Speaker Verification Using Support Vector Machine with LLR-based Sequence Kernels

Yi-Hsiang CHAO
Department of Applied Geomatics
Ching Yun University
Taoyuan
yschao@cyu.edu.tw

Wei-Ho TSAI
Department of Electronic Engineering
National Taipei University of Technology
Taipei
whtsai@ntut.edu.tw

Hsin-Min WANG
Institute of Information Science
Academia Sinica
Taipei
whm@iis.sinica.edu.tw

Abstract—**Support vector machine (SVM) has been shown powerful in binary classification problems. In order to accommodate SVM to speaker verification problem, the concept of sequence kernel has been developed, which maps variable-length speech data into fixed-dimension vectors. However, constructing a suitable sequence kernel for speaker verification is still an issue. In this paper, we propose a new sequence kernel, named the log-likelihood ratio (LLR)-based sequence kernel, to incorporate LLR-based speaker verification approaches into SVM without needing to represent variable-length speech data as fixed-dimension vectors in advance. Our experimental results show that the proposed sequence kernels outperform the conventional kernel-based approaches.**

Keywords-log-likelihood ratio; speaker verification; sequence kernels; support vector machine

I. Introduction

In essence, speaker verification is a hypothesis testing problem that can be solved by using a log-likelihood ratio (LLR) test [1]. Given an input utterance U, the goal is to determine whether or not U was spoken by the target speaker. Let us consider the following two hypotheses:

H_0: U was spoken by the target speaker,
H_1: U was not spoken by the target speaker.

The LLR test can be expressed as

$$L(U) = \log p(U \mid \lambda) - \log p(U \mid \bar{\lambda}) \begin{cases} \geq \theta & \text{accept } H_0 \\ < \theta & \text{accept } H_1 , \end{cases} \tag{1}$$

where θ is a decision threshold; λ and $\bar{\lambda}$ are respectively the target speaker model and anti-mode that are usually formulated as Gaussian mixture models (GMMs) [1]. One popular approach is the GMM-UBM system [2] which is expressed as

$$L_{\text{UBM}}(U) = \log p(U \mid \lambda) - \log p(U \mid \Omega) . \tag{2}$$

This approach pools all speech data from a large number of background speakers, and trains a universal background model (UBM) Ω [2] for $\bar{\lambda}$. Instead of using a UBM, a well-known score normalization method, called T-norm [3], is to train a set of background models $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ using speech from several representative speakers, called a cohort [4], which simulates potential impostors. Then, the mean μ_U and the standard deviation σ_U of N log-likelihoods, $\log p(U \mid \lambda_i)$, $i = 1, 2, \ldots, N$, are calculated. The decision function of T-norm is formed by:

$$L_{\text{Tnorm}}(U) = \frac{\log p(U \mid \lambda) - \mu_U}{\sigma_U} . \tag{3}$$

T-norm and GMM-UBM are the two predominant LLR-based approaches for speaker verification in the last decade. More recently, SVM based on sequence kernels [5-9] have been found to outperform traditional LLR-based approaches. Such SVM methods map variable-length speech data into fixed-dimension vectors, and the mapping overcomes the limitation that the conventional SVM is awkward to handle the dynamic patterns like speech. However, constructing a proper sequence kernel tied in with speaker verification is still an issue. Bengio [7] proposed a sequence kernel-based decision function as follows:

$$L_{\text{Bengio}}(U) = a_1 \log p(U \mid \lambda) - a_2 \log p(U \mid \Omega) + b, \tag{4}$$

where a_1, a_2, and b are adjustable parameters estimated using SVM. The input to SVM is a two-dimension vector

$[\log p(U\,|\,\lambda)\ -\log p(U\,|\,\Omega)]^T$. An extended version of Eq. (4) using the Fisher kernel and the LR score-space kernel for SVM was investigated in [8]. The supervector kernel [6,9] is another kind of sequence kernel that is formed by concatenating the parameters of a GMM or maximum likelihood linear regression (MLLR) matrices. All the above-mentioned methods have to convert a variable-length utterance into a fixed-dimension vector before a kernel function is computed. Since the fixed-dimension vector is formed independent of the kernel computation, this process is not optimal in terms of overall design. In this paper, we propose a new kernel, named the LLR-based sequence kernel, which attempts to compute the kernel function without representing utterances into fixed-dimension vectors in advance. Our goal is to integrate SVM-based methods with LLR-based speaker verification approaches by embedding an LLR in the sequence kernel.

The remainder of the paper is organized as follows. Section II introduces the support vector machine. Section III describes the proposed LLR-based sequence kernel. Section IV, contains the experimental results. Finally, in Section V, we present our conclusions.

II. Support Vector Machine

Kernel techniques [10] have been widely applied to pattern recognition and classification problems. Among the techniques, support vector machine (SVM) is the most prevalent method. The goal of SVM is to find a separating hyperplane that maximizes the margin between classes. Following [11], the decision function of SVM is expressed as

$$L_{\mathrm{SVM}}(U) = \sum_{j=1}^{J} y_j \alpha_j k(U_j, U) + b\ , \tag{5}$$

where each training samples U_j, j = 1, 2,…, J, is labeled by either $y_j = 1$ (the hypothesis H_0) or $y_j = -1$ (the hypothesis H_1); $k(U_j, U)$ is the kernel function. The coefficients α_j and b can be solved by using the quadratic programming techniques [12]. Note that most α_j are equal to zero, and training samples associated with non-zero α_j are called support vectors. A few support vectors play a key role in deciding the optimal margin between classes in SVM.

III. LLR-based Sequence Kernels

A. Mercer Kernels

The effectiveness of SVM depends crucially on how the kernel function $k(\cdot)$ is designed. A kernel function represents a certain measurement of similarities between samples in a mapped feature space. It must be symmetric, positive definite, and conform to Mercer's condition [10]. There are a number of kernel functions [10] used in different applications. One special form of kernel, named sequence kernel, is represented by

$$k(U_1, U_2) = \Phi(U_1)^T \Phi(U_2)\ , \tag{6}$$

where U_1 and U_2 are two sequences, and $\Phi(\cdot)$ is a nonlinear function that implicitly maps each variable-length sequence into a fixed-dimension vector in feature space F. Such a kernel function is expressed by the inner product of two vectors $\Phi(U_1)$ and $\Phi(U_2)$. However, $\Phi(\cdot)$ might be incomputable in most kernels since the dimension of F could be infinite. Under this circumstance, the choice of $\Phi(\cdot)$ becomes tricky and far from optimal in terms of the overall design for the kernel function.

In this work, we try to represent a sequence kernel as the inner product of two computable quantities instead of function $\Phi(\cdot)$ that is usually incomputable. We define a new form of sequence kernel in accordance with the property of Mercer kernels [10] as

$$k(U_1, U_2) = f(U_1) \cdot f(U_2)\ , \tag{7}$$

where $f(\cdot)$ is any function that satisfies $f : U \to \Re$. In this case, each variable-length sequence is converted into a real number rather than an implicit higher-dimension vector. It is our hope to design a function $f(\cdot)$ that can reflect the speaker voice characteristics underlying speech utterance U, rather than some illusive higher-dimension vectors. Toward this end, we define function f in connection with LLR as

$$f(U_i) = \log \frac{p(U_i\,|\,\lambda_{s_i})}{p(U_i\,|\,\bar{\lambda}_{s_i})}\ , \tag{8}$$

where i = 1 or 2; λ_{s_i} and $\bar{\lambda}_{s_i}$ are, respectively, the target model and anti-model for a target speaker s_i that is claimed to produce utterance U_i. By substituting Eq. (8) into Eq. (7), we obtain

$$\begin{aligned} k(U_1, U_2) &= f(U_1) \cdot f(U_2) \\ &= \log \frac{p(U_1\,|\,\lambda_{s_1})}{p(U_1\,|\,\bar{\lambda}_{s_1})} \cdot \log \frac{p(U_2\,|\,\lambda_{s_2})}{p(U_2\,|\,\bar{\lambda}_{s_2})}. \end{aligned} \tag{9}$$

Ideally, Eq. (8) is positive when utterance U_i is produced by the target speaker s_i, and is negative otherwise. Thus, it can be observed that Eq. (9) is positive if both U_1 and U_2 belong to the same hypothesis, and is negative otherwise. Table I summarizes the relation between the ground truth of two utterances and the sign of the resulting kernel function in Eq. (9). The tabulated relations show that Eq. (9) satisfies the property of a similarity measurement.

TABLE I. The Relation Between The Ground Truth Of Two Utterances And The Resulting LLR-Based Sequence Kernel, Where "+" And "-" Denote The Positive And Negative Value, Respectively.

U_1	U_2	$f(U_1)$	$f(U_2)$	$k(U_1, U_2)$
accept H_0	accept H_0	+	+	+
accept H_0	accept H_1	+	-	-
accept H_1	accept H_0	-	+	-
accept H_1	accept H_1	-	-	+

B. Composite Kernels

It should be noted that Eq. (8) is only an example of the proposed concept of LLR-based kernel. There are a number of other different LLR-based kernel functions that we can define. For example, the function f can be $L_{\text{UBM}}(U)$ in Eq. (2) or $L_{\text{Tnorm}}(U)$ in Eq. (3). Recognizing the fact that $L_{\text{UBM}}(U)$ and $L_{\text{Tnorm}}(U)$ are current state-of-the-art approaches for speaker verification, we further propose to integrate them into the kernel function. Complying with the closure property of Mercer kernels [10], we define a composite kernel function as

$$k_{\text{com}}(U_1,U_2) = k_{\text{UBM}}(U_1,U_2) + k_{\text{Tnorm}}(U_1,U_2), \tag{10}$$

where

$$k_{\text{UBM}}(U_1,U_2) = L_{\text{UBM}}(U_1) \cdot L_{\text{UBM}}(U_2)\,, \tag{11}$$

and

$$k_{\text{Tnorm}}(U_1,U_2) = L_{\text{Tnorm}}(U_1) \cdot L_{\text{Tnorm}}(U_2)\,. \tag{12}$$

IV. Experiments

A. Experimental setup

Our speaker verification experiments were conducted on the speech data extracted from the extended M2VTS database (XM2VTSDB) [13]. In accordance with "Configuration II" described in Table II [14], the database was divided into three subsets: "Training", "Evaluation" [1], and "Test". In our experiments, we used "Training" to build each target speaker's model and anti-model, and "Evaluation" to estimate the decision threshold θ in Eq. (1), and the coefficients α_j and b of SVM in Eq. (5). The performance of speaker verification was then evaluated on the "Test" subset.

TABLE II. 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			

As shown in Table II, a total of 293 speakers [2] in the database were divided into 199 clients (target speakers), 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-frequency cepstral coefficients (MFCCs) [15] and their first time derivatives, by a 32-ms Hamming-windowed frame with 10-ms shifts.

We used 12 (2×2×3) utterances/client from sessions 1 and 2 to train the client 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 UBM, represented by a GMM with 256 mixture components; 50 closest speakers were chosen from these 198 clients as a cohort. Here, the degree of closeness is measured in terms of the pairwise distance defined in [1]:

$$d(\lambda_i,\lambda_j) = \log\frac{p(U_i \mid \lambda_i)}{p(U_i \mid \lambda_j)} + \log\frac{p(U_j \mid \lambda_j)}{p(U_j \mid \lambda_i)}, \tag{13}$$

where λ_i and λ_j were speaker models trained using the i-th speaker's utterances U_i and the j-th speaker's utterances U_j, respectively. 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 estimate α_j, b and θ. However, recognizing the fact that the kernel method 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.

B. Experimental results

We implemented three SVM systems using the individual LLR-based sequence kernel functions in the following:

1) $k_{\text{UBM}}(U_1, U_2)$ in Eq. (11) ("UBMKernel"),

2) $k_{\text{Tnorm}}(U_1, U_2)$ in Eq. (12) with 50 closest cohort models ("TnormKernel"), and

3) $k_{\text{com}}(U_1, U_2)$ in Eq. (10) ("Compositekernel").

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

1) $L_{\text{UBM}}(U)$ in Eq. (2) ("GMM-UBM"),

2) $L_{\text{Tnorm}}(U)$ in Eq. (3) with 50 closest cohort models ("Tnorm_50c"), and

3) $L_{\text{Bengio}}(U)$ in Eq. (4) using an RBF kernel function with $\sigma = 10$ ("GMM-UBM/SVM").

Fig. 1 shows the results of speaker verification evaluated on the "Test" subset in terms of DET curves [16]. From Fig. 1, we observe that "Compositekernel" obviously outperforms all the baseline systems. It can also been seen that the curve "UBMKernel" overlaps the curve "GMM-UBM" while the curve "TnormKernel" overlaps the curve "Tnorm_50c". We speculate that the SVM system with the kernel function represented by a single LLR, $L_{\text{UBM}}(U)$ or $L_{\text{Tnorm}}(U)$, will

[1] This is usually called the "Development" set in the speech recognition community. We use "Evaluation" in accordance with the configuration of the XM2VTS database.

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

degenerate to the prime LLR system, $L_{\mathrm{UBM}}(U)$ or $L_{\mathrm{Tnorm}}(U)$, respectively. Table III summarizes the experimental results based on the minimum detection cost function (DCF) [17], defined as

$$C_{DET} = C_{Miss} \times P_{Miss} \times P_{Target} + C_{Fa} \times P_{Fa} \times (1 - P_{Target}) , \quad (14)$$

where P_{Miss} and P_{Fa} are the miss probability and the false-alarm probability, respectively; C_{Miss} and C_{Fa} are the respective relative costs of the detection errors; and P_{Target} is the *a priori* probability of the target speaker. In our experiments, C_{Miss} and C_{Fa} were both set to 1, and P_{Target} = 0.5. This special case of DCF is known as the half total error rate (HTER) [18]. From Table III, it is clear that "Compositekernel" achieves the best performance with a 3.85% relative improvement in terms of the minimum DCF for the "Test" subset, compared to "GMM-UBM/SVM", which was the best result of the baseline systems.

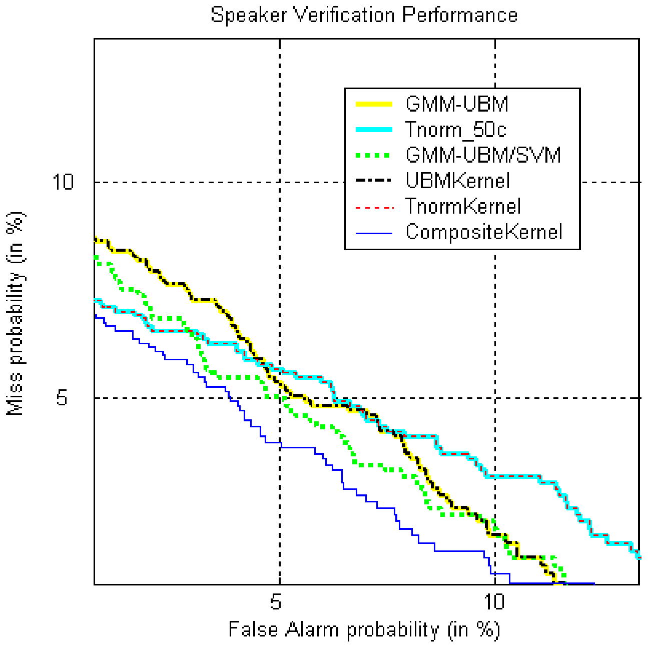

Figure 1. DET curves for "Test".

TABLE III. MINIMUN DCFS FOR "TEST".

Methods	minDCF
GMM-UBM	0.0508
Tnorm_50c	0.0465
GMM-UBM/SVM	0.0467
UBMKernel	0.0508
TnormKernel	0.0465
Compositekernel	0.0449

V. CONCLUSION

In this paper, we have presented a new kernel, named the log-likelihood ratio (LLR)-based sequence kernel, for SVM-based speaker verification. The proposed sequence kernel can be computed without needing to represent the variable-length speech into a fixed-dimensional vector in the sequence kernel in advance. This allows the state-of-the-art LLR-based speaker verification approaches to be integrated into the SVM framework under a joint design strategy. Our experimental results have shown that the proposed sequence kernel methods outperform the conventional kernel-based approaches.

ACKNOWLEDGMENT

This work was funded by the National Science Council, Taiwan, under Grant: NSC99-2218-E-231-003.

REFERENCES

[1] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," Speech Communication, vol.17, pp. 91-108, 1995.

[2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," Digital Signal Processing, vol. 10, pp. 19-41, 2000.

[3] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification system," Digital Signal Processing, vol. 10, no. 1, pp. 42-54, 2000.

[4] A. E. Rosenberg, J. Delong, C. H. Lee, B. H. Juang, and F. K. Soong, "The use of cohort normalized scores for speaker verification", in Proc. ICSLP, Canada, pp. 599-602, 1992.

[5] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition", Computer Speech and Language, vol. 20, no. 2-3, pp. 210-229, 2006.

[6] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machine using GMM supervectors for speaker verification", IEEE Signal Processing Letters, vol. 13, no. 5, pp. 308-311, 2006.

[7] S. Bengio and J. Mariéthoz, "Learning the decision function for speaker verification", in Proc. ICASSP, USA, pp. 425-428, 2001.

[8] V. Wan and S. Renals, "Speaker verification using sequence discriminant support vector machines", IEEE Trans. Speech and Audio Processing, vol. 13, no. 2, pp. 203-210, 2005.

[9] Zahi N. Karam and W. M. Campbell, "A multi-class MLLR kernel for SVM speaker recognition", in Proc. ICASSP, 2008.

[10] Ralf Herbrich, Learning Kernel Classifiers: Theory and Algorithms, MIT Press, Cambridge, 2002.

[11] C. Burges, "A tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery, vol.2, pp. 121-167, 1998.

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

[13] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, "XM2VTSDB: the extended M2VTS database," in Proc. AVBPA, 1999.

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

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

[16] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance", in Proc. Eurospeech1997.

[17] http://www.nist.gov/speech/tests/spk/index.htm

[18] J. Lindberg, J. Koolwaaij, H. P. Hutter, D. Genoud, J. B. Pierrot, M. Blomberg, and F. Bimbot, "Techniques for a priori decision threshold estimation in speaker verification," in Proc. RLA2C, Avignon, pp. 89-92, 1998