A Novel Alternative Hypothesis Characterization Using Kernel Classifiers for LLR-based Speaker Verification

Yi-Hsiang Chao[1,2], Hsin-Min Wang[1], and Ruei-Chuan Chang[1,2]

[1] Institute of Information Science, Academia Sinica, Taipei, Taiwan
[2] Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
{yschao, whm}@iis.sinica.edu.tw, rc@cc.nctu.edu.tw

Abstract. In a log-likelihood ratio (LLR)-based speaker verification system, the alternative hypothesis is usually ill-defined and hard to characterize a priori, since it 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: Speaker verification, Log-likelihood ratio, Kernel Fisher Discriminant, 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 [1] detector 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

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

where $p(U \mid H_i),\ i = 0, 1,$ is the likelihood of the 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, and 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 [2]:

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

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 [8], which simulates potential impostors. If $\Psi(\cdot)$ is an average function [1], 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 [3], i.e.,

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

or the geometric mean [4], 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) [2]. 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 proved to be absolutely superior to any other, since the selection of $\Psi(\cdot)$ 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 involve any optimization process. The issues of selection, size, and combination of background models motivate us to design a more comprehensive function, $\Psi(\cdot)$, 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) [6] and Support Vector Machine (SVM) [7], to optimally separate the LLR samples of the null hypothesis from those of the alternative hypothesis. Speaker verification experiments conducted on both the XM2VTSDB database and the ISCSLP2006 speaker recognition evaluation database show that the proposed methods outperform classical LLR-based approaches.

The remainder of this paper is organized as follows. Section 2 describes the analysis of the alternative hypothesis in our approach. Sections 3 and 4 introduce the kernel classifiers 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 Analysis of the Alternative Hypothesis

First of all, we redesign the function $\Psi(\cdot)$ in Eq. (2) as

$$\Psi(\mathbf{u}) = (p(U\,|\,\lambda_1)^{\alpha_1} \cdot p(U\,|\,\lambda_2)^{\alpha_2} \cdot \ldots \cdot p(U\,|\,\lambda_N)^{\alpha_N})^{1/(\alpha_1+\alpha_2+\ldots+\alpha_N)}, \tag{7}$$

where $\mathbf{u} = [p(U\,|\,\lambda_1), p(U\,|\,\lambda_2), \ldots, p(U\,|\,\lambda_N)]^T$ is an $N\times 1$ vector and α_i is the weight of the likelihood $p(U\,|\,\lambda_i)$, $i = 1,2,\ldots, N$. This function gives N background models different weights according to their individual contribution to the alternative hypothesis. It is clear that Eq. (7) is equivalent to a geometric mean function when $\alpha_i = 1$, $i = 1,2,\ldots, N$. If some background model λ_i contrasts with an input utterance U, the likelihood $p(U\,|\,\lambda_i)$ may be extremely small, and thus cause the geometric mean to approximate zero. In contrast, by assigning a favorable weight to each background model, the function $\Psi(\cdot)$ defined in Eq. (7) 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. (7) will be more robust and reliable than that obtained by a geometric mean function. It is also clear that Eq. (7) will reduce to a maximum function when $\alpha_{i^*} = 1$, $i^* = \arg\max_{1\le i\le N} \log p(U\,|\,\lambda_i)$; and $\alpha_i = 0$, $\forall i \ne i^*$.

By substituting Eq. (7) into Eq. (2) and letting $w_i = \alpha_i / (\alpha_1 + \alpha_2 + \ldots + \alpha_N)$, $i = 1,2,\ldots, N$, we obtain

$$\begin{aligned} L(U) &= \log \frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_1)^{w_1} \cdot p(U\,|\,\lambda_2)^{w_2} \cdot \ldots \cdot p(U\,|\,\lambda_N)^{w_N}} \\ &= \log\left(\left(\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_1)}\right)^{w_1} \cdot \left(\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_2)}\right)^{w_2} \cdot \ldots \cdot \left(\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_N)}\right)^{w_N} \right) \\ &= w_1 \log \frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_1)} + w_2 \log \frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_2)} + \ldots + w_N \log \frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_N)} \\ &= \mathbf{w}^T \mathbf{x} \begin{cases} \ge \theta & \text{accept} \\ < \theta & \text{reject,} \end{cases} \end{aligned} \tag{8}$$

where $\mathbf{w} = [w_1, w_2 \ldots, w_N]^T$ is an $N\times 1$ weight vector and $\mathbf{x}$ is an $N \times 1$ vector in the space R^N, expressed by

$$\mathbf{x} = [\log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_1)}, \log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_2)}, ..., \log\frac{p(U\,|\,\lambda)}{p(U\,|\,\lambda_N)}]^T. \quad \textbf{(9)}$$

The implicit idea in Eq. (9) is that the speech utterance *U* can be represented by a characteristic vector **x**.

If we replace the threshold θ in Eq. (8) with a bias *b*, the equation can be rewritten as

$$L(U) = \mathbf{w}^T \mathbf{x} + b = f(\mathbf{x}), \quad \textbf{(10)}$$

where *f*(**x**) forms a so-called linear discriminant classifier. This classifier translates the goal of solving an LLR measure into the optimization of **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 **w** and *b*. Since the bias *b* plays the same role as the decision threshold θ of the conventional LLR-based detector defined in Eq. (1), which can be determined through a trade-off between false acceptance and false rejection, the main goal here is to find **w**.

3 Kernel Classifiers

Intuitively, *f*(**x**) in Eq. (10) can be solved via linear discriminant training algorithms [9]. 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. (10), can be re-defined as

$$f(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x}) + b, \quad \textbf{(11)}$$

which constitutes a linear discriminant classifier 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*(·) must be symmetric, positive definite and conform to Mercer's condition [7]. A number of kernel functions exist, such as the simplest dot product kernel function *k*(**x**, **y**) = $\mathbf{x}^T\mathbf{y}$, and the very popular Radial Basis Function (RBF) kernel *k*(**x**, **y**) = $\exp(-\,\|\mathbf{x}-\mathbf{y}\|^2 / 2\,\sigma^2)$ in which σ is a tunable parameter. Existing techniques, such as KFD [6] or SVM [7], can be applied to implement Eq. (11).

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 the KFD is to find a direction **w** in the feature space *F* such that the following Fisher's criterion function *J*(**w**) is maximized:

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

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

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

and

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

where $\mathbf{m}_i^\Phi = (1/n_i)\Sigma_{s=1}^{n_i}\Phi(\mathbf{x}_s^i)$, and *i* = 1, 2, is the mean vector of the *i*-th class in *F*. Let $\mathbf{X}_1 \cup \mathbf{X}_2 = \{\mathbf{x}_1, \mathbf{x}_2,..,\mathbf{x}_l\}$ and $l = n_1 + n_2$. Since the solution of **w** must lie in the span of all training data samples mapped in *F* [6], **w** can be expressed as

$$\mathbf{w} = \sum_{j=1}^{l} \alpha_j \Phi(\mathbf{x}_j). \qquad (15)$$

Let $\boldsymbol{\alpha}^T = [\alpha_1, \alpha_2, \ldots, \alpha_l]$. Accordingly, Eq. (11) can be re-written as

$$f(\mathbf{x}) = \sum_{j=1}^{l} \alpha_j k(\mathbf{x}_j, \mathbf{x}) + b. \qquad (16)$$

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

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T \mathbf{M} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \mathbf{N} \boldsymbol{\alpha}}, \qquad (17)$$

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

$$\mathbf{M} = (\boldsymbol{\eta}_1 - \boldsymbol{\eta}_2)(\boldsymbol{\eta}_1 - \boldsymbol{\eta}_2)^T \qquad (18)$$

and

$$\mathbf{N} = \sum_{i=1,2} \mathbf{K}_i (\mathbf{I}_{n_i} - \mathbf{1}_{n_i}) \mathbf{K}_i^T, \qquad (19)$$

respectively, where $\boldsymbol{\eta}_i$ is an *l*×1 vector with $(\boldsymbol{\eta}_i)_j = (1/n_i)\Sigma_{s=1}^{n_i} k(\mathbf{x}_j, \mathbf{x}_s^i)$, $\mathbf{K}_i$ is an *l* ×n_i matrix with $(\mathbf{K}_i)_{js} = k(\mathbf{x}_j, \mathbf{x}_s^i)$, $\mathbf{I}_{n_i}$ is an n_i×n_i identity matrix, and $\mathbf{1}_{n_i}$ is an n_i×n_i matrix with all entries equal to $1/n_i$. Following [6], the solution for $\boldsymbol{\alpha}$, which maximizes $J(\boldsymbol{\alpha})$ defined in Eq. (17), is the leading eigenvector of $\mathbf{N}^{-1}\mathbf{M}$.

3.2 Support Vector Machine (SVM)

Alternatively, Eq. (11) 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 [7], $\mathbf{w}$ is expressed as

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

which yields

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

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,

$$Q(\boldsymbol{\alpha}) = \sum_{j=1}^{l} \alpha_j - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j), \tag{22}$$

subject to the constraints,

$$\sum_{j=1}^{l} y_j \alpha_j = 0, \text{ and } 0 \le \alpha_j \le C, \ \forall j, \tag{23}$$

where C is a penalty parameter [7]. The problem can be solved using quadratic programming techniques [10]. 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 $\mathbf{x}$ in Eq. (9); 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 [1] 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 [1], computed by

$$d(\lambda_i, \lambda_j) = \log \frac{p(X_i \mid \lambda_i)}{p(X_i \mid \lambda_j)} + \log \frac{p(X_j \mid \lambda_j)}{p(X_j \mid \lambda_i)}, \tag{24}$$

where λ_i and λ_j are speaker models trained using the *i*-th speaker's utterances X_i and the *j*-th speaker's utterances X_j, respectively. Two cohort selection methods yield the following two (*B*+1)x1 characteristic vectors:

$$\mathbf{x} = [\tilde{p}_0(U)\ \tilde{p}_1^c(U)\ \ldots\ \tilde{p}_B^c(U)]^T \tag{25}$$

and

$$\mathbf{x} = [\tilde{p}_0(U)\ \tilde{p}_1^c(U)\ \ldots\ \tilde{p}_{B/2}^c(U)\ \tilde{p}_1^f(U)\ \ldots\ \tilde{p}_{B/2}^f(U)]^T, \tag{26}$$

where $\tilde{p}_0(U) = \log p(U \mid \lambda) / p(U \mid \Omega)$, $\tilde{p}_i^c(U) = \log p(U \mid \lambda) / p(U \mid \lambda_{\text{closest } i})$, and $\tilde{p}_i^f(U) = \log\ p(U \mid \lambda) / p(U \mid \lambda_{\text{farthest } i})$. $\lambda_{\text{closest } i}$ and $\lambda_{\text{farthest } i}$ 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 XM2VTSDB database [11] and the ISCSLP2006 speaker recognition evaluation (ISCSLP2006-SRE) database [12].

For the performance evaluation, we adopt the Detection Error Tradeoff (DET) curve [13]. In addition, the NIST Detection Cost Function (DCF) [14], which reflects the performance at a single operating point on the DET curve, is also used. The DCF is defined as

$$C_{DET} = C_{Miss} \times P_{Miss} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm} \times (1 - P_{Target}), \tag{27}$$

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_{Target} 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 $P_{Target} = 0.5$, i.e., $\text{HTER} = (P_{Miss} + P_{FalseAlarm}) / 2$.

5.1 Evaluation on the XM2VTSDB Database

The first set of speaker verification experiments was conducted on speech data extracted from the XM2VTSDB multi-modal database [11]. In accordance with "Configuration II" described in [11], the database was divided into three subsets: "Training", "Evaluation", and "Test". In our experiments, we used the "Training" subset to build the individual client's model and the world model, and the "Evaluation" subset to estimate the decision threshold θ in Eq. (1) and the parameters **w** and *b* in Eq. (11). The performance of speaker verification was then evaluated on the "Test" subset. As shown in Table 1, a total of 293 speakers[1] in the database were divided into 199 clients, 25 "evaluation impostors", and 69 "test impostors". Each speaker participated in four recording sessions at approximately one-month intervals, and each recording session consisted of two shots. In a shot, every speaker was prompted to utter 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". Using a 32-ms Hamming-windowed frame with 10-ms shifts, each utterance (sampled at 32 kHz) was converted into a stream of 24-order feature vectors, each consisting of 12 Mel-scale cepstral coefficients [5] and their first time derivatives.

Table 1. Configuration of the XM2VTSDB speech database.

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			

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) [1] 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 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, which yielded 1,194 (6×199) client samples and 119,400 (24×25×199) impostor samples, to estimate θ, $\mathbf{w}$, and b. However, because 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, which produced 1,194 (6×199) client trials and 329,544 (24×69×199) impostor trials.

5.1.1 Experiment Results

We implemented the proposed LLR system in four ways: KFD with Eq. (25) ("KFD_w_20c"), KFD with Eq. (26) ("KFD_w_10c_10f"), SVM with Eq. (25) ("SVM_w_20c"), and SVM with Eq. (26) ("SVM_w_10c_10f"). Both SVM and KFD used an RBF kernel function with $\sigma = 5$. For the performance comparison, we used five systems as our baselines: 1) $L_1(U)$ with the 20 closest cohort models

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

("L1_20c"), 2) $L_1(U)$ with the 10 closest cohort models plus the 10 farthest cohort models ("L1_10c_10f"), 3) $L_2(U)$ with the 20 closest cohort models ("L2_20c"), 4) $L_3(U)$ with the 20 closest cohort models ("L3_20c"), and 5) $L_4(U)$ ("L4").

Fig. 1 shows the results of the baseline systems tested on the "Evaluation" subset in DET curves [13]. We observe that the curves "L1_10c_10f" and "L4" are better than the others. Thus, in the second experiment, we focused on the performance improvements of our proposed LLR systems over these two baselines.

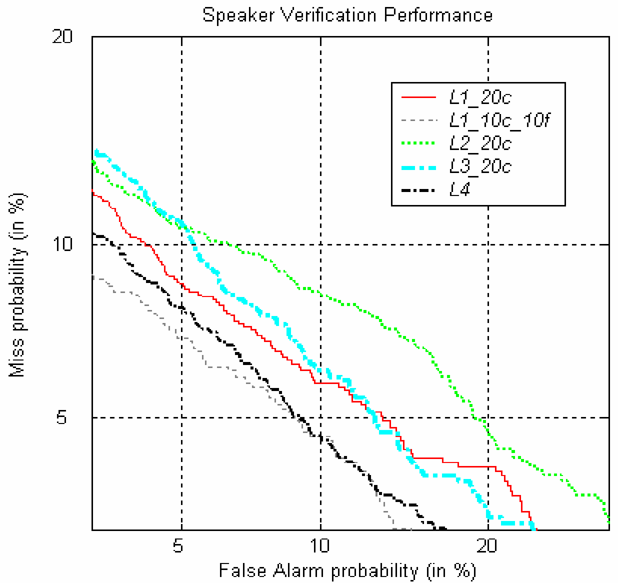

Fig. 1. Baselines: DET curves for the XM2VTSDB "Evaluation" subset.

Fig. 2 shows the results of our proposed LLR systems versus the baseline systems evaluated on the "Test" subset. It is clear that the proposed LLR systems, including KFD and SVM, outperform the baseline LLR systems, while KFD performs better than SVM.

An analysis of the results based on the HTER is given in Table 2. For each approach, the decision threshold, θ or b, was used to minimize the HTER on the "Evaluation" subset, and then applied to the "Test" subset. From Table 2, we observe that, for the "Test" subset, a 30.68% relative improvement was achieved by "KFD_w_20c", compared to "L1_10c_10f" – the best baseline system.

5.2 Evaluation on the ISCSLP2006-SRE Database

We participated in the text-independent speaker verification task of the ISCSLP2006 Speaker Recognition Evaluation (SRE) plan [12]. The database, which was provided by Chinese Corpus Consortium (CCC) [15], contained 800 clients. The length of the training data for each client ranged from 21 seconds to 1 minute and 25 seconds; the average length was approximately 37.06 seconds.

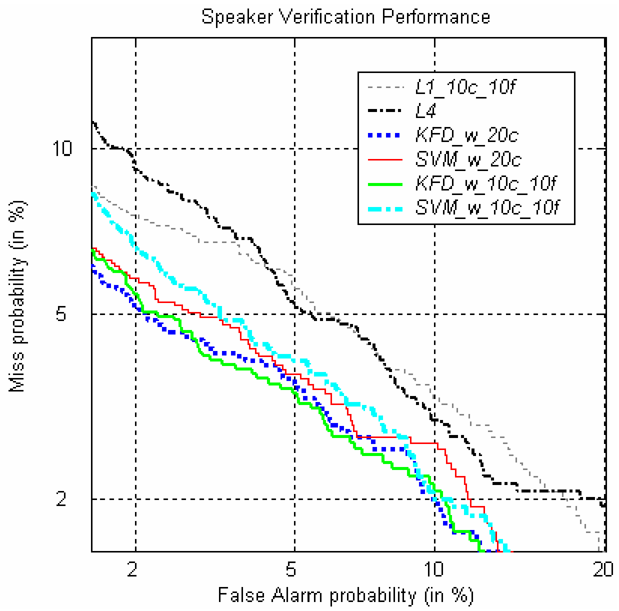

Fig. 2. Best baselines vs. our proposed LLR systems: DET curves for the XM2VTSDB "Test" subset.

Table 2. HTERs for "Evaluation" and "Test" subsets (The XM2VTSDB task).

	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
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

We sorted the clients according to the length of their training data in descending order. For the first 100 clients, we cut two 4-second segments from the end; and for the remaining 700 clients, we cut one 4-second segment from the end, as the "Evaluation" data to estimate θ, **w**, and *b*. For each client, the remaining training data was used for "Training" to build that client's model. In the implementation, all the "Training" data was pooled to train a UBM [2] with 1,024 mixture components. Then, the mean vectors of each client's GMM were adapted from the UBM by his/her "Training" data. In the evaluation stage, each client was treated as an "evaluation impostor" of the other 799 clients. In this way, we had 900 (2×100+700) client samples and 719,100 (900×799) impostor samples. We applied all the client samples and 2,400 randomly selected impostor samples to estimate **w** of the kernel classifiers.

According to the evaluation plan, the ratio of true clients to imposters in the "Test" subset should be approximately 1:20. Therefore, we applied the 900 client samples and 18,000 randomly selected impostor samples to estimate the decision threshold, θ or *b*. The "Test" data consisted of 5,933 utterances.

The signal processing front-end was same as that applied in the XM2VTSDB task.

5.2.1 Experiment Results

Fig. 3 shows the results of the proposed LLR system using KFD with Eq. (26) and *B* = 100 ("KFD_w_50c_50f") versus the baseline GMM-UBM [2] system tested on 5,933 "Test" utterances in DET curves. The proposed LLR system clearly outperforms the baseline GMM-UBM system. According to the ISCSLP2006 SRE plan, the performance is measured by the NIST DCF with $C_{Miss} = 10$, $C_{FalseAlarm} = 1$, and $P_{Target} = 0.05$. In each system, the decision threshold, θ or *b*, was selected to minimize the DCF on the "Evaluation" data, and then applied to the "Test" data. The minimum DCFs for the "Evaluation" data and the associated DCFs for the "Test" data are given in Table 3. We observe that "KFD_w_50c_50f" achieved a 34.08% relative improvement over "GMM-UBM".

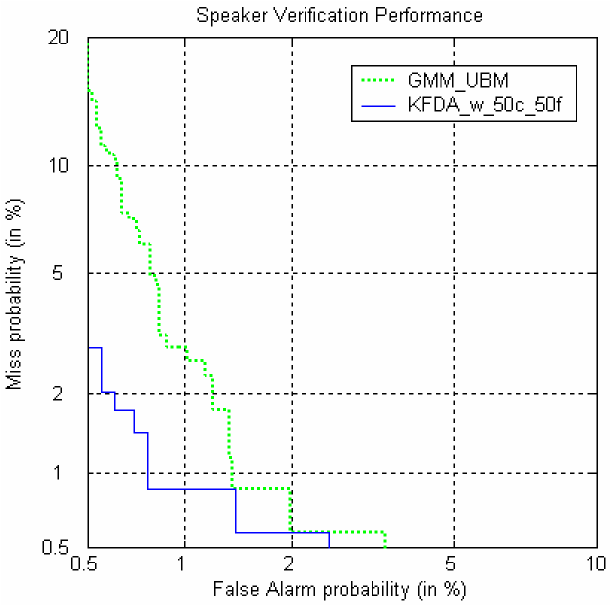

Fig. 3. DET curves for the ISCSLP2006-SRE "Test" subset.

Table 3. DCFs for "Evaluation" and "Test" subsets (The ISCSLP2006-SRE task).

	min DCF for "Evaluation"	DCF for "Test"
GMM-UBM	0.0129	0.0179
KFD_w_50c_50f	0.0067	0.0118

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.

Acknowledgments. This work was funded by the National Science Council, Taiwan, under Grant: NSC94-2213-E-001-009.

References

1. Reynolds, D. A.: Speaker Identification and Verification using Gaussian Mixture Speaker Models. Speech Communication, Vol.17. (1995) 91-108
2. Reynolds, D. A., Quatieri, T. F., Dunn, R. B.: Speaker Verification using Adapted Gaussian Mixture Models. Digital Signal Processing, Vol. 10. (2000) 19-41
3. Higgins, A., Bahler, L., Porter, J.: Speaker Verification using Randomized Phrase Prompting. Digital Signal Processing, Vol. 1. (1991) 89-106
4. Liu, C. S., Wang, H. C., Lee, C. H.: Speaker Verification using Normalized Log-Likelihood Score. IEEE Trans. Speech and Audio Processing, Vol. 4. (1996) 56-60
5. Huang, X., Acero, A., Hon, H. W.: Spoken Language Processing. Prentics Hall, New Jersey (2001)
6. Mika, S., Rätsch, G., Weston, J. Schölkopf, B., Müller, K. R.: Fisher Discriminant Analysis with Kernels. Neural Networks for Signal Processing IX. (1999) 41-48
7. Burges, C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, Vol.2. (1998) 121-167
8. Rosenberg, A. E., Delong, J., Lee, C. H., Juang, B. H., Soong, F. K.: The use of Cohort Normalized Scores for Speaker Verification. Proc. ICSLP (1992)
9. Duda, R. O., Hart, P. E., Stork, D. G.: Pattern Classification. 2nd edn. John Wiley & Sons, New York (2001)
10. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, New York (1998)
11. Luettin, J., Maître, G.: Evaluation Protocol for the Extended M2VTS Database (XM2VTSDB). IDIAP-COM 98-05, IDIAP (1998)
12. Chinese Corpus Consortium (CCC): Evaluation Plan for ISCSLP'2006 Special Session on Speaker Recognition (2006)
13. Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M.: The DET Curve in Assessment of Detection Task Performance. Proc. Eurospeech (1997)
14. http://www.nist.gov/speech/tests/spk/index.htm
15. http://www.CCCForum.org