear SVM. However, such methods are based on the assumption that the observed data of different classes is linearly separable, which is obviously not feasible in most practical cases with nonlinearly separable data. To solve this problem more effectively, we propose using a kernel-based nonlinear discriminant classifier. It is hoped that data from different classes, which is not linearly separable in the original input space R^N, can be separated linearly in a certain higher dimensional (maybe infinite) feature space F via a nonlinear mapping Φ. Let Φ(**x**) denote a vector obtained by mapping **x** from R^N to F. Then, the objective function, based on Eq. (11), can be re-defined as:

$$f(\mathbf{x}) = \mathbf{w}_F{}^T \Phi(\mathbf{x}) + b \ , \tag{16}$$

which constitutes a linear discriminant classifier in F, where $\mathbf{w}_F$ is a weight vector in F.

In practice, it is difficult to determine the kind of mapping that would be applicable; therefore, the computation of Φ(**x**) might be infeasible. To overcome this difficulty, a promising approach is to characterize the relationship between the data samples in F, instead of computing Φ(**x**) directly. This is achieved by introducing a kernel function k(**x**, **y**)=<Φ(**x**),Φ(**y**)>, which is the dot product of two vectors Φ(**x**) and Φ(**y**) in F. The kernel function $k(\cdot)$ must be symmetric, positive definite and conform to Mercer's condition [Burges 1998].

A number of kernel functions exist, such as the simplest dot product kernel function $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$, and the very popular Radial Basis Function (RBF) kernel $k(\mathbf{x}, \mathbf{y}) = \exp(-\ ||\mathbf{x} - \mathbf{y}||^2 / 2\sigma^2)$ in which σ is a tunable parameter. Existing techniques, such as KFD [Mika 1999] or SVM [Burges 1998], can be applied to implement Eq. (16).

3.1 Kernel Fisher Discriminant (KFD)

Suppose the i-th class has n_i data samples, $\mathbf{X}_i = \{\mathbf{x}_1^i, .., \mathbf{x}_{n_i}^i\}$, i = 1, 2. The goal of KFD is to seek a direction $\mathbf{w}_F$ in the feature space F such that the following Fisher's criterion function $J(\mathbf{w}_F)$ is maximized:

$$J(\mathbf{w}_F) = \frac{\mathbf{w}_F{}^T \mathbf{S}_b^\Phi \mathbf{w}_F}{\mathbf{w}_F{}^T \mathbf{S}_w^\Phi \mathbf{w}_F}, \tag{17}$$

where $\mathbf{S}_b^\Phi$ and $\mathbf{S}_w^\Phi$ are, respectively, the between-class scatter matrix and the within-class scatter matrix in F defined as:

$$\mathbf{S}_b^\Phi = (\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)(\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)^T \tag{18}$$

and

$$\mathbf{S}_w^\Phi = \sum_{i=1,2} \sum_{\mathbf{x} \in \mathbf{X}_i} (\Phi(\mathbf{x}) - \mathbf{m}_i^\Phi)(\Phi(\mathbf{x}) - \mathbf{m}_i^\Phi)^T, \tag{19}$$

where $\mathbf{m}_i^\Phi = (1/n_i)\sum_{s=1}^{n_i} \Phi(\mathbf{x}_s^i)$, and i = 1, 2, is the mean vector of the i-th class in F. Let $\mathbf{X}_1 \cup \mathbf{X}_2 = \{\mathbf{x}_1^1, .., \mathbf{x}_{n_1}^1\} \cup \{\mathbf{x}_1^2, .., \mathbf{x}_{n_2}^2\} = \{\mathbf{x}_1, .., \mathbf{x}_l\}$ and $l = n_1 + n_2$. Since the solution of $\mathbf{w}_F$ must lie in the span of all training data samples mapped in *F* [Mika 1999], $\mathbf{w}_F$ can be expressed as:

$$\mathbf{w}_F = \sum_{j=1}^{l} \alpha_j \Phi(\mathbf{x}_j). \tag{20}$$

Let $\boldsymbol{\alpha}^T = [\alpha_1, \alpha_2,..., \alpha_l]$. Accordingly, Eq. (16) can be re-written as:

$$f(\mathbf{x}) = \sum_{j=1}^{l} \alpha_j k(\mathbf{x}_j, \mathbf{x}) + b. \tag{21}$$

Our goal, therefore, changes from finding $\mathbf{w}_F$ to finding $\boldsymbol{\alpha}$, which maximizes

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T \mathbf{M} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \mathbf{N} \boldsymbol{\alpha}}, \tag{22}$$

where **M** and **N** are computed by:

$$\mathbf{M} = (\boldsymbol{\eta}_1 - \boldsymbol{\eta}_2)(\boldsymbol{\eta}_1 - \boldsymbol{\eta}_2)^T \tag{23}$$

and

$$\mathbf{N} = \sum_{i=1,2} \mathbf{K}_i (\mathbf{I}_{n_i} - \mathbf{1}_{n_i}) \mathbf{K}_i^T, \tag{24}$$

respectively, where $\boldsymbol{\eta}_i$ is an $l\times 1$ vector whose *j*-th element $(\boldsymbol{\eta}_i)_j = (1/n_i)\sum_{s=1}^{n_i} k(\mathbf{x}_j, \mathbf{x}_s^i)$, j = 1,2,..., l; $\mathbf{K}_i$ is an $l\times n_i$ matrix with $(\mathbf{K}_i)_{js} = k(\mathbf{x}_j, \mathbf{x}_s^i)$; $\mathbf{I}_{n_i}$ is an $n_i\times n_i$ identity matrix; and $\mathbf{1}_{n_i}$ is an $n_i\times n_i$ matrix with all entries equal to $1/n_i$. Following Mika [Mika 1999], the solution for $\boldsymbol{\alpha}$, which maximizes $J(\boldsymbol{\alpha})$ defined in Eq. (22), is the leading eigenvector of $\mathbf{N}^{-1}\mathbf{M}$.

3.2 Support Vector Machine (SVM)

Alternatively, Eq. (16) can be solved with an SVM, the goal of which is to seek a separating hyperplane in the feature space *F* that maximizes the margin between classes. Following Burges [Burges 1998], $\mathbf{w}_F$ is expressed as:

$$\mathbf{w}_F = \sum_{j=1}^{l} y_j \alpha_j \Phi(\mathbf{x}_j), \tag{25}$$

which yields

$$f(\mathbf{x}) = \sum_{j=1}^{l} y_j \alpha_j k(\mathbf{x}_j, \mathbf{x}) + b, \tag{26}$$

where each training sample $\mathbf{x}_j$ belongs to one of the two classes identified by the label $y_j \in \{-1,1\}$, j=1, 2,..., l. We can find the coefficients α_j by maximizing the objective function,

$$Q(\boldsymbol{\alpha}) = \sum_{j=1}^{l} \alpha_j - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j), \tag{27}$$

subject to the constraints,

$$\sum_{j=1}^{l} y_j \alpha_j = 0, \text{ and } 0 \le \alpha_j \le C_\alpha, \ \forall j, \tag{28}$$

where C_α is a penalty parameter [Burges 1998]. The problem can be solved using quadratic programming techniques [Vapnik 1998]. Note that most α_j are equal to zero, and the training samples associated with non-zero α_j are called *support vectors*. A few support vectors act as the key to deciding the optimal margin between classes in the SVM. An SVM with a dot product kernel function is known as a Linear SVM.

4. Formation of the Characteristic Vector

In our experiments, we use B+1 background models, consisting of B cohort set models and one world model, to form the characteristic vector **x** in Eq. (10); and B cohort set models for $L_1(U)$ in Eq. (3), $L_2(U)$ in Eq. (4), and $L_3(U)$ in Eq. (5). Two cohort selection methods [Reynolds 1995] are used in the experiments. One selects the B closest speakers to each client; and the other selects the B/2 closest speakers to, plus the B/2 farthest speakers from, each client. The selection is based on the speaker distance measure [Reynolds 1995], computed by:

$$d(\lambda_i, \lambda_j) = \log\frac{p(U_i \mid \lambda_i)}{p(U_i \mid \lambda_j)} + \log\frac{p(U_j \mid \lambda_j)}{p(U_j \mid \lambda_i)}, \tag{29}$$

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. Two cohort selection methods yield the following two (B+1)×1 characteristic vectors:

$$\mathbf{x} = \left[\log\frac{p(U \mid \lambda)}{p(U \mid \Omega)} \quad \log\frac{p(U \mid \lambda)}{p(U \mid \lambda_{\text{cst } 1})} \quad \ldots \quad \log\frac{p(U \mid \lambda)}{p(U \mid \lambda_{\text{cst } B})} \right]^T \tag{30}$$

and

$$\mathbf{x} = [\log\frac{p(U|\lambda)}{p(U|\Omega)} \log\frac{p(U|\lambda)}{p(U|\lambda_{\text{cst}1})} \cdots \log\frac{p(U|\lambda)}{p(U|\lambda_{\text{cst}B/2})} \log\frac{p(U|\lambda)}{p(U|\lambda_{\text{fst}1})} \cdots \log\frac{p(U|\lambda)}{p(U|\lambda_{\text{fst}B/2})}]^T , \tag{31}$$

where $\lambda_{\text{cst } i}$ and $\lambda_{\text{fst } i}$ are the i-th closest model and the i-th farthest model of the client model λ, respectively.

5. Experiments

We evaluate the proposed approaches on two databases: the XM2VTS database [Messer 1999] and the ISCSLP2006 speaker recognition evaluation (ISCSLP2006-SRE) database [Chinese Corpus Consortium 2006].

For the performance evaluation, we adopt the Detection Error Tradeoff (DET) curve [Martin 1997]. In addition, the NIST Detection Cost Function (DCF), which reflects the performance at a single operating point on the DET curve, is also used. The DCF is defined as:

$$C_{DET} = C_{Miss} \times P_{Miss} \times P_{T\arg et} + C_{FalseAlarm} \times P_{FalseAlarm} \times (1 - P_{T\arg et}), \tag{32}$$

where P_{Miss} and $P_{FalseAlarm}$ are the miss probability and the false-alarm probability, respectively, C_{Miss} and $C_{FalseAlarm}$ are the respective relative costs of detection errors, and $P_{T\arg et}$ is the *a priori* probability of the specific target speaker. A special case of the DCF is known as the Half Total Error Rate (HTER), where C_{Miss} and $C_{FalseAlarm}$ are both equal to 1, and $P_{T\arg et} = 0.5$, i.e., $\text{HTER} = (P_{Miss} + P_{FalseAlarm})/2$.

5.1 Evaluation on the XM2VTS Database

The first set of speaker verification experiments was conducted on speech data extracted from the XM2VTS database [Messer 1999], which is a multimodal database consisting of face images, video sequences, and speech recordings taken on 295 subjects. The raw database contained approximately 30 hours of digital video recordings, which was then manually annotated. Each subject participated in four recording sessions at approximately one-month intervals, and each recording session consisted of two shots. In a shot, every subject was prompted to read three sentences “0 1 2 3 4 5 6 7 8 9”, “5 0 6 9 2 8 1 3 7 4”, and “Joe took father’s green shoe bench out” at his/her normal pace. The speech was recorded by a microphone clipped to the subject’s shirt.

In accordance with Configuration II of the evaluation protocol described in Luettin [Luettin 1998], the XM2VTS database was divided into three subsets: “Training”, “Evaluation”, and “Test”. In our speaker verification experiments, we used the “Training” subset to build the individual client’s model and the world model[1], and the “Evaluation” subset to estimate the decision threshold θ in Eq. (1) and the parameters $\mathbf{w}$, $\mathbf{w}_F$, and b in

[1] Currently, we do not have an external resource to train the world model and the background models. We follow the evaluation protocol in [Luettin 1998], which suggests “If a world model is needed, as in speaker verification, a client-dependent world model can be trained from all other clients but the actual client. Although not optimal, it is a valid method.” We will train the world model and the background models using an external resource in our future work.

Eq. (11) or Eq. (16). The performance of speaker verification was then evaluated on the "Test" subset. As shown in Table 1, a total of 293 speakers[2] in the database were divided into 199 clients, 25 "evaluation impostors", and 69 "test impostors".

Table 1. Configuration II of the XM2VTS database.

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

We used 12 (2×2×3) utterances/speaker from sessions 1 and 2 to train the individual client's model, represented by a Gaussian Mixture Model (GMM) [Reynolds 1995] with 64 mixture components. For each client, the other 198 clients' utterances from sessions 1 and 2 were used to generate the world model, represented by a GMM with 256 mixture components; 20 or 40 speakers were chosen from these 198 clients as the cohort. Then, 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 estimate θ, $\mathbf{w}$, $\mathbf{w}_F$, and b. However, as a kernel-based classifier can be intractable when a large number of training samples is involved, we reduced the number of impostor samples from 119,400 to 2,250 using a uniform random selection method. In the performance evaluation, we tested 6 utterances/client in session 4 and 24 utterances/test-impostor over the four sessions, which produced 1,194 (6×199) client trials and 329,544 (24×69×199) impostor trials. Table 2 summarizes all the parametric models used in each system.

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

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

Table 2. A summary of the parametric models used in each system for the XM2VTS task.

System	H_0	H_1	
	a 64-mixture client GMM	a 256-mixture world model	*B* 64-mixture cohort GMMs
L_1	√		√
L_2	√		√
L_3	√		√
L_4	√	√	
L_5	√	√	
L_6	√	√	√

5.1.1 Experiment Results

First, *B* was set to 20 in the experiments. We implemented the proposed LLR system based on linear-based classifiers (FLD and Linear SVM) and kernel-based classifiers (KFD and SVM) in eight ways: 1) FLD with Eq. (30) ("FLD_w_20c"), 2) FLD with Eq. (31) ("FLD_w_10c_10f"), 3) Linear SVM with Eq. (30) ("LSVM_w_20c"), 4) Linear SVM with Eq. (31) ("LSVM_w_10c_10f"), 5) KFD with Eq. (30) ("KFD_w_20c"), 6) KFD with Eq. (31) ("KFD_w_10c_10f"), 7) SVM with Eq. (30) ("SVM_w_20c"), and 8) SVM with Eq. (31) ("SVM_w_10c_10f"). Both SVM and KFD used an RBF kernel function with σ= 5. For performance comparison, we used six systems as our baselines: 1) $L_1(U)$ with the 20 closest cohort models ("*L*1_20c"), 2) $L_1(U)$ with the 10 closest cohort models plus the 10 farthest cohort models ("*L*1_10c_10f"), 3) $L_2(U)$ with the 20 closest cohort models ("*L*2_20c"), 4) $L_3(U)$ with the 20 closest cohort models ("*L*3_20c"), 5) $L_4(U)$ ("*L*4"), and 6) $L_5(U)$ using an RBF kernel function with σ= 10 ("*L*5").

Figure 1 shows the results of the baseline systems evaluated on the "Test" subset in DET curves. We observe that the curves "*L*1_10c_10f", "*L*4" and "*L*5" are better than the others. Thus, in the subsequent experiments, we focused on the performance improvements of our proposed LLR systems over these three baselines.

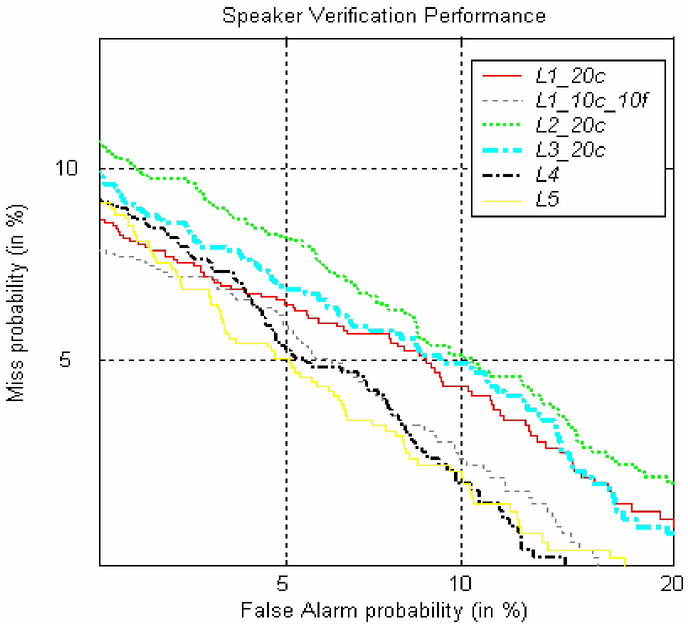

Figure 1. Baselines: DET curves for the XM2VTS "Test" subset (B = 20).

The results of our proposed LLR systems, based on linear-based classifiers and kernel-based classifiers, versus the baseline systems evaluated on the "Test" subset are shown in Figs. 2 and 3, respectively. It is clear that the proposed LLR systems based on either linear-based classifiers or kernel-based classifiers outperform the baseline systems, while KFD perform better than SVM.

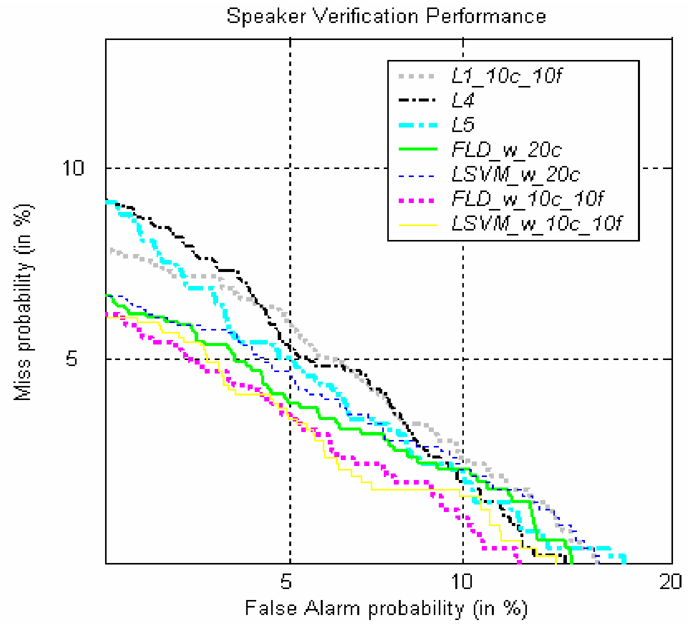

Figure 2. Best baselines vs. our proposed LLR systems based on linear-based classifiers: DET curves for the XM2VTS "Test" subset (B = 20).

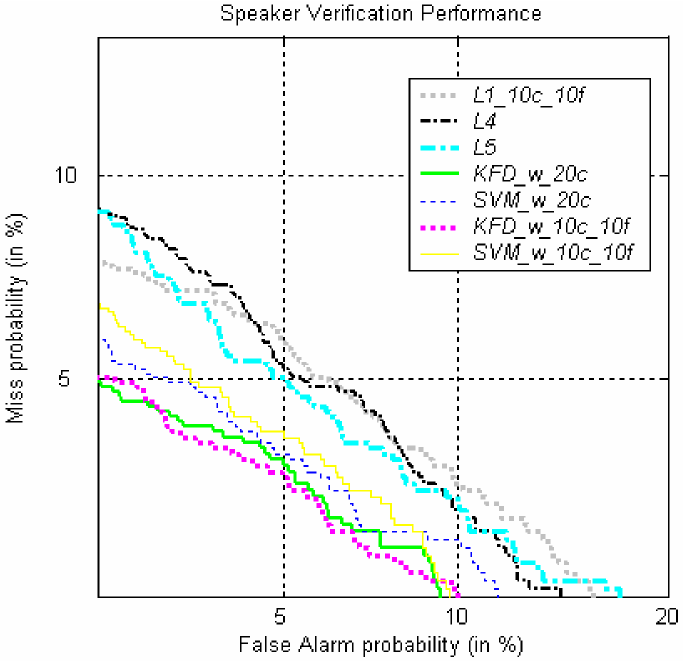

Figure 3. Best baselines vs. our proposed LLR systems based on kernel-based classifiers: DET curves for the XM2VTS "Test" subset (B = 20).

An analysis of the results based on HTER is given in Table 3. For each approach, the decision threshold, θ or b, was used to minimize HTER on the "Evaluation" subset and then applied to the "Test" subset. From Table 3, we observe that all the proposed LLR systems outperform the baseline systems and, for the "Test" subset, a 29.72% relative improvement was achieved by "KFD_w_20c", compared to "*L*5" – the best baseline system. The advantage of integrating multiple background models in our methods could be the reason why the proposed LLR systems based on the linear SVM ("LSVM_w_20c" and "LSVM_w_10c_10f") outperform "*L*5", which applied the kernel-based SVM in $L_5(U)$. We also observe that, in the proposed LLR systems, all of the kernel-based methods outperform the linear-based methods.

To analyze the effect of the number of background models, we implemented several proposed LLR systems and baseline systems with B = 40. An analysis of the results based on the HTER is given in Table 4. Compared to Table 3, the performance of each system with B = 40 is, in general, better than that of its counterpart with B = 20, but not always. For instance, "KFD_w_20c_20f" in Table 4 achieved a lower HTER for "Evaluation" but a higher HTER for "Test", compared to "KFD_w_10c_10f" in Table 3. This may be the result of overtraining. However, from Table 4, it is clear that the superiority of the proposed LLR systems over the baseline systems is again demonstrated.

Table 3. HTERs for the XM2VTS "Evaluation" and "Test" subsets (B = 20).

	min HTER for "Evaluation"	HTER for "Test"
*L*1_20c	0.0676	0.0535
*L*1_10c_10f	0.0589	0.0515
*L*2_20c	0.0776	0.0635
*L*3_20c	0.0734	0.0583
*L*4	0.0633	0.0519
*L*5	0.0590	0.0508
FLD_w_20c	0.0459	0.0433
LSVM_w_20c	0.0472	0.0495
FLD_w_10c_10f	0.0468	0.0455
LSVM_w_10c_10f	0.0453	0.0434
KFD_w_20c	0.0247	0.0357
SVM_w_20c	0.0320	0.0414
KFD_w_10c_10f	0.0232	0.0389
SVM_w_10c_10f	0.0310	0.0417

Table 4. HTERs for the XM2VTS "Evaluation" and "Test" subsets (B = 40).

	min HTER for "Evaluation"	HTER for "Test"
*L*1_40c	0.0675	0.0493
*L*1_20c_20f	0.0589	0.0506
*L*2_40c	0.0765	0.0597
*L*3_40c	0.0722	0.0554
KFD_w_40c	0.0074	0.0345
SVM_w_40c	0.0189	0.0386
KFD_w_20c_20f	0.0050	0.0416
SVM_w_20c_20f	0.0192	0.0403

5.2 Evaluation on the ISCSLP2006-SRE Database

We participated in the text-independent speaker verification task of the ISCSLP2006 Speaker Recognition Evaluation (ISCSLP2006-SRE) plan [Chinese Corpus Consortium 2006]. The database contained 800 clients. Each client has one long training utterance, ranging in duration from 21 to 85 seconds, with an average length of 37.06 seconds. In addition, there are 5,933 utterances in the "Test" subset, each of which ranges in duration from 5 seconds to 54 seconds, with an average length of 15.66 seconds. Each test utterance is associated with the client claimed by the speaker, and the task is to judge whether it is true or false. The ratio of true clients to imposters is approximately 1:20. The answer sheet was released after the evaluation finished.

To form the "Evaluation" subset for estimating θ, $\mathbf{w}$, $\mathbf{w}_F$, and b, we extracted some speech from each client's training utterance in the following way. First, we sorted the 800 clients in descending order according to the length of their training utterances. Then, for the first 100 clients, we cut two 4-second segments from the end of each client's training utterance; however, for the remaining 700 clients, we only cut one 4-second segment from the end of each client's training utterance. This yielded 900 (2×100+700) "Evaluation" utterances. In estimating θ, $\mathbf{w}$, $\mathbf{w}_F$, and b, each "Evaluation" utterance served as a client sample for its associated client, but acted as an imposter sample for each of the remaining 799 clients. This yielded 900 client samples and 719,100 (900×799) impostor samples. We used all the client samples and 2,400 randomly-selected impostor samples to estimate $\mathbf{w}_F$ of the kernel-based classifiers. To determine θ or b, we used the 900 client samples and 18,000 randomly-selected impostor samples. This follows the suggestion in the ISCSLP2006-SRE Plan that the ratio of true clients to imposters in the "Test" subset should be approximately 1:20.

The remaining portion of each client's training utterance was used as "Training" to train that client's model through UBM-MAP adaptation [Reynolds 2000]. This was done by first pooling all the speech in "Training" to train a UBM [Reynolds 2000] with 1,024 mixture Gaussian components, and then adapting the mean vectors of the UBM to each client's GMM according to his/her "Training" utterance.

The signal processing front-end was same as that applied in the XM2VTS task.

5.2.1 Experiment Results

The GMM-UBM [Reynolds 2000] system is the current state-of-the-art approach for the text-independent speaker verification task. Thus, in this part, we focus on the performance improvements of our methods over the baseline GMM-UBM system.

As with the GMM-UBM system, we used the fast scoring method [Reynolds 2000] for likelihood ratio computation in the proposed methods. Both the client model λ and the B cohort models were adapted from the UBM Ω. Since 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, the value of C was set to 5.

B was set to 100 in the experiments. We implemented the proposed LLR system in four ways: 1) KFD with Eq. (30) ("KFD_w_100c"), 2) KFD with Eq. (31) ("KFD_w_50c_50f"), 3) SVM with Eq. (30) ("SVM_w_100c"), and 4) SVM with Eq. (31) ("SVM_w_50c_50f"). We compared the proposed systems with the baseline GMM-UBM system and Bengio *et al.*'s system (*L*5). Figure 4 shows the results of experiments conducted on 5,933 "Test" utterances in DET curves. The proposed LLR systems clearly outperform the baseline GMM-UBM system and Bengio *et al.*'s system (*L*5). According to the ISCSLP2006 SRE plan, the performance is measured by the NIST DCF with $C_{Miss}=10$, $C_{FalseAlarm}=1$, and $P_{T\arg et}=0.05$. In each system, the decision threshold, θ or b, was selected to minimize the DCF on the "Evaluation" subset, and then applied to the "Test" subset. The minimum DCFs for the "Evaluation" subset and the associated DCFs for the "Test" subset are given in Table 5. We observe that "KFD_w_50c_50f" achieved a 34.08% relative improvement over "GMM-UBM", and a 19.73% relative improvement over "*L*5".

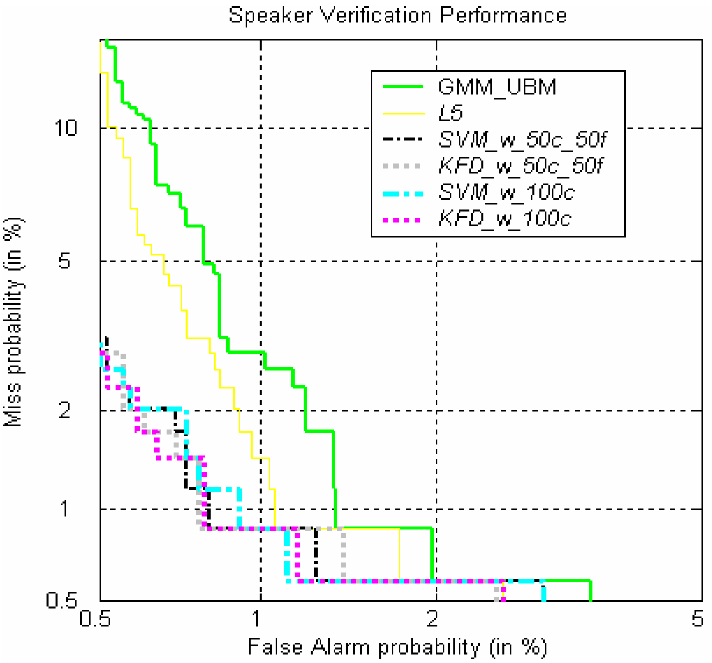

Figure 4. DET curves for the ISCSLP2006-SRE "Test" subset.

Table 5. DCFs for the ISCSLP2006-SRE "Evaluation" and "Test" subsets.

	min DCF for "Evaluation"	DCF for "Test"
GMM-UBM	0.0129	0.0179
*L*5	0.0120	0.0147
KFD_w_50c_50f	0.0067	0.0118
SVM_w_50c_50f	0.0067	0.0123
KFD_w_100c	0.0063	0.0145
SVM_w_100c	0.0076	0.0142

6. Conclusions

We have presented a new LLR measure for speaker verification that improves the characterization of the alternative hypothesis by integrating multiple background models in a more effective and robust way than conventional methods. This new LLR measure is formulated as a non-linear classification problem and solved by using kernel-based classifiers, namely, the Kernel Fisher Discriminant and Support Vector Machine, to optimally separate the LLR samples of the null hypothesis from those of the alternative hypothesis. Experiments, in which the proposed methods were applied to two speaker verification tasks, showed notable improvements in performance over classical LLR-based approaches. Finally, it is worth noting that the proposed methods can be applied to other types of data and hypothesis testing problems.

References

Bengio, S. and J. Mariéthoz, “Learning the Decision Function for Speaker Verification,” In *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, 2001, Salt Lake City, USA, pp. 425-428.

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

Chinese Corpus Consortium (CCC), “Evaluation Plan for ISCSLP’2006 Special Session on Speaker Recognition,” 2006.

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

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

Liu, C. S., H. C. Wang and C. H. Lee, “Speaker Verification Using Normalized Log-Likelihood Score,” *IEEE Trans. Speech and Audio Processing*, 4, 1996, pp.56-60.

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

Martin, A., G. Doddington, T. Kamm, M. Ordowski and M. Przybocki, “The DET Curve in Assessment of Detection Task Performance,” In *Proceedings of Eurospeech*, 1997, pp. 1895-1898.

Messer, K., J. Matas, J. Kittler, J. Luettin, and G. Maitre, “XM2VTSDB: The Extended M2VTS Database,” In *Proceedings of the 2nd International Conference on Audio and Video-based Biometric Person Authentication*, 1999, Washington D. C., USA, pp. 72-77.

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

Reynolds, D. A., “Speaker Identification and Verification Using Gaussian Mixture Speaker Models,” *Speech Communication*, 17, 1995, pp. 91-108.

Reynolds, D. A., T. F. Quatieri and R. B. Dunn, “Speaker Verification Using Adapted Gaussian Mixture Models,” *Digital Signal Processing*, 10, 2000, pp. 19-41.

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 *Proceedings of International Conference on Spoken Language Processing*, 1992, Banff, Canada, pp. 599-602.

Wan ,V. and S. Renals, “Speaker Verification Using Sequence Discriminant Support Vector Machines,” *IEEE Trans. Speech and Audio Processing*, 13(2), 2005, pp. 203-210.

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