Local Pairwise Linear Discriminant Analysis for Speaker Verification

Liang He, *Member, IEEE*, Xianhong Chen, Can Xu, Jia Liu, *Member, IEEE*, and Michael T. Johnson, *Senior Member, IEEE*

***Abstract*—Linear discriminant analysis—probabilistic linear discriminant analysis (LDA-PLDA) is a standard and effective backend in the field of speaker verification. The object of LDA is to perform dimensionality reduction while minimizing within-class covariance and maximizing between-class covariance. For a target class (or speaker), our task is to make a binary decision about whether a test utterance is from a specific target speaker. Generally, the nontarget test utterances that are close to the target speaker are easily misjudged. Inspired by this idea, we propose a local pairwise linear discriminant analysis (LPLDA) algorithm. This new method focuses on maximizing the local pairwise covariance, which represents the local structure between the target class samples and neighboring nontarget class samples, instead of the between-class covariance, which represents the global structure of the data. Experiments on the NIST SRE 2010, 2014, and 2016 database show that, the proposed LPLDA-PLDA backend has significant performance improvements over the LDA-PLDA backend.**

***Index Terms*—Linear discriminant analysis (LDA), local pairwise linear discriminant analysis (LPLDA), probabilistic linear discriminant analysis (PLDA), speaker verification.**

I. Introduction

MAINSTREAM speaker verification systems, such as Ivector [1], Dvector [2], and Xvector [3], can be divided into two components: the frontend feature extractor and the backend classifier. The frontend feature extractor realizes the mapping from an utterance to a fixed length vector, while the backend classifier models these vectors for decision making.

Recently, the GMM-Ivector [1], ASR-DNN-Ivector [4], end-to-end DNN [2], [5], [6], and time-delay neural network (TDNN) [3] frontends have been investigated to extract more informative vectors for speaker verification. Although there are a wide variety of frontends, there are comparatively few effective backends. Although the DNN approach has been well studied [7], the most effective and widely used backend is still linear discriminant analysis—probabilistic linear discriminant analysis (LDA-PLDA) [8], [9]. LDA is a classical dimensionality reduction algorithm, which simultaneously minimizes within-class covariance and maximizes between-class covariance [10]. There has been some recent works aimed at improving the performance of LDA for speaker verification. One of these is source normalized LDA [11], [12], which improves the robustness in mismatched evaluation conditions by normalizing scatter matrices across different sources. Another approach is weighted LDA [13]. When calculating the between-class scatter matrix, weighted LDA improves performance by weighting class pairs in inverse proportion to their distance. Based on the idea that the distribution of Ivectors may not be Gaussian, nonparametric discriminant analysis (NDA) [14], [15] calculates both within- and between-class scatter matrices on a local basis using a nearest neighbor rule.

For a specific target speaker, neighboring nontarget utterances (local confusable vectors) are much more important for learning a decision boundary, so emphasizing these instead of treating all examples equally should be beneficial [14]–[16]. Based on this idea, we introduce a local pairwise linear discriminant analysis (LPLDA) algorithm to emphasize local structure instead of global structure on data.

II. Linear Discriminant Analysis

The core elements of LDA are the within-class scatter matrix and the between-class scatter matrix. Let $\mathbf{x}_i^s$ represents the ith sample (vector) of class s, the within-class and between-class scatter matrix are defined as follows [1], [10], [17]:

$$\mathbf{S}_w = \sum_{s=1}^{S} \frac{1}{n_s} \sum_{i=1}^{n_s} (\mathbf{x}_i^s - \boldsymbol{\mu}_s)(\mathbf{x}_i^s - \boldsymbol{\mu}_s)^t$$
$$\mathbf{S}_b = \sum_{s=1}^{S} (\boldsymbol{\mu}_s - \boldsymbol{\mu})(\boldsymbol{\mu}_s - \boldsymbol{\mu})^t \tag{1}$$

where S is the total class number, n_s is the sample number, and $\boldsymbol{\mu}_s$ is the sample mean of the sth class. $\boldsymbol{\mu}$ is the overall sample mean. LDA can be formulated as an optimization problem to find a subspace V that maximizes the ratio of the between-class scattering to the within-class scattering, as

$$V = \underset{V}{\operatorname{argmax}} \frac{\det(V^t \mathbf{S}_b V)}{\det(V^t \mathbf{S}_w V)}. \tag{2}$$

Manuscript received June 10, 2018; revised September 2, 2018; accepted September 3, 2018. Date of publication September 10, 2018; date of current version September 13, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61403224, Grant 61370034, and Grant 61273268. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sebastien Marcel. *(Corresponding author: Liang He.)*

L. He, X. Chen, C. Xu, and J. Liu are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: heliang@mail.tsinghua.edu.cn; chenxianhong@mail.tsinghua.edu.cn; xucan@mail.tsinghua.edu.cn; liuj@mail.tsinghua.edu.cn).

M. T. Johnson is with the Electrical and Computer Engineering, University of Kentucky, Lexington, KY 40506 USA (e-mail: mike.johnson@uky.edu).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2018.2869107

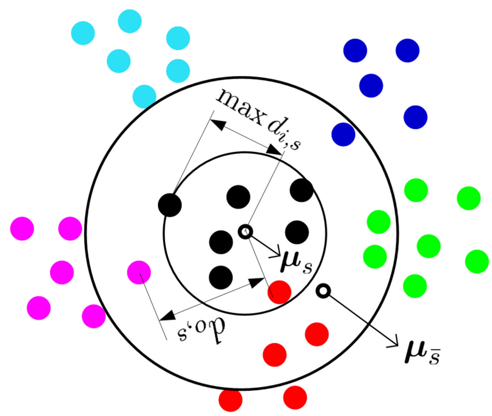

Fig. 1. Method for finding local pairs. Colors represent class identities. For a target class s (black dots), $\boldsymbol{\mu}_s$ is its mean vector. $\boldsymbol{\mu}_{\bar{s}}$ is the mean of its confusable samples $\mathbf{x}_j^{\bar{s}}$ (dots located in the outer circle but not black), $n_{\bar{s}} = 7$. The number of out-of-class samples located in the inner circle is $n_{*,o,s} = 1$. The $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_{\bar{s}}$ form a pair.

In comparison, the between-class scatter matrix in NDA [14], [15] is defined as follows:

$$\mathbf{S}_b = \sum_{s=1}^{S} \sum_{s'=1, s' \neq s}^{S} \sum_{i=1}^{n_s} \omega_i^{s,s'} (\mathbf{x}_i^s - \boldsymbol{\mu}_i^{s,s'})(\mathbf{x}_i^s - \boldsymbol{\mu}_i^{s,s'})^t \quad (3)$$

where $\boldsymbol{\mu}_i^{s,s'}$ is the local mean for $\mathbf{x}_i^s$ from class s', and $\omega_i^{s,s'}$ is the weight function. For more details, please refer to [14] and [15].

III. Local Pairwise Linear Discriminant Analysis

A. Motivation

The testing phase for speaker verification requires determining whether or not the test utterance and the enrollment utterance are from the same target speaker. In this binary task, a neighboring nontarget utterance is more confusable than other nontarget utterances. Because of this, shifting more attention from the overall data structure to the local structure of the target class samples and its neighboring nontarget class samples should be beneficial. For LDA, the between-class scatter matrix describes the overall distribution of all S classes, not just the target class and its neighboring nontarget samples. Thus, we use a local pairwise scatter matrix instead of the between-class scatter matrix to focus on impostor samples, which are most similar to the target speaker, with the goal of improving speaker verification performance.

B. Algorithm

The algorithm flow of the proposed LPLDA approach is the same as LDA, except that LDA needs to calculate the between-class scatter matrix, and LPLDA needs to calculate the local pairwise scatter matrix.

LPLDA needs to find confusable nontarget samples for each class. For the target class s with black dots shown in Fig. 1, the $n_{\bar{s}}$ out-of-class samples that are nearest to $\boldsymbol{\mu}_s$ but do not belong to class s are called confusable samples of class s, and denoted as $\mathbf{x}_j^{\bar{s}}$, where $1 \leq j \leq n_{\bar{s}}$. These confusable samples and the target class samples form a pair. Then local pairwise scatter matrix is defined as follows:

$$\begin{aligned}\mathbf{S}_{\text{lp}} &= \sum_{s=1}^{S} \frac{1}{2} \Bigg[\left(\boldsymbol{\mu}_s - \frac{\boldsymbol{\mu}_s + \boldsymbol{\mu}_{\bar{s}}}{2}\right)\left(\boldsymbol{\mu}_s - \frac{\boldsymbol{\mu}_s + \boldsymbol{\mu}_{\bar{s}}}{2}\right)^t \\ &\quad + \left(\boldsymbol{\mu}_{\bar{s}} - \frac{\boldsymbol{\mu}_s + \boldsymbol{\mu}_{\bar{s}}}{2}\right)\left(\boldsymbol{\mu}_{\bar{s}} - \frac{\boldsymbol{\mu}_s + \boldsymbol{\mu}_{\bar{s}}}{2}\right)^t \Bigg] \\ &= \frac{1}{4} \sum_{s=1}^{S} (\boldsymbol{\mu}_s - \boldsymbol{\mu}_{\bar{s}})(\boldsymbol{\mu}_s - \boldsymbol{\mu}_{\bar{s}})^t \end{aligned} \quad (4)$$

where $\boldsymbol{\mu}_{\bar{s}}$ is the mean vector of confusable samples of class s, $\boldsymbol{\mu}_{\bar{s}} = \frac{1}{n_{\bar{s}}} \sum_{j=1}^{n_{\bar{s}}} \mathbf{x}_j^{\bar{s}}$. The approach is called local pairwise LDA because the calculation of the scatter matrix for each class s emphasizes pairs formed by the class samples and the identified confusable samples. The aim of LPLDA is to maximize the covariance between target class samples and the associated confusable samples. Comparing S_b in (1) and S_{lp} in (4), we find that both are formed by a vector multiplied with its transpose, which is the subtraction of two means, so S_{lp} has a similar mathematical property to S_b, e.g., symmetry, positive (semi-) definite.

The number of confusable samples $n_{\bar{s}}$ has an impact on system performance. As shown in Fig. 1, the largest distance between in-class samples and $\boldsymbol{\mu}_s$ is denoted as $\max d_{i,s}$. The distance between out-of-class samples and $\boldsymbol{\mu}_s$ is denoted as $d_{o,s}$. The number of out-of-class samples with $d_{o,s} < \max d_{i,s}$ (out-of-class samples in the inner circle) is denoted as $n_{*,o,s}$. For choosing $n_{\bar{s}}$, two cases are considered. In the first case, $n_{*,o,s}$ is small (or zero), as shown in Fig. 1, which means there are few out-of-class samples located in the inner circle. In this case, $n_{\bar{s}}$ is set to be proportional to n_s, and $n_{\bar{s}} = k_1 n_s$. In the other case, $n_{*,o,s}$ is large, which means that there are many out-of-class samples located in the inner circle. In this case, $n_{\bar{s}}$ is set to be proportional to $n_{*,o,s}$, and $n_{\bar{s}} = k_2 n_{*,o,s}$, k_1 and k_2 are constants. The final $n_{\bar{s}}$ is chosen as the larger of the two cases, $n_{\bar{s}} = \max(k_1 n_s, k_2 n_{*,o,s})$. The influence of k_1 and k_2 on algorithm performance is studied in Section IV.

To calculate the distance between two samples, as all samples are represented by length-normalized vectors, the inner product of two vectors is used [1], [18]. It should be noted that a larger inner product indicates a smaller distance.

The difference between NDA and LPLDA includes the selection of nearest neighbors and the computation of the between-class covariance. The nearest neighbors are found for every sample x_i^s in NDA and for every class s in LPLDA. If we compare the between-class scatter matrices, the computational cost of (3) is significantly greater than that of (4).

IV. Experiments

To investigate the effect of LPLDA and LPLDA-PLDA backends, experiments are carried out on NIST SRE10 [19], SRE14 [20], and SRE16 [21] datasets. Both classical GMM-Ivector [1] and TDNN-Xvector [3] frontends are investigated.

A. SRE10

The SRE10 includes two types of conditions: core–core and coreext–coreext. The core–core condition 5 (core–core-c5) is a telephone–telephone task, containing 580 models, 678 test segments, and 30 204 trials. The coreext–coreext condition

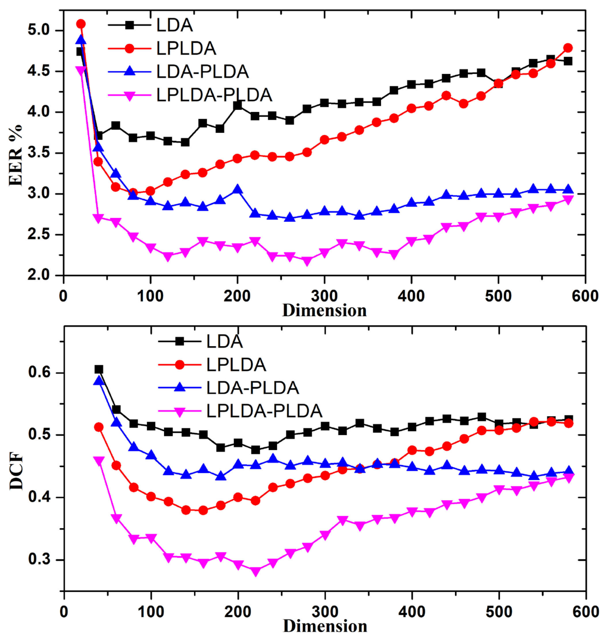

Fig. 2. Curves of EER and MDCF10 varied with the dimension on coreext–coreext c5 female condition.

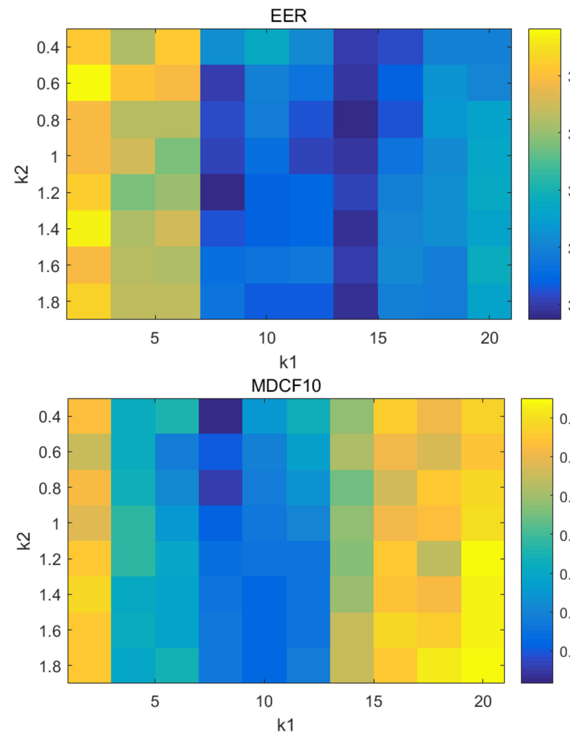

Fig. 3. Performance of LPLDA varies with k_1 and k_2 on coreext–coreext c5 female condition.

(coreext–coreext-c5) is also a telephone–telephone task, containing 4267 models, 767 test segments, and 416 119 trials. According to SRE10's evaluation plan, the detection cost function is computed with parameters $C_{\text{miss}} = 1$, $C_{\text{False Alarm}} = 1$, and $P_{\text{Target}} = 0.001$, which means that the SRE10 is characterized by evaluating system performance in the very low false alarm region.

Our GMM-Ivector frontend is configured as follows. A 13-dimensional static MFCC is extracted, with delta and delta–delta appended. The 39-dimensional vectors are subjected to cepstral mean and variance normalization. An energy-based voice active detection is applied. UBMs with 2048 gender-dependent Gaussian components are used. The rank of the total variability matrix is 600. LDA, LPLDA, and PLDA are all of dimension 150. We use NIST SRE 04-08 telephone utterances and Switchboard Phase II Part 1/2/3 and Cellular Part 1/2 to train all our models. This dataset comprises 6374 speakers, 64 742 utterances, and about 5524 h.

Figs. 2 and 3 are used for parameter tuning. Fig. 2 shows the EER and MDCF10 as a function of LDA and LPLDA dimension. As the dimension increases from 40 to 580 with a stride 20, the EER and MDCF first decrease and then increase. Based on this observation, in the following SRE10 experiments, the dimension is set to be 150. Fig. 3 shows the performance of LPLDA as a function of k_1 and k_2. Similar to the above results, as k_1 increases from 2 to 20 with a stride 2, the EER and MDCF first decrease and then increase, asynchronously. The result does not change much with k_2 because most of the training data fit the first case, where $n_{*,o,s}$ is small. For the rest of our experiments (SRE10, SRE14, and SRE16), $k_1 = 10$ and $k_2 = 1.2$ are kept.

From Table I, we can see that PLDA has an average (across condition and gender) relative improvement over LDA of 34.6% (EER) and 9.4% (MDCF10). LPLDA has an average relative improvement over LDA of 22.8% (EER) and 26.4% (MDCF10). Interestingly, this shows that PLDA is more effective at reducing EER, and LPLDA is more effective at reducing MDCF10.

TABLE I
EXPERIMENT RESULTS ON NIST SRE10

	core-core-c5		coreext-coreext-c5	
female	EER[%]	MDCF10	EER[%]	MDCF10
cosine	6.48	0.662	6.81	0.706
LDA	4.85	0.466	4.02	0.497
PLDA	2.82	0.391	3.01	0.476
LPLDA	3.66	0.329	3.24	0.358
LDA-PLDA	2.75	0.435	2.86	0.471
LPLDA-PLDA	**1.89**	**0.214**	**2.38**	**0.299**
male	EER[%]	MDCF10	EER[%]	MDCF10
cosine	6.80	0.484	6.75	0.612
LDA	4.25	0.427	3.76	0.458
PLDA	2.55	0.416	2.57	0.389
LPLDA	3.12	0.317	2.97	0.355
LDA-PLDA	2.83	0.313	2.57	0.353
LPLDA-PLDA	**2.27**	**0.263**	**2.24**	**0.274**

Combined, LPLDA-PLDA has an average relative improvement over LDA-PLDA of 20.1% (EER) and 31.4% (MDCF10). This combined result shows that the cascaded use of LPLDA and PLDA can significantly reduce EER and MDCF10 at the same time. The DET curve shown in Fig. 4 is more intuitive. The absolute value of the slope of the curves with LPLDA is lower than those without LPLDA, which further demonstrates that LPLDA is a very effective method, especially in the low false alarm region.

B. SRE14

The SRE14 Ivector challenge takes vectors instead of speech as input to compare the performance of speaker verification backends. The dataset for this task is gender independent, and contains 1306 speaker models, 9634 test segments, and 12 582 004 trials. Each speaker model has 5 Ivectors. The

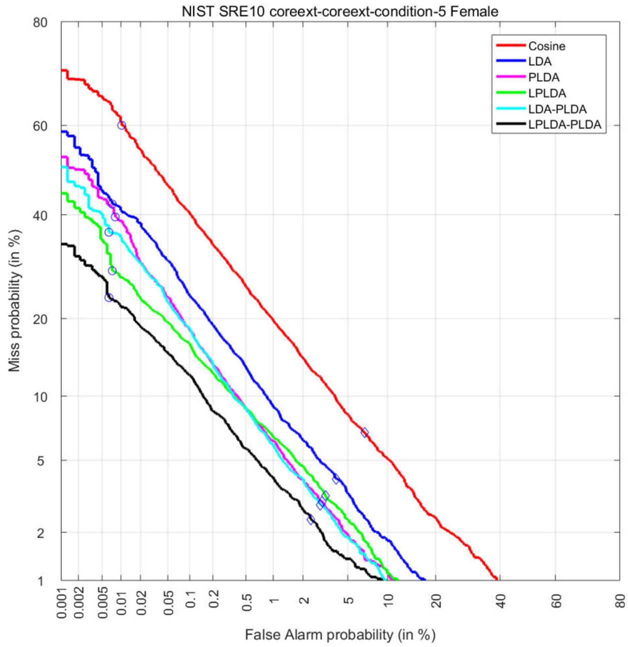

Fig. 4. DET curves on coreext–coreext c5 female condition.

TABLE II
EXPERIMENT RESULTS ON NIST SRE14

	Progress Set		Evaluation Set	
	EER[%]	MDCF14	EER[%]	MDCF14
cosine	4.78	0.386	4.46	0.378
LDA	4.44	0.343	4.04	0.332
LPLDA	4.05	0.335	3.86	0.321
PLDA	2.58	0.244	2.41	0.227
LDA-PLDA	2.18	0.239	2.09	0.228
LPLDA-PLDA	**1.96**	**0.210**	**1.86**	**0.194**

trials are randomly divided into a progress subset (40%) and an evaluation subset (60%). In addition, NIST provided a development set, containing 36 572 i-vectors. All Ivectors are 600-dimensional. Development data with labels are used to train LDA, LPLDA, and PLDA. The dimensions of LDA, LPLDA, and PLDA are 300.

Table II compares the performance of different systems on SRE14. It can been seen that LPLDA-PLDA has best performance. Compared with LDA-PLDA, the EER relative improvement is 10.1% and 11.0% on the progress and evaluation sets, respectively. The MDCF14 relative improvement of MDCF14 is 12.1% and 14.9%. To the best of our knowledge, the LPLDA-PLDA is the best single backend among works reported in the literatures [7], [20].

In order to give a more intuitive demonstration, we randomly select 80 speakers from SRE14 enrollment set and use t-SNE to visualize the effectiveness of LDA and LPLDA [22]. In Fig. 5, the points in the left subfigure are from LDA and the points in the right subfigure are from LPLDA. We can see that the points of the same speaker (same color) are more compactly clustered and the boundaries between different speakers are more clear in the right subfigure. In the left subfigure, the points of the same speaker are less compactly clustered and seem to spread out spatially, especially in the middle area. This characteristic may result in verification errors.

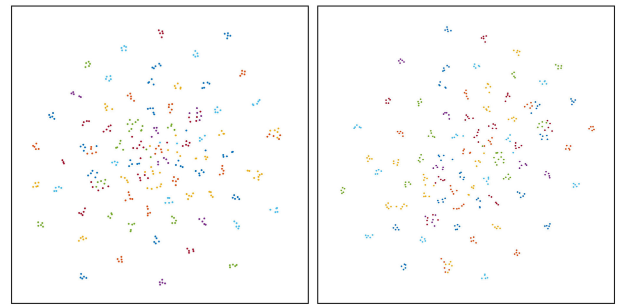

Fig. 5. Visualize dimensionality reduction of LDA (left) and LPLDA (right) with t-SNE.

TABLE III
EXPERIMENT RESULTS ON NIST SRE16

	Unequalized		Equalized	
Ivector	EER[%]	MDCF16	EER[%]	MDCF16
LDA-PLDA	15.34	0.915	15.91	0.908
LPLDA-PLDA	**14.47**	**0.882**	**15.12**	**0.875**
Xvector	EER[%]	MDCF16	EER[%]	MDCF16
LDA-PLDA	11.73	0.929	12.41	0.898
LPLDA-PLDA	**10.80**	**0.886**	**11.33**	**0.846**

We believe that the reason for the above situation is that the objective function of LDA focuses on making the overall variance between classes larger and does not try to minimize the confusion between them individually. In contrast, LPLDA focuses more on the covariance between target class samples and confusable nontarget samples, and tries to make this covariance larger and improve the verification performance.

C. SRE16

The SRE16 fixed training condition is challenging for three main reasons: language mismatch, duration mismatch, and increased duration variability [21]. We use Ivector and Xvector [3] to examine the performance of proposed LPLDA algorithm. The data usage and system configuration are given in [3] in detail, without data augmentation. Out-of-domain PLDA is considered. The dimensions of LDA, LPLDA, and PLDA are 150.

The experimental results of LDA-PLDA and LPLDA-PLDA are listed in Table III. Compared with LDA-PLDA, the relative improvement of the LPLDA-PLDA approach is not as large as for the SRE10 and SRE14 datasets. Despite this, LPLDA-PLDA still outperforms LDA-PLDA in both EER and MDCF16, under both Ivector and Xvector conditions. This further demonstrates that the LPLDA-PLDA approach works well with different frontends.

V. CONCLUSION

In this letter, we propose an LPLDA for speaker verification. A local pairwise scatter matrix is used for dimensionality reduction, which considers the local covariance between target class samples and neighboring nontarget class samples, instead of the traditional LDA approach that focuses on between-class covariance. Experimental results on SRE10, SRE14, and SRE16 show that, compared with LDA, this simple and novel LPLDA approach improves system performance. Overall, LPLDA-PLDA outperforms LDA-PLDA with a relative improvement of 20.1% EER and 31.4% MDCF10 on SRE10, and 11.1% EER and 12.9% MDCF14 on SRE14.

REFERENCES

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[2] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5115–5119.

[3] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. INTERSPEECH*, 2017, pp. 999–1003.

[4] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2014, pp. 1714–1718.

[5] C. Li *et al.*, "Deep speaker: An end-to-end neural speaker embedding system," 2017, arXiv:1705.02304v1.

[6] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Proc. INTERSPEECH*, 2017, pp. 1487–1492.

[7] O. Ghahabi and J. Hernando, "Deep learning backend for single and multisession i-vector speaker recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 4, pp. 807–817, Apr. 2017.

[8] D. G. Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker verification systems," in *Proc. INTERSPEECH*, 2011, pp. 249–252.

[9] S. Ioffe, "Probabilistic linear discriminant analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2006, pp. 531–542.

[10] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.

[11] M. McLaren and D. van Leeuwen, "Source-normalised-and-weighted LDA for robust speaker recognition using i-vectors," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 5456–5459.

[12] M. McLaren and D. van Leeuwen, "Source-normalized LDA for robust speaker recognition using i-vectors from multiple speech sources," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 3, pp. 755–766, Mar. 2012.

[13] A. Kanagasundaram, D. Dean, R. Vogt, M. McLaren, S. Sridharan, and M. Mason, "Weighted LDA techniques for i-vector based speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 4781–4784.

[14] S. O. Sadjadi, J. W. Pelecanos, and W. Zhu, "Nearest neighbor discriminant analysis for robust speaker recognition," in *Proc. INTERSPEECH*, 2014, pp. 1860–1864.

[15] S. O. Sadjadi, S. Ganapathy, and J. W. Pelecanos, "The IBM 2016 speaker recognition system," in *Proc. Odyssey*, 2016, pp. 174–180.

[16] L. He and J. Liu, "PRISM: A statistical modeling framework for text-independent speaker verification," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process.*, 2015, pp. 529–533.

[17] P. M. Bousquet, A. Larcher, D. Matrouf, J. F. Bonastre, and O. Plchot, "Variance-spectra based normalization for i-vector standard and probabilistic linear discriminant analysis," in *Proc. Odyssey*, 2012.

[18] G. Daniel and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. INTERSPEECH*, 2011, pp. 249–252.

[19] A. F. Martin and C. S. Greenberg, "The NIST 2010 speaker recognition evaluation," in *Proc. INTERSPEECH 2010, 11th Annu. Conf. Int. Speech Commun. Assoc.*, Makuhari, Japan, Sep. 2010, pp. 2726–2729.

[20] C. S. Greenberg *et al.*, "The NIST 2014 speaker recognition i-vector machine learning challenge," in *Proc. Odyssey 2014*, 2014, pp. 224–230.

[21] S. O. Sadjadi *et al.*, "The 2016 NIST speaker recognition evaluation," in *Proc. INTERSPEECH*, 2017, pp. 1353–1357.

[22] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 2, pp. 2579–2605, 2008.