Speaker-Aware Linear Discriminant Analysis in Speaker Verification

Naijun Zheng[1], *Xixin Wu*[1], *Jinghua Zhong*[2], *Xunying Liu*[1], *Helen Meng*[1]

[1] The Chinese University of Hong Kong, Hong Kong SAR, China
[2] SpeechX Limited, Shenzhen, China

{njzheng, wuxx, xyliu,hmmeng}@se.cuhk.edu.hk, jhzhong@speechx.cn

Abstract

Linear discriminant analysis (LDA) is an effective and widely used discriminative technique for speaker verification. However, it only utilizes the information on global structure to perform classification. Some variants of LDA, such as local pairwise LDA (LPLDA), are proposed to preserve more information on the local structure in the linear projection matrix. However, considering that the local structure may vary a lot in different regions, summing up related components to construct a single projection matrix may not be sufficient. In this paper, we present a speaker-aware strategy focusing on preserving distinct information on local structure in a set of linear discriminant projection matrices, and allocating them to different local regions for dimension reduction and classification. Experiments on NIST SRE2010 and NIST SRE2016 show that the speaker-aware strategy can boost the performance of both LDA and LPLDA backends in i-vector systems and x-vector systems.

Index Terms: Linear discriminant analysis (LDA), speaker verification, speaker-aware

1. Introduction

Speaker verification technologies have been developed for several decades and applied in many daily applications like voice assistants [1]. Gaussian Mixture Model (GMM) based i-vector systems [2], which are developed from the Joint Factor Analysis (JFA) [3] method, has laid the foundation for many state-of-the-art speaker recognition systems. In recent years, deep neural networks (DNN) have been used in place of GMM, such as in well-trained automatic speech recognition (ASR) neural networks [4]. Recently, end-to-end neural networks [5, 6] are also used in speaker verification, where the networks are directly trained to discriminate among speakers, and the outputs of the hidden layers are regarded as the embedding vectors for different speakers. These systems compress both channel and speaker information into a low-dimensional total variability space and represents variable-length utterances with low- and fixed-dimensional identifying vectors. Then, at the backend of the system, LDA [7] and probabilistic LDA (PLDA) [8] are applied on the extracted embedding vectors for inter-session compensation and classification [9, 10, 11].

Although the LDA method is applied in linear space, it is still an efficient and popular backend process in most classification systems. The idea behind LDA is to maximize the between-class covariance while minimizing the within-class covariance [12]. However, LDA only makes use of information based on global structure to compute the between-class covariance, while neglecting information on local structures. Recently, variants of LDA have been proposed, focusing on the utilization of information on local structure, such as nearest neighbor discriminant analysis (NDA) [13] and local pairwise linear discriminant analysis (LPLDA) [14]. The common idea behind these methods is to pay more attention to information on the local structure, i.e., focusing on nearby non-target samples rather than distant non-target samples. The NDA method incorporates this idea by introducing weight parameters according to distance when computing between-class covariance. In the LPLDA method, computation of the between-class covariance has been adapted towards measuring the distance between target samples and nearby non-target samples.

In the above methods, information on local structure is considered in the summation for computing between-class covariance to construct a single linear transformation space for all speakers. However, since the local structure information may vary across different regions [10, 15], a single projection matrix may not be sufficient to address this nonlinear problem. To make use of the distinctive information offered by the local structure, this paper presents a speaker-aware strategy to allocate different linear discriminant projection matrices to different speakers in the training dataset. More specifically — during training, for each speaker, the proposed method considers the utterances of neighboring speakers and gives different weights for computation of between-class covariances and the within-class covariances to obtain a set of linear discriminant projection matrices. Then, for testing, projection matrices in the corresponding local regions are selected for dimension reduction and classification scoring, which can better utilize the distinctive information in the local structure to boost the performance over existing LDA and LPLDA methods.

2. LDA and its variants

LDA is an efficient dimension reduction method, which utilizes class information for supervised classification. The key idea behind LDA is to minimize the ratio between the within-class covariance and between-class covariance. Let x_i^c denotes the i-th sample of class (speaker) c, the within-class scatter matrix $\mathbf{S_w}$ and between-class scatter matrix $\mathbf{S_b}$ can be represented as

$$\begin{aligned}\mathbf{S_w} &= \sum_{c=1}^{C}\sum_{i=1}^{N_c}(x_i^c-\mu_c)(x_i^c-\mu_c)^T \\ \mathbf{S_b} &= \sum_{c=1}^{C} N_c(\mu_c-\bar{\mu})(\mu_c-\bar{\mu})^T\end{aligned} \tag{1}$$

where C and N_c denote the number of the classes and the samples in class c respectively, μ_c is the mean of samples in class c and $\bar{\mu}$ denote the overall mean, that is, $\bar{\mu} = \sum_c N_c\mu_c / \sum_c N_c$. Then the projection matrix of LDA $\mathbf{W}$ can be obtained as:

$$\mathbf{W}_{opt} = \arg\max_{W} \operatorname{tr}((\mathbf{W^T S_w W})^{\mathbf{-1}} \mathbf{W^T S_b W}) \tag{2}$$

where $\operatorname{tr}\{\cdot\}$ denotes the matrix trace function, and columns of $\mathbf{W}_{opt}$ are the eigenvectors of $\mathbf{S_w^{-1} S_b}$ corresponding to the largest eigenvalues.

Since the LDA method only considers the global structure between the classes, it may not work well in difficult situations, e.g., verifying whether two closely positioned samples belong to the same class or not. Hence, variants of LDA have been proposed to focus on learning the local structure, such as nearest neighbor discriminant analysis (NDA) [13] and local pairwise linear discriminant analysis (LPLDA) [14].

NDA computes the between-class scatter matrix by giving different weights to neighboring samples, i.e.,[1]

$$\mathbf{S_b} = \sum_{c=1}^{C}\sum_{i=1}^{N_c} w_i^{c\bar{c}}(x_i^c - \bar{\mu}_i^{\bar{c}})(x_i^c - \bar{\mu}_i^{\bar{c}})^T, \tag{3}$$

where $\bar{\mu}_i^{\bar{c}}$ is the mean of the K-nearest samples of x_i^c outside of class c, and the weight parameter $w_i^{c\bar{c}}$ in (3) is defined as:

$$w_i^{c\bar{c}} = \frac{\min\{d(x_i^c, NN_K(x_i^c, c)), d(NN_K(x_i^c, \bar{c}))\}}{d(x_i^c, NN_K(x_i^c, c)) + d(NN_K(x_i^c, \bar{c}))}, \tag{4}$$

where $NN_k(x_i^c, c)$ and $NN_k(x_i^c, \bar{c})$ denote the k-th nearest sample of x_i^c from class c and outside of class c respectively, and $d(\cdot)$ denotes the cosine distance $d(a,b) = 1 - \cos(a,b)$. The within-class scatter matrix is computed in a similar way as (3) except for setting the weight to 1 and replacing $\bar{\mu}_i^{\bar{c}}$ with the mean of the K-nearest samples inside of the class c $\bar{\mu}_i^c$.

LPLDA learns the local structure by selecting confusable samples to construct negative classes with respect to each target (positive) class. For each target class, it draws an inner circle covering all the target samples with the class mean as the center point. Then, it counts the non-target samples in the inner circle to determine how many neighboring non-target samples should be selected to construct the negative class. In order to better discriminate the target class samples from selected confusable samples, a local pairwise scatter matrix is constructed. The scatter matrix $\mathbf{S_{lp}}$ can be computed as follows:

$$\mathbf{S_{lp}} = \sum_{c=1}^{C} N_c(\mu_c - \bar{\mu}_{\bar{c}})(\mu_c - \bar{\mu}_{\bar{c}})^T, \tag{5}$$

where $\bar{\mu}_{\bar{c}}$ is the mean vector of the negative class with respect to the target class c. The linear projection matrix can be computed with the eigenvectors of $\mathbf{S_w^{-1}S_{lp}}$.

3. Speaker-aware linear discriminant analysis

In the above methods, information about the local structure is captured in the summation during computation of the between-class scatter matrix in order to construct a single linear transformation space. Considering that the local structure information may not be consistent across different regions, only a single linear projection may not be able to capture the variability. We proposed a speaker-aware strategy in computing a set of linear projection matrices for the classes, which can be applied to existing LDA and LPLDA methods.

3.1. Strategy for LDA

For simplification, we use *sw-LDA* to denote the speaker-aware strategy for the LDA method. In computing the scatter matrices for speaker s, we will give a weight w_{sc} to denote the importance of another speaker c in relation with speaker s. The weight for each speaker can be viewed as a re-sampling rate to focus the linear transformation on neighboring samples for speaker s, while filtering out the distant samples. Then, the within-class and between-class scatter matrices for speaker s can be computed as:

$$\mathbf{S_w}(s) = \sum_{c=1}^{C}\sum_{i=1}^{N_c} w_{sc}(x_i^c - \mu_c)(x_i^c - \mu_c)^T. \tag{6}$$

$$\mathbf{S_b}(s) = \sum_{c=1}^{C} N_c w_{sc}(\mu_c - \hat{\mu}_s)(\mu_c - \hat{\mu}_s)^T, \tag{7}$$

where $\hat{\mu}_s$ is the weighted mean vector of the overall samples, which can be computed as $\hat{\mu}_s = \sum_{c=1}^{C} N_c w_{sc}\mu_c / \sum_{c=1}^{C} N_c w_{sc}$. Finally, the *sw-LDA* matrix $\mathbf{W}(s)$ for speaker s can be computed using the eigenvectors of $\mathbf{S_w}(s)^{-1}\mathbf{S_b}(s)$.

Since the set of weight values determines the balance between the global structure vis-à-vis local structure, the setting of weights will impact performance – this will be elaborated in Section 4.

3.2. Strategy for LPLDA

We use *sw-LPLDA* to denote LPLDA with speaker-aware strategy. Compared with LDA, LPLDA pays more attention on confusable samples. Based on Eq.(5), the scatter matrix $\mathbf{S_{lp}}(s)$ for speaker s can be computed as:

$$\mathbf{S_{lp}}(s) = \sum_{c=1}^{C} N_c w_{sc}(\mu_c - \bar{\mu}_{\bar{c}})(\mu_c - \bar{\mu}_{\bar{c}})^T. \tag{8}$$

The within-class scatter matrix $\mathbf{S_w}(s)$ is computed as Eq.(6). Then the *sw-LDA* matrix for speaker s can be computed with the eigenvectors of $\mathbf{S_w}(s)^{-1}\mathbf{S_{lp}}(s)$.

4. Weight setting and scoring methods

4.1. Weight setting

We first analyze the data distribution in the training set to inform weight setting. For every pair of speakers, we compute the cosine distances between their mean vectors, i.e., $D(s,c) = cos(\mu_s, \mu_c)$ and use them to estimate the distribution of the distance between the different utterances by considering the number of utterances $\{N_c\}$ of each speaker.

Figure 1 shows the distribution of the cosine distances between different utterances in the training dataset, where the embedding vectors are extracted from the DNN i-vector system. As we can see, the distribution density of the distance in the whole training dataset p_t can be modeled as a Gaussian model $\mathcal{N}(m \approx 0, \sigma^2)$ with approximately zero mean. The positive part of the distribution contains more information on the local structure, where two speakers can have closer distances and are generally more indistinguishable. In order to pay more attention to the positive part, we construct another Gaussian distribution $p_{tp} \sim \mathcal{N}(\sigma, \sigma^2)$ to give higher weights on the neighboring samples, where the mean is set as the standard deviation σ. The weight value $\{w_{sc}|c = 1...C, c \neq s\}$ for speaker s can be computed as:

[1] Different from the original formation, here we apply the one-versus-rest strategy used in [16], which can lead to better performance. (`http://www.cvc.uab.es/~jordi/nda.txt`)

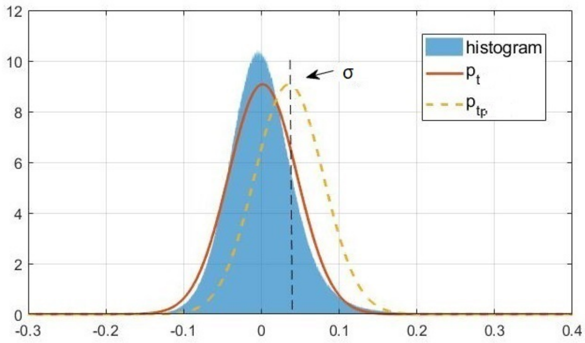


Figure 1: *Histogram of the cosine distances between the different utterances in the training dataset using a DNN i-vector system, where the solid curve is the estimated Gaussian density and the dashed curve is the constructed Gaussian density with mean σ.*

$$
\begin{aligned}
\hat{w}_{sc} &= \max\left\{\min\left\{\frac{p_{tp}(D(s,c))}{p_t^s(D(s,c))}, T_{max}\right\}, T_{min}\right\}, c \neq s \\
\hat{w}_{ss} &= \max\{w_{sc}\}_{c \neq s} \\
w_{sc} &= \frac{\hat{w}_{sc}}{\sum_c \hat{w}_{sc}}
\end{aligned}
\tag{9}
$$

where T_{min} and T_{max} are the lower bound and upper bound of the weight values to avoid overfitting, and p_t^s is estimated from single speaker s, that is, $p_t^s \sim \mathcal{N}(m_s, \sigma_s^2)$, $m_s = \frac{\sum_{c=1,c\neq s}^{C} N_c D(s,c)}{\sum_{c=1\neq s}^{C} N_c}$ and $\sigma_s^2 = \frac{\sum_{c=1,c\neq s}^{C} N_c (D(s,c)-m_s)^2}{\sum_{c=1,c\neq s}^{C} N_c}$.

4.2. Classification Scoring

After computing the set of linear projection matrices for all speakers in the training dataset, we can use them to verify whether the enrollment embedding vector x_e and the testing embedding vector x_t belong to the same speaker. The steps are

1. We first find the closet speakers s_e and s_t in the training dataset based on the enrollment vector and testing vector respectively, that is, $s_{e/t} = \arg_c \max cos(\mu_c, x_{e/t})$
2. Then we apply the linear projection matrices of these two speakers $\mathbf{W}(s_e)$ and $\mathbf{W}(s_t)$ to do transformation, that is:

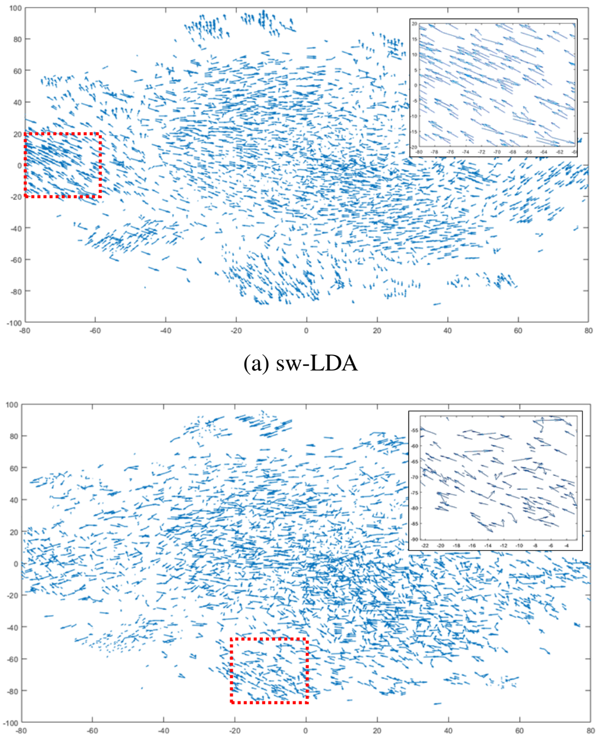
$$
\begin{aligned}
y_{ee} = \mathbf{W}(s_e)^T x_e, y_{et} = \mathbf{W}(s_t)^T x_e, \\
y_{te} = \mathbf{W}(s_e)^T x_t, y_{tt} = \mathbf{W}(s_t)^T x_t,
\end{aligned}
\tag{10}
$$

3. Finally, we take the average of the score(y_{ee}, y_{te}) and score (y_{et}, y_{tt}) as the final score.

When scoring two embedding vectors, we need to project them into the same linear spaces. Meanwhile, when applying PLDA as the scoring backend, we also need to compute a set of PLDA transformations using the corresponding weight values of each speaker to score them.

5. Experiments

The experiments are carried out on NIST SRE2010 [17] and NIST SRE2016 datasets [18]. In the experiments of SRE2010 dataset, both the GMM i-vector and DNN i-vector [19] systems are investigated, and performances in the coreext-coreext and core-core test conditions are discussed. Two short-duration test conditions (10sec-10sec, conv-10sec) are also investigated in the DNN i-vector system. NIST SRE2004∼2008 datasets and Switchboard Phase II part 1/2/3 and Cellular Part 1/2 are used to train the GMM and DNN models. The Mel-frequency cepstral coefficient (MFCC) is used as the input feature, and its dimensionality in GMM i-vector system is set to 39 with its delta and delta-delta deviations, while in DNN i-vector system it is set to 60. The universal background models (UBMs) in the GMM i-vector system are trained with 2048 gender-independent components. The i-vector dimensionality is set at 600. For backends training, only the SRE datasets are used, and the number of the speakers in the training dataset is 3805. The dimensionality for linear transformation is set at 200.[2]



(a) sw-LDA

(b) sw-LPLDA

Figure 2: *Local structure information learned in the projection matrices. The region in the red box is enlarged and shown in the top right corner.*

In the experiments with the SRE2016 dataset, the x-vector system [6] is investigated, where a well-trained time-delay neural network (TDNN) [3] is applied to extract the embedding vectors. NIST SRE2004∼2010 datasets are used to train the backend, and noise and reverberation additions are applied for data augmentation. The dimensionality for linear transformation is set to 150.

T_{min} and T_{max} in Eq.(9) are set to 1.5 and 10 empirically to obtain a proper balance between global structure and local structure. The parameters for NDA and LPLDA are set to be the same as that in [13] and [14].

[2] For more details, please refer to the codes (on Kaldi platform[20]) in `https://github.com/njzheng/LCLDA`

[3] `http://www.kaldi-asr.org/models/m3`

5.1. Information on local structure in projection matrices

In order to examine the information on local structure learned in different linear projection matrices, we select the first column (the eigenvector corresponding to the largest eigenvalue) of the linear matrices obtained for each speaker, and apply t-SNE [21] method to represent them as arrows on the 2-D plane (with length and direction), as shown in Figure 2. The coordinates of the arrows correspond to the speaker mean vectors which are also transformed by t-SNE method. As we can see in Figure 2(a), different local regions have different main directions, which means that the information on local structure is distinct and can be utilized individually to obtain higher discrimination. Figure 2(b) shows higher complexity compared with Figure 2(a), since *sw-LPLDA* focuses on discriminating nearby confusable samples, which can partition the space in greater detail.

5.2. Performance evaluation

In the following experiments, equal error rate (EER) and minimum detection cost function (mDCF)[17] are used to measure performance. In the GMM i-vector system, the results for coreext-coreext and core-core with condition 5 are shown in Table 1. The LDA and LPLDA applying the speaker-aware strategy show improvements in both EER and mDCF10.[4] In the coreext-coreext condition, with PLDA backend, the relative improvement of *sw-LPLDA* can reach at 13.5% in EER and 8.7% in mDCF10 over the original LPLDA method.

Table 1: *Results on NIST2010 using the GMM i-vector system*

	coreext-coreext		core-core	
	EER%	mDCF10	EER%	mDCF10
LDA	3.473	0.5116	3.814	0.5038
NDA	3.111	0.4304	**2.966**	0.4329
LPLDA	2.860	0.4239	3.107	0.4193
sw-LDA	3.125	0.4776	3.249	0.4784
sw-LPLDA	**2.762**	**0.3942**	3.107	**0.3993**
LDA-PLDA	1.855	0.3854	1.836	0.3670
NDA-PLDA	1.758	0.2867	1.836	0.3061
LPLDA-PLDA	1.646	0.2908	1.554	0.3442
sw-LDA-PLDA	1.688	0.3268	1.554	0.3343
sw-LPLDA-PLDA	**1.423**	**0.2656**	**1.412**	**0.2849**

In the DNN i-vector system, the improvement in the coreext and core test conditions are shown in Table 2. In the coreext-coreext test condition, the relative improvements over the original LDA and LPLDA setups are respectively at 13.6% and 10.9% in EER and 12.3% and 12.1% in mDCF10. Combining with the PLDA backend, the relative improvements can respectively reach 17.3% and 17.8% in EER, and 17.6% and 10.7% in mDCF10.

We also investigate the performance in short-duration test conditions with the DNN i-vector system, where the test utterances are much shorter than that in training dataset. The results are shown in Table 3, where *sw-LPLDA* can obviously improve the mDCF08[5] performance in both conditions.

Experimental results using the SRE2016 dataset are shown in Table 4 with the x-vector system. The dimensionality of the backend is set at 150, which are trained with data augmentation. Out-of-domain PLDA backends are considered. The primary measurement mDCF16 is defined as the average cost at two specific points on the detection cost function (DET) curve [18], and evaluations are conducted in equalized and unequalized modes. With the proposed speaker-aware strategy, improvements in EER and mDCF16 are consistent for both LDA and LPLDA methods, which shows the effectiveness of this strategy in the end-to-end speaker embedding system.

Table 2: *Results on NIST2010 using the DNN i-vector system*

	coreext-coreext		core-core	
	EER%	mDCF10	EER%	mDCF10
LDA	2.106	0.3359	2.260	0.3460
NDA	1.785	0.3171	2.260	0.2580
LPLDA	1.785	0.3071	2.119	0.2486
sw-LDA	1.827	0.2946	1.977	0.2835
sw-LPLDA	**1.590**	**0.2699**	**1.836**	**0.2373**
LDA-PLDA	1.046	0.2405	1.130	0.2667
NDA-PLDA	1.088	0.2184	1.130	0.2060
LPLDA-PLDA	1.018	0.2127	1.271	0.1681
sw-LDA-PLDA	0.865	0.1981	**0.848**	0.1554
sw-LPLDA-PLDA	**0.823**	**0.1899**	0.989	**0.1283**

Table 3: *Results on NIST2010 in short-duration conditions*

	conv-10sec		10sec-10sec	
	EER%	mDCF08	EER%	mDCF08
LDA	5.930	0.2764	11.72	0.5053
NDA	5.391	0.2437	**10.26**	0.4915
LPLDA	5.391	0.2342	10.62	0.4619
sw-LDA	**4.852**	0.2470	10.81	0.4803
sw-LPLDA	5.121	**0.2143**	10.44	**0.4471**
LDA-PLDA	3.774	0.2174	8.974	0.3778
NDA-PLDA	**3.504**	0.1957	8.242	0.3873
LPLDA-PLDA	3.774	0.2010	**8.059**	0.3731
sw-LDA-PLDA	**3.504**	0.1916	8.242	**0.3537**
sw-LPLDA-PLDA	3.774	**0.1699**	8.242	**0.3553**

Table 4: *Results on NIST2016 with the x-vector system*

	equalized		unequalized	
	EER%	mDCF16	EER%	mDCF16
LDA-PLDA	11.41	0.8580	11.21	0.8914
NDA-PLDA	10.87	0.8377	11.00	0.8644
LPLDA-PLDA	11.20	0.8280	11.21	0.8473
sw-LDA-PLDA	11.12	0.8509	11.15	0.8895
sw-LPLDA-PLDA	**10.57**	**0.8160**	**10.66**	**0.8424**

6. Conclusion

In this paper, a speaker-aware strategy is proposed at the backends of the GMM/DNN i-vector and x-vector systems. A set of local linear projection matrices are used to learn the distinctive information on the local structure in different regions. Experiments using the SRE2010 and SRE2016 datasets show that both LDA and LPLDA methods show consistent and significant improvements with the speaker-aware strategy. On the SRE10 dataset, *sw-LDA-PLDA* outperforms LDA-PLDA with a relative improvement of 13.7% in EER and 17.0% in MDCF, and *sw-LPLDA-PLDA* outperforms LPLDA-PLDA with a relative improvement of 10.3% in EER and 13.4% in MDCF.

7. Acknowledgements

This project is partially supported by the Theme-based Research Scheme of the Hong Kong SAR Government's Research Grants Council (Project No. T45-407/19N).

[4] mDCF10: $C_{miss} = 1, C_{fa} = 1$ and $P_{target} = 0.001$.

[5] mDCF08: $C_{miss} = 10, C_{fa} = 1$ and $P_{target} = 0.01$.

8. References

[1] W. Diao, X. Liu, Z. Zhou, and K. Zhang, "Your voice assistant is mine: How to abuse speakers to steal information and control your phone," in *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones; Mobile Devices*, ser. SPSM '14. ACM, 2014, pp. 63–74.

[2] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[3] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[4] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 1695–1699.

[5] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[6] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[7] A. M. Martínez and A. C. Kak, "Pca versus lda," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[8] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[9] M. Li and B. Yuan, "2D-LDA: A statistical linear discriminant analysis for image matrix," *Pattern Recognition Letters*, vol. 26, no. 5, pp. 527–532, 2005.

[10] T.-K. Kim and J. Kittler, "Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 3, pp. 318–327, 2005.

[11] A. Khosravani and M. M. Homayounpour, "Nonparametrically trained plda for short duration i-vector speaker verification," *Computer Speech & Language*, vol. 52, pp. 105–122, 2018.

[12] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data—with application to face recognition," *Pattern recognition*, vol. 34, no. 10, pp. 2067–2070, 2001.

[13] S. O. Sadjadi, J. Pelecanos, and W. Zhu, "Nearest neighbor discriminant analysis for robust speaker recognition," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[14] L. He, X. Chen, C. Xu, J. Liu, and M. T. Johnson, "Local pairwise linear discriminant analysis for speaker verification," *IEEE Signal Processing Letters*, vol. 25, no. 10, pp. 1575–1579, 2018.

[15] Z. Fan, Y. Xu, and D. Zhang, "Local linear discriminant analysis framework using sample neighbors," *IEEE Transactions on Neural Networks*, vol. 22, no. 7, pp. 1119–1132, 2011.

[16] S. O. Sadjadi, S. Ganapathy, and J. W. Pelecanos, "The IBM 2016 speaker recognition system," *arXiv preprint arXiv:1602.07291*, 2016.

[17] A. F. Martin and C. S. Greenberg, "The NIST 2010 speaker recognition evaluation," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[18] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. S. Greenberg, D. A. Reynolds, E. Singer, L. P. Mason, and J. Hernandez-Cordero, "The 2016 nist speaker recognition evaluation." in *Interspeech*, 2017, pp. 1353–1357.

[19] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[21] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.