Improving the Characterization of the Alternative Hypothesis via Kernel Discriminant Analysis for Likelihood Ratio-based Speaker Verification

Yi-Hsiang Chao[1,2], *Wei-Ho Tsai*[3], *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
[3] Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan
{yschao,whm}@iis.sinica.edu.tw, whtsai@en.ntut.edu.tw, rc@cc.nctu.edu.tw

Abstract

The performance of a likelihood ratio-based speaker verification system is highly dependent on modeling of the target speaker's voice (the null hypothesis) and characterization of non-target speakers' voices (the alternative hypothesis). To better characterize the ill-defined alternative hypothesis, this study proposes a new likelihood ratio measure based on a composite-structure Gaussian mixture model, the so-called GMM2. Motivated by the combined use of a variety of background models to represent the alternative hypothesis, GMM2 is designed with an inner set of mixture weights connected to the significance of each individual Gaussian density, and an outer set of mixture weights connected to the significance of each individual background model. Through the use of kernel discriminant analysis namely, Kernel Fisher Discriminant (KFD) or Support Vector Machine (SVM), GMM2 is trained in such a manner that the utterances of the null hypothesis can be optimally separated from those of the alternative hypothesis.

Index Terms: speaker verification, likelihood ratio, kernel Fisher discriminant, support vector machine

1. Introduction

In essence, speaker verification is a hypothesis testing problem that is commonly solved by using a likelihood ratio (LR) test [1]. Given an input utterance *U*, the goal is to determine whether or not *U* was spoken by the target (hypothesized) speaker. Consider the following two hypotheses:

$$\begin{aligned} H_0 &: U \text{ is from the target speaker,} \\ H_1 &: U \text{ is not from the target speaker.} \end{aligned} \qquad (1)$$

The LR test can be expressed as

$$L(U) = \frac{p(U \mid H_0)}{p(U \mid H_1)} \begin{cases} \geq \theta & \text{accept } H_0 \\ < \theta & \text{accept } H_1 \text{ (reject } H_0 \text{)}, \end{cases} \qquad (2)$$

where $p(U \mid H_i)$, $i = 0,1$, is the likelihood of hypothesis H_i given the utterance *U*, and θ is a threshold. H_0 and H_1 are called the null hypothesis and the alternative hypothesis, respectively. Mathematically, H_0 and H_1 can be characterized by some parametric models, such 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 target speaker, H_1 does not involve any specific speaker, and thus lacks explicit data for modeling. The approaches that have been proposed to better characterize H_1 can be collectively expressed in the following form [2]:

$$p(U \mid \bar{\lambda}) = \Psi(p(U \mid \lambda_1), ..., p(U \mid \lambda_N)), \qquad (3)$$

where Ψ() is a function of the likelihoods computed for a set of background models {λ_1, λ_2,…, λ_N}. For example, the background model set can be obtained from *N* representative speakers, called a cohort set [8], which simulates potential impostors. If Ψ() is an average function [1], the LR is computed using

$$L_1(U) = \log p(U \mid \lambda) - \log\left\{\frac{1}{N}\sum_{i=1}^{N} p(U \mid \lambda_i)\right\}. \qquad (4)$$

Alternatively, the average function can be replaced by various functions, such as the maximum [3] and the geometric mean [4]. A special case arises when *N* = 1, in which a single background model is usually trained by pooling all the available data from a large number of speakers. This is called the world model [2]. The LR in this case becomes

$$L_2(U) = \log p(U \mid \lambda) - \log p(U \mid \Omega), \qquad (5)$$

where Ω denotes the world model.

However, none of the LR measures developed so far has proved to be absolutely superior to the others, 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 involve any optimization process. Thus, the resulting system is far from optimal in terms of verification accuracy. To better handle this problem, in this study, we formulate Ψ() as a combination of the likelihoods computed for all the background models. The combination is then optimized using kernel discriminant analysis such that the samples of the null hypothesis can be optimally separated from those of the alternative hypothesis.

The remainder of this paper is organized as follows. Section 2 presents the problem formulation of our approach for speaker verification. Section 3 introduces kernel discriminant analysis used in this work. Section 4 presents our experiment results. Finally, in Section 5, we present our conclusions.

2. Problem formulation

Our objective is to design a function Ψ() that best combines *N* background models according to their individual significance to the classifier. The combination is assumed to be of the form:

$$p(U \mid \bar{\lambda}) = \Psi(p(U \mid \lambda_1), ..., p(U \mid \lambda_N)) = \sum_{i=1}^{N} w_i \, p(U \mid \lambda_i), \qquad (6)$$

where w_i is a weight for $p(U \mid \lambda_i)$, i = 1,…, N, and $w_1 + w_2 + \ldots + w_N = 1$. Suppose all the N background models are Gaussian Mixture Models (GMMs) [1], then Eq. (6) can be viewed as a mixture Gaussian density function. From this perspective, the anti-model $\bar{\lambda}$ is considered as a GMM with two layers of mixture weights, in which one layer stands for each background model and the other for the combination of background models. We refer to $\bar{\lambda}$ as *2-layer GMM* (*GMM2*), since it involves both inner and outer mixture weights. GMM2 is different from the world model Ω in that it is designed to characterize the relationship between individual background models through the use of outer mixture weights, rather than simply pool all the available data and train a single background model. Note that these inner and outer mixture weights are trained by different algorithms. Specifically, the inner mixture weights are estimated using the standard expectation-maximization (EM) algorithm [5], while the outer mixture weights are trained on the basis of the LR.

By applying Eq. (6) to Eq. (2) and reversing the LR, we obtain

$$\begin{aligned}\frac{1}{L(U)} &= \frac{\sum_{i=1}^{N} w_i p(U \mid \lambda_i)}{p(U \mid \lambda)} \\ &= w_1 \frac{p(U \mid \lambda_1)}{p(U \mid \lambda)} + \ldots + w_N \frac{p(U \mid \lambda_N)}{p(U \mid \lambda)} \\ &= \mathbf{w}^T \mathbf{x} \begin{cases} \le \theta' & \text{accept} \\ > \theta' & \text{reject}, \end{cases}\end{aligned} \tag{7}$$

where $\mathbf{w} = [w_1, w_2 \ldots, w_N]^T$ is an N×1 vector of outer mixture weights, the new threshold $\theta' = 1/\theta$, and $\mathbf{x}$ is an $N \times 1$ vector in the space R^N, which is expressed by

$$\mathbf{x} = [\frac{p(U \mid \lambda_1)}{p(U \mid \lambda)} \; \frac{p(U \mid \lambda_2)}{p(U \mid \lambda)} \ldots \frac{p(U \mid \lambda_N)}{p(U \mid \lambda)}]^T. \tag{8}$$

In this way, each speech utterance U is represented by a characteristic vector $\mathbf{x}$, which is analogous to the *anchor model* technique [9] if the background models are regarded as the anchor models. In the sequel, we further rewrite Eq. (7) by

$$\frac{1}{L(U)} = \mathbf{w}^T \mathbf{x} + b = f(\mathbf{x}), \tag{9}$$

where $f(\mathbf{x})$ forms a so-called linear discriminant classifier, in which the bias b plays the same role as the decision threshold θ' in Eq. (7). This classifier translates the goal of solving an LR test problem into the optimization of $\mathbf{w}$ and b, such that the utterances of the 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.

3. Kernel discriminant analysis

Intuitively, $f(\mathbf{x})$ in Eq. (9) can be solved via linear discriminant training algorithms [10]. However, such a method is based on the assumption that the observed data of different classes is linearly separable, which is obviously not adequate 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 the 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($\mathbf{x}$) denote a vector obtained by mapping $\mathbf{x}$ from R^N to F. The objective based on Eq. (9) can be re-defined as

$$f(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x}) + b, \tag{10}$$

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($\mathbf{x}$) can be infeasible. To overcome this difficulty, a promising approach is to characterize the relationship between the data samples in F, instead of computing Φ($\mathbf{x}$) directly. This is achieved by introducing a kernel function $k(\mathbf{x}, \mathbf{y}) = <\Phi(\mathbf{x}), \Phi(\mathbf{y})>$, which is the dot product of two vectors Φ($\mathbf{x}$) and Φ($\mathbf{y}$) in F. The kernel function $k(\cdot)$ must be symmetric positive definite and conform to Mercer's condition [7]. A number of kernel functions exist, such as the simple dot product kernel function, i.e., $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$, and the popular Exponential Radial Basis Function (ERBF) kernel, i.e., $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\| / 2\sigma^2)$, where σ is a tunable parameter. Existing techniques, such as KFD [6] or SVM [7], can be applied to implement Eq. (10).

3.1. Kernel Fisher Discriminant (KFD)

The purpose of KFD is to find a direction $\mathbf{w}$ in the feature space F that maximizes the between-class scatter, while minimizing the within-class scatter. Since the solution of $\mathbf{w}$ must lie in the span of all training data samples mapped in F [6], it can be expressed as

$$\mathbf{w} = \sum_{j=1}^{l} \alpha_j \Phi(\mathbf{x}_j), \tag{11}$$

where l is the number of training data samples. Let $\boldsymbol{\alpha}^T = [\alpha_1, \alpha_2, \ldots, \alpha_l]$. Our goal therefore changes from finding $\mathbf{w}$ to finding $\boldsymbol{\alpha}$. Accordingly, Eq. (10) is equivalent to

$$f(\mathbf{x}) = \sum_{j=1}^{l} \alpha_j k(\mathbf{x}_j, \mathbf{x}) + b, \tag{12}$$

where $\boldsymbol{\alpha}$ and b can be solved by an approach similar to Fisher's Linear Discriminant (FLD) [6].

3.2. Support Vector Machine (SVM)

Alternatively, Eq. (10) can be solved with 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

$$\mathbf{w} = \sum_{j=1}^{l} y_j \alpha_j \Phi(\mathbf{x}_j), \tag{13}$$

which yields

$$f(\mathbf{x}) = \sum_{j=1}^{l} y_j \alpha_j k(\mathbf{x}_j, \mathbf{x}) + b, \tag{14}$$

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. The coefficients α_j and b can be solved using the quadratic programming techniques in [11]. Note that α_j is non-zero for a few support vectors, and zero otherwise. An SVM with a dot product kernel function is known as a Linear SVM.

The inner and outer mixture weights of GMM2 are estimated via the EM algorithm and the kernel discriminant analysis, respectively. That is to say, the GMM2 integrates the Bayesian learning and discriminative training algorithms. The objective is to optimize the classifier by considering the null hypothesis and the alternative hypothesis jointly.

4. Experiments

4.1. Experimental setup

The speaker verification experiments were conducted on speech data extracted from the XM2VTSDB multi-modal database [12]. In accordance with "Configuration II" described in [12], the database was divided into three subsets: "Training", "Evaluation", and "Test". In our experiments, we used "Training" to build the individual client's model and anti-model, and "Evaluation" to optimize **w** and b. 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 4 recording sessions at approximately one-month intervals, and each recording session consisted of 2 shots. In a shot, every speaker was prompted to utter 3 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". 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, by a 32-ms Hamming-windowed frame with 10-ms shifts.

Table 1. *Configuration of the 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 GMM 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; B speakers were chosen from these 198 clients as the cohort. Then, we used 6 utterances/client from session 3, along with 24 (4×2×3) utterances/evaluation-impostor, which yielded 1,194 (6×199) client examples and 119,400 (24×25×199) impostor examples, to optimize **w** and b. However, recognizing the fact that a kernel-based classifier can be intractable when a huge amount of training examples involves, we downsized the number of impostor examples 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.

4.2. Background model selection

We used B+1 background models, consisting of one world model and B cohort set models, to form the characteristic vector **x** in Eq. (8), and B cohort set models to form $L_1(U)$ in Eq. (4). Two cohort selection methods [1] were applied in this experiment. One selected the B closest speakers for each client, and the other selected the B/2 closest speakers plus the B/2 farthest speakers for each client. The selection was based on the speaker distance measure [1], computed by

$$d(\lambda_i, \lambda_j) = \log \frac{p(X_i \mid \lambda_i)}{p(X_i \mid \lambda_j)} + \log \frac{p(X_j \mid \lambda_j)}{p(X_j \mid \lambda_i)}, \tag{15}$$

where λ_i and λ_j were 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 yielded the following two (B+1)x1 characteristic vectors:

$$\mathbf{x} = [\tilde{p}_0(U)\ \tilde{p}_1^c(U)\ ...\ \tilde{p}_B^c(U)]^T, \tag{16}$$

and

$$\mathbf{x} = [\tilde{p}_0(U)\ \tilde{p}_1^c(U)\ ...\ \tilde{p}_{B/2}^c(U)\ \tilde{p}_1^f(U)\ ...\ \tilde{p}_{B/2}^f(U)]^T, \tag{17}$$

where $\tilde{p}_0(U) = p(U|\Omega)/p(U|\lambda)$, $\tilde{p}_i^c(U) = p(U \mid \lambda_{\text{closest } i})/p(U \mid \lambda)$, $\tilde{p}_i^f(U) = p(U \mid \lambda_{\text{farthest } i})/p(U \mid \lambda)$, $i = 1,\ldots, B$ for Eq. (16), and $i = 1,\ldots, B/2$ for Eq. (17). $\lambda_{\text{closest } i}$ and $\lambda_{\text{farthest } i}$ are the i-th closest model and the i-th farthest model of the client model λ, respectively. In the experiments, B was set to 20.

4.3. Experimental results

We implemented the proposed LR system via KFD with Eq. (16) (curve "KFD_w_20c"), KFD with Eq. (17) (curve "KFD_w_10c_10f"), SVM with Eq. (16) (curve "SVM_w_20c"), and SVM with Eq. (17) (curve "SVM_w_10c_10f"), respectively. Both SVM and KFD used an ERBF kernel function with $\sigma = 5$. For performance comparison, three systems, $L_1(U)$ with 20 closest cohort models (curve "L1_20c"), $L_1(U)$ with 10 closest cohort models plus 10 farthest cohort models (curve "L1_10c_10f"), and $L_2(U)$ were used as our baselines.

Fig. 1 shows the results of speaker verification conducted on "Evaluation" with DET curves [13], obtained equivalently by adjusting the decision threshold, i.e., θ or b. Though this experiment was an inside test for our proposed LR system, it is clear that KFD performs better than SVM. To verify that the proposed LR systems are superior to the baseline systems, experiments were conducted on "Test". The results, as shown in Fig. 2, confirm that both the proposed LR systems, SVM and KFD, outperform the baseline systems. We can also see from Fig. 2 that the performances of SVM and KFD are similar, but this is not the case in Fig. 1. We speculate that the KFD classifier might have over-learned the training examples. In addition, it can be seen that there is no significant difference in

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

performance between the background model sets used in Eq. (16) and Eq. (17).

Further analysis of the results via the equal error rate (EER) showed that, for "Test", a 14.87% relative improvement was achieved by SVM_w_20c (EER = 4.35%), compared to 5.11% of $L_2(U)$, which was the best result of the baseline systems.

5. Conclusions

This study has investigated the feasibility of improving the characterization of the alternative hypothesis by combining multiple background models more effectively and robustly. The combination has been formulated as a structure of two-layer Gaussian mixture modeling called GMM2. The GMM2 is first trained via the Bayesian learning algorithm, and further optimized by using kernel discriminant analysis. In this way, the classifier is optimized by considering the null hypothesis and the alternative hypothesis jointly, so that, based on the GMM2-based LR measure, the samples of the null hypothesis can be optimally separated from those of the alternative hypothesis. Experiments conducted on a speaker-verification task showed that the GMM2-based LR measure improves performance significantly. The proposed method can be applied to other types of data and hypothesis testing problems.

6. Acknowledgements

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

7. References

[1] Reynolds, D. A., "Speaker Identification and Verification using Gaussian Mixture Speaker Models", Speech Communication, vol.17, pp. 91-108, 1995.

[2] Reynolds, D. A., Quatieri, T. F., and Dunn, R. B., "Speaker Verification using Adapted Gaussian Mixture Models", Digital Signal Processing, vol. 10, pp. 19-41, 2000.

[3] Higgins, A., Bahler, L., and Porter, J., "Speaker Verification using Randomized Phrase Prompting", Digital Signal Processing, vol. 1, no. 2, pp. 89-106, 1991.

[4] Liu, C. S., Wang, H. C., and Lee, C. H., "Speaker Verification using Normalized Log-Likelihood Score", IEEE Trans. Speech and Audio Processing, vol. 4, pp. 56-60, 1996.

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

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

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

[8] Rosenberg, A. E., Delong, J., Lee, C. H., Juang, B. H., and Soong, F. K., "The use of Cohort Normalized Scores for Speaker Verification", Proc. ICSLP1992.

[9] Sturim, D. E., Reynolds, D. A., Singer, E., and Cambell, J. P., "Speaker Indexing in Large Audio Databases using Anchor Models", Proc. ICASSP2001.

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

[11] Vapnik, V., Statistical Learning Theory, John Wiley & Sons, New York, 1998.

[12] Luettin, J., and Maître, G., Evaluation Protocol for the Extended M2VTS Database (XM2VTSDB), IDIAP-COM 98-05, IDIAP, 1998.

[13] Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M., "The DET Curve in Assessment of Detection Task Performance", Proc. Eurospeech1997.

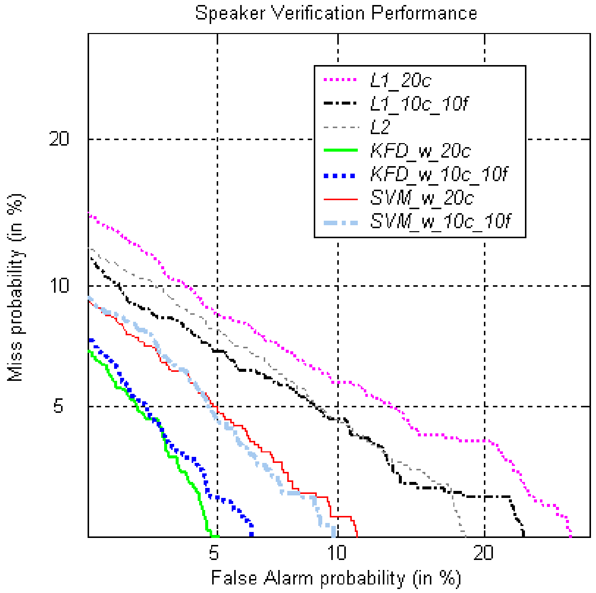

Figure 1 *DET curves for "Evaluation".*

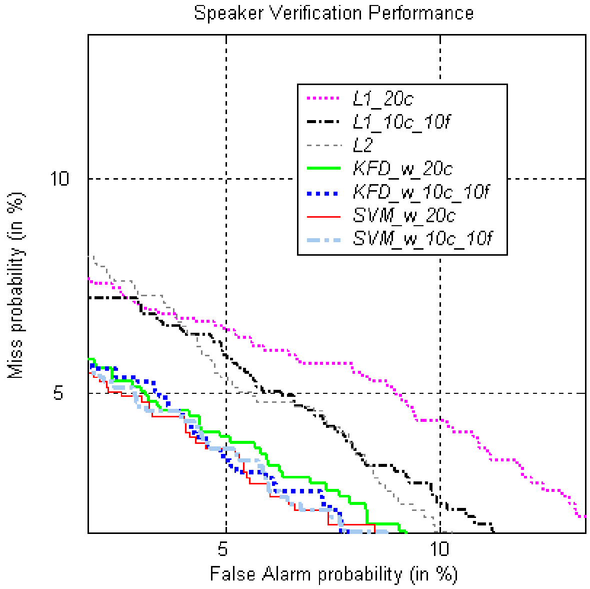

Figure 2 *DET curves for "Test".*