Speaker Verification System Using LLR-Based Multiple Kernel Learning

Yi-Hsiang Chao

Abstract Support Vector Machine (SVM) has been shown powerful in pattern recognition problems. SVM-based speaker verification has also been developed to use the concept of sequence kernel that is able to deal with variable-length patterns such as speech. In this paper, we propose a new kernel function, named the Log-Likelihood Ratio (LLR)-based composite sequence kernel. This kernel not only can be jointly optimized with the SVM training via the Multiple Kernel Learning (MKL) algorithm, but also can calculate the speech utterances in the kernel function intuitively by embedding an LLR in the sequence kernel. Our experimental results show that the proposed method outperforms the conventional speaker verification approaches.

Keywords Log-Likelihood ratio · Speaker verification · Support vector machine · Multiple kernel learning · Sequence kernel

1 Introduction

The task of speaker verification problem is to determine whether or not an input speech utterance U was spoken by the target speaker. In essence, speaker verification is a hypothesis test problem that is generally formulated as a Log-Likelihood Ratio (LLR) [1] measure. Various LLR measures have been designed [1–4]. One popular LLR approach is the GMM-UBM system [1], which is expressed as

$$L_{\mathrm{UBM}}(U) = \log p(U|\lambda) - \log p(U|\Omega), \tag{1}$$

Y.-H. Chao (✉)
Department of Applied Geomatics, Chien Hsin University of Science and Technology, Taoyuan, Taiwan
e-mail: yschao@uch.edu.tw

J. J. (Jong Hyuk) Park et al. (eds.), *Multimedia and Ubiquitous Engineering*, Lecture Notes in Electrical Engineering 240, DOI: 10.1007/978-94-007-6738-6_18,

where λ is a target speaker Gaussian Mixture Model (GMM) [1] trained using speech from the claimed speaker, and Ω is a Universal Background Model (UBM) [1] trained using all the speech data from a large number of background speakers. Instead of using a single model UBM, an alternative approach is to train a set of background models $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ using speech from several representative speakers, called a cohort [2], which simulates potential impostors. This leads to several LLR measures [3], such as

$$L_{\text{Max}}(U) = \log p(U|\lambda) - \max_{1 \le i \le N} \log p(U|\lambda_i), \tag{2}$$

$$L_{\text{Ari}}(U) = \log p(U|\lambda) - \log\left(\sum\nolimits_{i=1}^{N} p(U|\lambda_i)/N\right), \tag{3}$$

$$L_{\text{Geo}}(U) = \log p(U|\lambda) - \left(\sum\nolimits_{i=1}^{N} \log p(U|\lambda_i)\right)/N, \tag{4}$$

and a well-known score normalization method called T-norm [4]:

$$L_{\text{Tnorm}}(U) = L_{\text{Geo}}(U)/\sigma_U, \tag{5}$$

where σ_U is the standard deviation of N scores, $\log p(U|\lambda_i), i = 1, 2, \ldots, N$.

In recent years, Support Vector Machine (SVM)-based speaker verification methods [5–8] have been proposed and successfully found to outperform traditional LLR-based approaches. Such SVM methods use the concept of sequence kernels [5–8] that can deal with variable-length input patterns such as speech. Bengio [5] proposed an SVM-based decision function:

$$L_{\text{Bengio}}(U) = a_1 \log p(U|\lambda) - a_2 \log p(U|\Omega) + b, \tag{6}$$

where a_1, a_2, and b are adjustable parameters estimated using SVM. An extended version of Eq. (6) using the Fisher kernel and the LR score-space kernel for SVM was investigated in [6]. The supervector kernel [7] is another kind of sequence kernel for SVM that is formed by concatenating the parameters of a GMM or Maximum Likelihood Linear Regression (MLLR) [8] matrices. Chao [3] proposed using SVM to directly fuse multiple LLR measures into a unified classifier with an LLR-based input vector. All the above-mentioned methods have the same point that must 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 function, named the LLR-based composite sequence kernel, which attempts to compute the kernel function without needing to represent utterances into fixed-dimension vectors in advance. This kernel not only can be jointly optimized with the SVM training via the Multiple Kernel Learning (MKL) [9] algorithm, but also can calculate the speech utterances in the kernel function intuitively by embedding an LLR in the sequence kernel.

2 Kernel-Based Discriminant Framework

In essence, there is no theoretical evidence to indicate what sort of LLR measures defined in Eqs. (1)–(5) is absolutely superior to the others. An intuitive way [3] to improve the conventional LLR-based speaker verification methods would be to fuse multiple LLR measures into a unified framework by virtue of the complementary information that each LLR can contribute. Given M different LLR measures, $L_m(U)$, $m = 1, 2,\ldots, M$, a fusion-based LLR measure [3] can be defined as

$$L_{\text{Fusion}}(U) = \mathbf{w}^T \Phi(U) + b, \tag{7}$$

where b is a bias, $\mathbf{w} = [w_1\, w_2 \ldots w_M]^T$ and $\Phi(U) = [L_1(U)\, L_2(U) \ldots L_M(U)]^T$ are the $M \times 1$ weight vector and LLR-based vector, respectively. The implicit idea of $\Phi(U)$ is that a variable-length input utterance U can be represented by a fixed-dimension characteristic vector via a nonlinear mapping function $\Phi(\cdot)$. Equation (7) forms a nonlinear discriminant classifier, which can be implemented by using the kernel-based discriminant technique, namely the Support Vector Machine (SVM) [10]. The goal of SVM is to find a separating hyperplane that maximizes the margin between classes. Following [10], $\mathbf{w}$ in Eq. (7) can be expressed as $\mathbf{w} = \sum_{j=1}^{J} y_j \alpha_j \Phi(U_j)$, which yields an SVM-based measure:

$$L_{\text{SVM}}(U) = \sum_{j=1}^{J} y_j \alpha_j k(U_j, U) + b, \tag{8}$$

where each training utterance U_j, $j = 1, 2,\ldots, J$, is labeled by either $y_j = 1$ (the positive sample) or $y_j = -1$ (the negative sample), and $k(U_j, U) = \Phi(U_j)^T \Phi(U)$ is the kernel function [10] represented by an inner product of two vectors $\Phi(U_j)$ and $\Phi(U)$. The coefficients α_j and b can be solved by using the quadratic programming techniques [10].

2.1 LLR-Based Multiple Kernel Learning

The effectiveness of SVM depends crucially on how the kernel function $k(\cdot)$ is designed. A kernel function must be symmetric, positive definite, and conform to Mercer's condition [10]. There are a number of kernel functions [10] used in different applications. For example, the sequence kernel [6] can take variable-length speech utterances as inputs. In this paper, we rewrite the kernel function in Eq. (8) as

$$k(U_j, U) = [L_1(U_j) \ldots L_M(U_j)][L_1(U) \ldots L_M(U)]^T = \sum_{m=1}^{M} k_m(U_j, U). \tag{9}$$

Complying with the closure property of Mercer kernels [10], Eq. (9) becomes a composite kernel represented by the sum of M LLR-base sequence kernels [11] defined by

$$k_m(U_j, U) = L_m(U_j) \cdot L_m(U), \tag{10}$$

where $m = 1, 2, \ldots, M$. Since the design of Eq. (9) does not involve any optimization process with respect to the combination of M LLR-base sequence kernels, we further redefine Eq. (9) as a new form, named the LLR-base composite sequence kernel, in accordance with the closure property of Mercer kernels [10]:

$$k_{\text{com}}(U_j, U) = \sum\nolimits_{m=1}^{M} \beta_m k_m(U_j, U), \tag{11}$$

where β_m is the weight of the m-th kernel function $k_m(\cdot)$ subject to $\sum_{m=1}^{M} \beta_m = 1$ and $\beta_m \geq 0, \forall m$. This combination scheme quantifies the unequal nature of M LLR-base sequence kernel functions by a set of weights $\{\beta_1, \beta_2, \ldots, \beta_M\}$. To obtain a reliable set of weights, we apply the MKL [9] algorithm. Since the optimization process is related to the speaker verification accuracy, this new composite kernel defined in Eq. (11) is expected to be more effective and robust than the original composite kernel defined in Eq. (9).

The optimal weights β_m can be jointly trained with the coefficients α_j of the SVM in Eq. (8) via the MKL algorithm [9]. Optimization of the coefficients α_j and the weights β_m can be performed alternately. First we update the coefficients α_j while fixing the weights β_m, and then we update the weights β_m while fixing the coefficients α_j. These two steps can be repeated until convergence. In this work, the MKL algorithm is implemented via the SimpleMKL toolbox developed by Rakotomamonjy et al. [9].

3 Experiments

3.1 Experimental Setup

Our speaker verification experiments were conducted on the speech data extracted from the extended M2VTS database (XM2VTSDB) [12]. In accordance with "Configuration II" described in Table 1 [12], the database was divided into three subsets: "Training", "Evaluation", and "Test". In our experiments, we used "Training" to build each target speaker GMM and background models, and "Evaluation" to estimate the coefficients α_j in Eq. (8) and the weights β_m 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 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-

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			

frequency cepstral coefficients (MFCCs) [13] and their first time derivatives, by a 32-ms Hamming-windowed frame with 10-ms shifts.

We used 12 ($2 \times 2 \times 3$) 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. Then, we used 6 utterances/client from session 3, along with 24 ($4 \times 2 \times 3$) utterances/evaluation-impostor, which yielded 1,194 (6×199) client examples and 119,400 ($24 \times 25 \times 199$) impostor examples, to estimate α_j and β_m. 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 ($24 \times 69 \times 199$) impostor trials.

3.2 Experimental Results

We implemented two SVM systems, $L_{\text{Fusion}}(U)$ in Eq. (7) ("LLRfusion") and $k_{\text{com}}(U_j, U)$ in Eq. (11) ("MKL_LLRfusion"), both of which are fused by five LLR-based sequence kernel functions defined in Eqs. (1)–(5). For the purpose of performance comparison, we used six baseline systems, $L_{\text{UBM}}(U)$ in Eq. (1) ("GMM-UBM"), $L_{\text{Bengio}}(U)$ in Eq. (6) ("GMM-UBM/SVM"), $L_{\text{Max}}(U)$ in Eq. (2) ("Lmax_50C"), $L_{\text{Ari}}(U)$ in Eq. (3) ("Lari_50C"), $L_{\text{Geo}}(U)$ in Eq. (4) ("Lgeo_50C"), and $L_{\text{Tnorm}}(U)$ in Eq. (5) ("Tnorm_50C"), where 50C represents 50 closest cohort models were used. Figure 1 shows the results of speaker verification evaluated on the "Test" subset in terms of DET curves [14]. We can observe that the curve "MKL_LLRfusion" not only outperforms six baseline systems, but also performs better than the curve "LLRfusion". Further analysis of the results via the minimum Half Total Error Rate (HTER) [14] showed that a 5.76 % relative improvement was achieved by "MKL_LLRfusion" (the minimum HTER = 3.93 %), compared to 4.17 % of "LLRfusion".

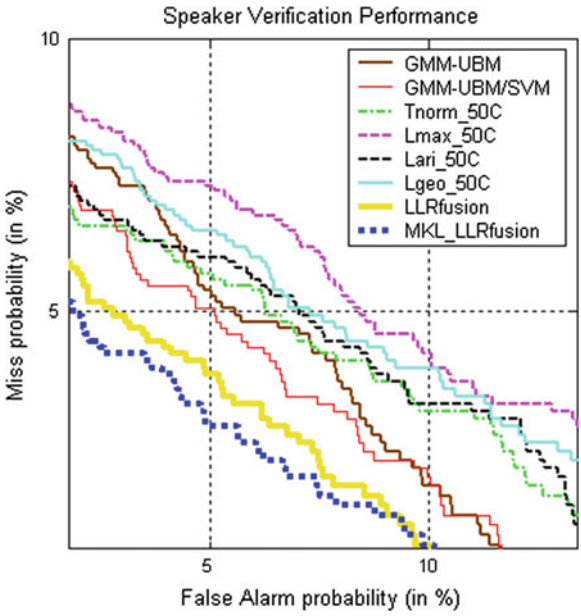

Fig. 1 DET curves for "Test"

4 Conclusion

In this paper, we have presented a new kernel function, named the Log-Likelihood Ratio (LLR)-based composite sequence kernel, for SVM-based speaker verification. This kernel function not only can be jointly optimized with the SVM training via the Multiple Kernel Learning (MKL) algorithm, but also can calculate the speech utterances in the kernel function intuitively by embedding an LLR in the sequence kernel. Our experimental results have shown that the proposed system outperforms the conventional speaker verification approaches.

Acknowledgments This work was funded by the National Science Council, Taiwan, under Grant: NSC101-2221-E-231-026.

References

1. Reynolds DA, Quatieri TF, Dunn RB (2000) Speaker verification using adapted Gaussian mixture models. Digit Signal Proc 10:19–41
2. Rosenberg AE, Delong J, Lee CH, Juang BH, Soong FK (1992) The use of Cohort Normalized scores for speaker verification. Proc, ICSLP
3. Chao YH, Tsai WH, Wang HM, Chang RC (2006) A kernel-based discrimination framework for solving hypothesis testing problems with application to speaker verification. Proceedings of the ICPR
4. Auckenthaler R, Carey M, Lloyd-Thomas H (2000) Score normalization for text-independent speaker verification system. Digit Signal Proc. 10:42–54

5. Bengio S, Mariéthoz J (2001) Learning the decision function for speaker verification. Proceedings of the ICASSP
6. Wan V, Renals S (2005) Speaker verification using sequence discriminant support vector machines. IEEE Trans Speech Audio Proc 13:203–210
7. Campbell WM, Sturim DE, Reynolds DA (2006) Support vector machine using GMM supervectors for speaker verification. IEEE Signal Proc Lett 13
8. Karam ZN, Campbell WM (2008) A multi-class MLLR Kernel for SVM speaker recognition. Proceedings of the ICASSP
9. Rakotomamonjy A, Bach F.R, Canu S, Grandvalet Y (2008) SimpleMKL. J. Mach Learn Res 9:2491–2521
10. Herbrich R (2002) Learning Kernel classifiers: theory and algorithms, MIT Press
11. Chao YH, Tsai WH, Wang HM (2010) Speaker verification using support vector machine with LLR-based sequence Kernels. Proceedings of the ISCSLP
12. Luettin J, Maître G (1998) Evaluation protocol for the extended M2VTS database (XM2VTSDB). IDIAP-COM 98-05, IDIAP
13. Huang X, Acero A, Hon HW (2001) Spoken language processing. Prentics Hall
14. Bengio S, Mariéthoz J (2004) The expected performance curve: a new assessment measure for person authentication. Proceedings ODYSSEY