Interspeech 2022
18-22 September 2022, Incheon, Korea

Unifying Cosine and PLDA Back-ends for Speaker Verification

Zhiyuan Peng[1,2], *Xuanji He*[2], *Ke Ding*[2], *Tan Lee*[1], *Guanglu Wan*[2]

[1]Department of Electronic Engineering, The Chinese University of Hong Kong
[2]Meituan

jerrypeng1937@gmail.com, {hexuanji,dingke02,wanguanglu}@meituan.com,
tanlee@ee.cuhk.edu.hk

Abstract

State-of-art speaker verification (SV) systems use a back-end model to score the similarity of speaker embeddings extracted from a neural network. The commonly used back-ends are the cosine scoring and the probabilistic linear discriminant analysis (PLDA) scoring. With the recently developed neural embeddings, the theoretically more appealing PLDA approach is found to have no advantage against or even be inferior to the simple cosine scoring in terms of verification performance. This paper presents an investigation on the relation between the two back-ends, aiming to explain the above counter-intuitive observation. It is shown that the cosine scoring is essentially a special case of PLDA scoring. In other words, by properly setting the parameters of PLDA, the two back-ends become equivalent. As a consequence, the cosine scoring not only inherits the basic assumptions for the PLDA but also introduces additional assumptions on speaker embeddings. Experiments show that the dimensional independence assumption required by the cosine scoring contributes most to the performance gap between the two methods under the domain-matched condition. When there is severe domain mismatch, the dimensional independence assumption does not hold and the PLDA would perform better than the cosine for domain adaptation.

Index Terms: speaker verification, cosine, PLDA, dimensional independence

1. Introduction

Speaker verification (SV) is the task of verifying the identity of a person from the characteristics of his or her voice. State-of-the-art SV systems are predominantly embedding based, comprising a front-end embedding extractor and a back-end scoring model. The front-end module transforms input speech into a compact embedding representation of speaker-related acoustic characteristics. The back-end model computes the similarity of two input speaker embeddings and determines whether they are from the same person.

There are two commonly used back-end scoring methods. One is the cosine scoring, which assumes the input embeddings are angularly discriminative. The SV score is defined as the cosine similarity of two embeddings x_1 and x_2, which are mean-subtracted and length-normalized [1], i.e.,

$$x_i \leftarrow \frac{x_i - \mu}{||x_i - \mu||_2}, \text{ for } i = 1, 2 \tag{1}$$

$$S_{\cos}(x_1, x_2) = x_1^T x_2 \tag{2}$$

The other method of back-end scoring is based on probabilistic linear discriminant analysis (PLDA) [2]. It takes the assumption that the embeddings (also mean-subtracted and length-normalized) are in general Gaussian distributed.

It has been noted that the standard PLDA back-end performs significantly better than the cosine back-end on conventional i-vector embeddings [3]. Unfortunately, with the powerful neural speaker embeddings that are widely used nowadays [4], the superiority of PLDA vanishes and even turns into inferiority. This phenomenon has been evident in some experimental studies [5, 6], especially when the front-end is trained with the additive angular margin (AAM)-softmax loss [7, 8].

The observation of PLDA being not as good as the cosine similarity is against the common sense of the back-end model design. Compared to the cosine, PLDA has more learnable parameters and incorporates additional speaker labels for training. Consequently, PLDA is generally considered to be more effective in discriminating speaker representations. This contradiction between experimental observations and theoretical expectation deserves thoughtful investigations on PLDA. Prior work [9–11] argued that the problem should arise from the neural speaker embeddings. It is noted in [11] that neural embeddings tend to be non-Gaussian for individual speakers and the distributions across different speakers are non-homogeneous. These irregular distributions cause the performance degradation of verification systems with the PLDA back-end. In relation to this perspective, a series of approaches have been proposed to regularize the neural embeddings [9–12].

In this paper, we try to present and substantiate a very different point of view from that in previous research. We argue that the suspected irregular distribution of speaker embeddings does not necessarily contribute to the inferiority of PLDA versus the cosine. Our view is based on the evidence that the cosine can be regarded as a special case of PLDA. This is indeed true but we have not yet found any work mentioning it. Existing studies have been treating the PLDA and the cosine scoring methods separately. We provide a short proof to unify them. The cosine scoring, as a special case of PLDA, also assumes speaker embeddings to be homogeneous Gaussian distributed. Therefore, if the neural speaker embeddings are distributed irregularly as previously hypothesized, both back-ends should exhibit performance degradation.

By unifying the cosine and the PLDA back-ends, it can be shown that the cosine puts stricter assumptions on the embeddings than the PLDA. Details of these assumptions are explained in Section 3. Among them, the dimensional independence (**dim-indep**) assumption, i.e.,

- Dimensions of speaker embeddings are mutually independent.

is found to play a key role in explaining the performance gap between the two back-ends. It is evidenced by incorporating the assumption into the training of PLDA, leading to the diagonal PLDA (DPLDA). This variant of PLDA shows a significant performance improvement under the domain-matched condition. However, when severe domain mismatch exists and

10.21437/Interspeech.2022-10021

back-end adaptation is needed, PLDA performs better than both the cosine and DPLDA. This is because the *dim-indep* assumption does not hold. Analysis on the between-/within-class covariance of speaker embeddings supports these statements.

In addition, consider the state-of-the-art SV system [13] using the front-end trained by AAM-softmax loss and the back-end of cosine scoring. We speculate that the AAM-softmax loss could regularize the front-end to generate *dim-indep* embeddings for domain-matched test data. When the test data significantly differs from the training data, the regularization on the extractor fails and the *dim-indep* assumption becomes invalid.

2. Review of PLDA

Theoretically PLDA is a probabilistic extension to linear discriminant analysis (LDA) [14]. It incorporates a Gaussian prior on the class centroids in LDA. Among the variants of PLDA, the two-covariance PLDA [15] has been commonly used in SV systems. A straightforward way to explain two-covariance PLDA is by using probabilistic graphical model [16].

2.1. Modeling

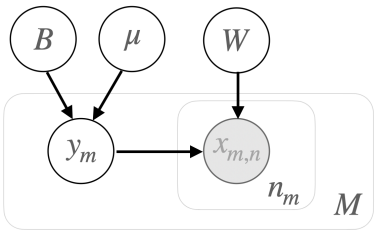

Figure 1: *The probabilistic graphical model of two-covariance PLDA*

Consider N speech utterances coming from M speakers, where the m-th speaker is associated with n_m utterances. With a front-end embedding extractor, each utterance can be represented by an embedding of D dimensions. The embedding of the n-th utterance from the m-th speaker is denoted as $x_{m,n}$. Let $\mathcal{X} = \{x_{m,n}\}_{1,1}^{M,n_m}$ represent these per-utterance embeddings. Additionally, PLDA supposes the existence of per-speaker embeddings $\mathcal{Y} = \{y_m\}_{m=1}^{M}$. They are referred to as latent *speaker identity variables* in [17].

With the graphical model shown in Fig.1, these embeddings are generated as follows,

- Randomly draw the per-speaker embedding $y_m \sim \mathcal{N}(y_m; \mu, B^{-1})$, for $m = 1, \cdots, M$;
- Randomly draw the per-utterance embedding $x_{m,n} \sim \mathcal{N}(x_{m,n}; y_m, W^{-1})$, for $n = 1, \cdots, n_m$.

where $\theta = \{\mu, B, W\}$ denotes the model parameters of PLDA. Note that B and W are precision matrices. The joint distribution $p_\theta(\mathcal{X}, \mathcal{Y})$ can be derived as,

$$p_\theta(\mathcal{X}, \mathcal{Y}) \propto \exp(-\frac{1}{2}\sum_{m=1}^{M}\Big[(y_m-\mu)^T B(y_m-\mu) + \sum_{n=1}^{n_m}(x_{m,n}-y_m)^T W(x_{m,n}-y_m)\Big]) \tag{3}$$

2.2. Training

Estimation of PLDA model parameters can be done with the iterative E-M algorithm, as described in Algorithm 1. The algorithm requires initialization of model parameters. In kaldi [18], the initialization strategy is to set $B = W = I$ and $\mu = 0$.

Algorithm 1 E-M training of two-covariance PLDA

Input: per-utterance embeddings $\mathcal{X} = \{x_{m,n}\}_{1,1}^{M,n_m}$
Initialization: $B = W = I, \mu = 0$
repeat
  (E-step): Infer the latent variable $y_m|\mathcal{X}$
    $L_m = B + n_m W$
    $y_m|\mathcal{X} \sim \mathcal{N}(L_m^{-1}(B\mu + W\sum_{n=1}^{n_m} x_{m,n}), L_m^{-1})$
  (M-step): Update θ by $\max_\theta \mathbb{E}_\mathcal{Y} \log p_\theta(\mathcal{X}, \mathcal{Y})$
    $\mu = \frac{1}{M}\sum_m \mathbb{E}[y_m|\mathcal{X}]$
    $B^{-1} = \frac{1}{M}\sum_m \mathbb{E}[y_m y_m^T|\mathcal{X}] - \mu\mu^T$
    $W^{-1} = \frac{1}{N}\sum_m \sum_n \mathbb{E}[(y_m - x_{m,n})(y_m - x_{m,n})^T|\mathcal{X}]$
until Convergence
Return B, W, μ

2.3. Scoring

Assuming the embeddings are mean-subtracted and length-normalized, we let $\mu \approx 0$ to simplify the scoring function. Given two per-utterance embeddings x_i, x_j, the PLDA generates a log-likelihood ratio (LLR) that measures the relative likelihood of the two embeddings coming from the same speaker. The LLR is defined as,

$$\begin{aligned} S_{\text{PLDA}}(x_i, x_j) &= \log \frac{p(x_i, x_j|\mathcal{H}_1)}{p(x_i, x_j|\mathcal{H}_0)} \\ &= \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)} \end{aligned} \tag{4}$$

where $\mathcal{H}_1$ and $\mathcal{H}_0$ represent the same-speaker and different-speaker hypotheses. To derive the score function, without loss of generality, consider a set of n_1 embeddings $\mathcal{X}_1 = \{x_{1,n}\}_{n=1}^{n_1}$ that come from the same speaker. It can be proved that

$$\log p(\mathcal{X}_1) = \frac{1}{2}\left(n_1^2\mu_1^T W(B + n_1 W)^{-1} W\mu_1 - \sum_{n=1}^{n_1} x_{1,n}^T W x_{1,n} + \log|B| + n_1\log|W| - \log|B + n_1 W| - n_1 D\log(2\pi)\right) \tag{5}$$

where $\mu_1 = \frac{1}{n_1}\sum_{n=1}^{n_1} x_{1,n}$. Details of proofs are provided[1]. By applying Eq.5 into Eq.4, the LLR can be expressed as

$$S_{\text{PLDA}}(x_i, x_j) \doteq \frac{1}{2}\left(x_i^T Q x_i + x_j^T Q x_j + 2x_i^T P x_j\right) \tag{6}$$

where $\doteq$ means equivalence up to a negligible additive constant, and

$$Q = W((B + 2W)^{-1} - (B + W)^{-1})W \tag{7}$$

$$P = W(B + 2W)^{-1}W \tag{8}$$

Note that $Q \prec 0$ and $P + Q \succeq 0$.

[1] https://github.com/JerryPeng21cuhk/Unifying-Cosine-and-PLDA-Back-ends-for-Speaker-Verification

3. Cosine as a typical PLDA

Relating Eq.6 to Eq.2 for the cosine similarity measure, it is noted that when $-Q = P = I$, the LLR of PLDA degrades into the cosine similarity, as $x_i^T x_i = 1$. It is also noted that the condition of $-Q = P = I$ is not required. PLDA is equivalent to the cosine if and only if $Q = \alpha I$ and $P = \beta I$, where $\alpha < 0, \alpha + \beta \geq 0$.

Given $W \succ 0$, we have

$$W = \frac{\beta(\beta - \alpha)}{-\alpha} I \tag{9}$$

$$B = \frac{\beta(\beta + \alpha)(\beta - \alpha)}{\alpha^2} I \tag{10}$$

Without loss of generality, we let $W = B = I$. In other words, the cosine is a typical PLDA with both within-class covariance W^{-1} and between-class covariance B^{-1} fixed as an identity matrix.

So far we consider only the simplest pairwise scoring. In the general case of many-vs-many scoring, the PLDA and cosine are also closely related. For example, let us consider two sets of embeddings $\mathcal{X}_1$ and $\mathcal{X}_2$ of size K_1 and K_2, respectively. Their centroids are denoted by μ_1 and μ_2. It can be shown,

$$S_{\text{PLDA}}(\mathcal{X}_1, \mathcal{X}_2) = \frac{K_1 K_2}{1 + K_1 + K_2} S_{\cos}(\mu_1, \mu_2) + \frac{1}{2} C(K_1, K_2) \tag{11}$$

$$C(K_1, K_2) = \frac{K_1^2 + K_2^2}{1 + K_1 + K_2} - \frac{K_1^2}{1 + K_1} - \frac{K_2^2}{1 + K_2} + \log(1 + \frac{K_1 K_2}{1 + K_1 + K_2}) \tag{12}$$

under the condition of $W = B = I$. The term $C(K_1, K_2)$ depends only on K_1 and K_2.

This has shown that the cosine puts more stringent assumptions than PLDA on the input embeddings. These assumptions are:

1. (**dim-indep**) Dimensions of speaker embeddings are mutually uncorrelated or independent;
2. Based on 1), all dimensions share the same variance value.

As the embeddings are assumed to be Gaussian, dimensional uncorrelatedness is equivalent to dimensional independence.

3.1. Diagonal PLDA

With Gaussian distributed embeddings, the *dim-indep* assumption implies that speaker embeddings have diagonal covariance. To analyse the significance of this assumption to the performance of SV backend, a diagonal constraint is applied to updating B and W in Algorithm 1, i.e.,

$$B^{-1} = \text{diag}(\frac{1}{M} \sum_m \mathbb{E}[y_m^{\circ 2} | \mathcal{X}] - \mu^{\circ 2}) \tag{13}$$

$$W^{-1} = \text{diag}(\frac{1}{N} \sum_m \sum_n \mathbb{E}[(y_m - x_{m,n})^{\circ 2} | \mathcal{X}]) \tag{14}$$

where $\circ 2$ denotes the Hadamard square. The PLDA trained in this way is named as the diagonal PLDA (DPLDA). The relationship between DPLDA and PLDA is similar to that between the diagonal GMM and the full-covariance GMM.

4. Experimental setup

Experiments are carried out with the Voxceleb1+2 [19] and the CNCeleb1 databases [20]. A vanilla ResNet34 [21] model is trained with 1029K utterances from 5994 speakers in the training set of Voxceleb2. Following the state-of-the-art training configuration[2], data augmentation with speed perturbation, reverberation and spectrum augmentation [22] is applied. The AAM-softmax loss [7] is adopted to produce angular-discriminative speaker embeddings.

The input features to ResNet34 are 80-dimension filterbank coefficients with mean normalization over a sliding window of up to 3 seconds long. Voice activity detection is carried out with the default configuration in kaldi[3]. The front-end module is trained to generate 256-dimension speaker embeddings, which are subsequently mean-subtracted and length-normalized. The PLDA backend is implemented in kaldi and modified to the DPLDA according to Eq. 13-14.

Performance evaluation is carried out on the test set in VoxCeleb1 and CNCeleb1. The evaluation metrics are equal error rate (EER) and decision cost function (DCF) with $p_{\text{tar}} = 0.01$ or 0.001.

4.1. Performance comparison between backends

As shown in Table 1, the performance gap between cosine and PLDA backends can be observed from the experiment on VoxCeleb. Cosine outperforms PLDA by relatively improvements of 51.61% in terms of equal error rate (EER) and 50.73% in terms of minimum Decision Cost Function with $P_{\text{tar}} = 0.01$ (DCF0.01). The performance difference becomes much more significant with DCF0.001. Similar results are noted on other test sets of VoxCeleb1 (not listed here for page limit).

The conventional setting of using LDA to preprocess raw speaker embeddings before PLDA is evaluated. It is labelled as *LDA+PLDA* in Table 1. Using LDA appears to have a negative effect on PLDA. This may be due to the absence of the *dim-indep* constraint on LDA. We argue that it is unnecessary to apply LDA to regularize the embeddings. The commonly used LDA preprocessing is removed in the following experiments.

Table 1: *Comparison of backends on VoxCeleb.*

	EER%	DCF0.01	DCF0.001
cos	**1.06**	**0.1083**	**0.1137**
PLDA	1.86	0.2198	0.3062
LDA+PLDA	2.17	0.2476	0.3715
DPLDA	1.11	0.1200	0.1426

The DPLDA incorporates the *dim-indep* constraint into PLDA training. As shown in Table 1, it improves the EER of PLDA from 1.86% to 1.11%, which is comparable to cosine. This clearly confirms the importance of *dim-indep*.

4.2. Performance degradation in Iterative PLDA training

According to the derivation in Section 3, PLDA implemented in Algorithm 1 is initialized as the cosine, e.g., $B = W = I$. However, the PLDA has been shown to be inferior to the cosine by the results in Table 1. Logically it would be expected that the performance of PLDA degrades in the iterative EM training.

[2]https://github.com/TaoRuijie/ECAPA-TDNN

[3]https://github.com/kaldi-asr/kaldi/blob/master/egs/voxceleb/v2/conf

Fig 2 shows the plot of EERs versus number of training iterations. Initially PLDA achieves exactly the same performance as cosine. In the first iteration, the EER seriously increases from 1.06% to 1.707%. For DPLDA, the *dim-indep* constraint shows an effect of counteracting the degradation.

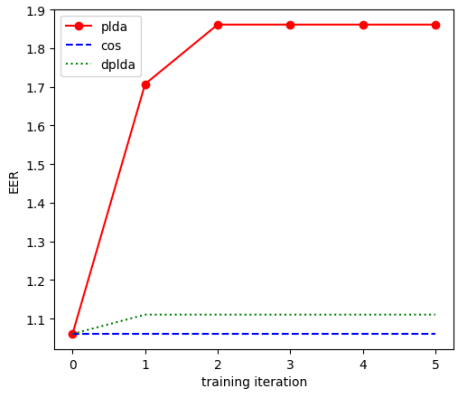

Figure 2: *PLDA gets worse in its iterative EM training*

4.3. When domain mismatch exists

The superiority of cosine over PLDA has been evidenced on the VoxCeleb dataset, of which both training and test data come from the same domain, e.g., interviews collected from YouTube. In many real-world scenarios, domain mismatch between training and test data commonly exists. A practical solution is to acquire certain amount of in-domain data and update the backend accordingly. The following experiment is to analyse the effect of domain mismatch on the performance of backend models.

The CNCeleb1 dataset is adopted as the domain-mismatched data. It is a multi-genre dataset of Chinese speech with very different acoustic conditions from VoxCeleb. The ResNet34 trained on VoxCeleb is deployed to exact embeddings from the utterances in CNCeleb1. The backends are trained and evaluated on the training and test embeddings of CNCeleb1.

As shown in Table2, the performance of both cosine and DPLDA are inferior to PLDA. Due to that the *dim-indep* assumption no longer holds, the diagonal constraint on covariance does not bring any performance improvement to cosine and DPLDA.

Table 2: *Comparison of backends on CNCeleb1*

	EER%	DCF0.01	DCF0.001
cos	10.11	0.5308	0.7175
PLDA	**8.90**	**0.4773**	**0.6331**
DPLDA	10.24	0.5491	0.8277

4.4. Analysis of between-/within-class covariances

To analyze the correlation of individual dimensions of the embeddings, the between-class and within-class covariances, B_0^{-1} and W_0^{-1}, are computed as follows,

$$B_0^{-1} = \frac{1}{M}\sum_{M} n_m y_m y_m^T - \mu_0 \mu_0^T \tag{15}$$

$$W_0^{-1} = \frac{1}{M}\sum_{m=1}^{M}\sum_{n=1}^{n_m}(x_{m,n} - y_m)(x_{m,n} - y_m)^T \tag{16}$$

where $\mu_0 = \frac{1}{N}\sum_{m=1}^{M}\sum_{n=1}^{n_m} x_{m,n}$ and $y_m = \frac{1}{n_m}\sum_{n=1}^{n_m} x_{m,n}$. These are the training equations of LDA and closely related to the M-step of PLDA. Note that for visualization, the elements in B_0^{-1} and W_0^{-1} are converted into their absolute value.

In Fig.3, both between-class and within-class covariances show clearly diagonal patterns, in the domain-matched case (plot on the top). This provides additional evidence to support the *dim-indep* assumption aforementioned. However, this assumption would be broken with strong domain-mismatched data in CNCeleb. As shown by the two sub-plots in the bottom of Fig 3, even though the within-class covariance plot on the right shows a nice diagonal pattern, it tends to vanish for the between-class covariance (plot on the left). Off-diagonal elements have large absolute value and the dimension correlation pattern appears, suggesting the broken of *dim-indep*. The numerical measure of diagonal index also confirms this observation.

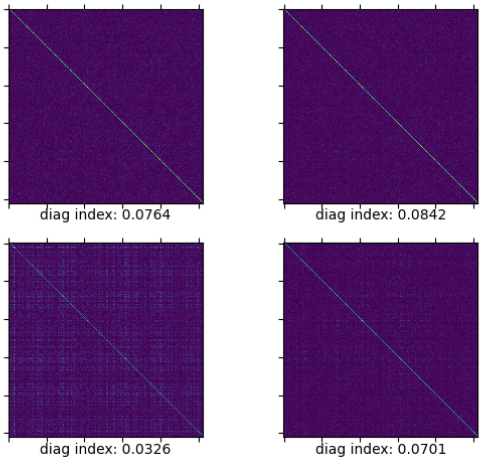

Figure 3: *between-class (left) and within-class (right) covariance of embeddings on the training data of VoxCeleb (top) and CN-Celeb (bottom). The diagonal index is computed as* $trace(G)/sum(G)$ *for a non-negative covariance matrix* G.

5. Conclusion

The reason why PLDA appears to be inferior to the cosine scoring with neural speaker embeddings has been exposed with both theoretical and experimental evidence. It has been shown that the cosine scoring is essentially a special case of PLDA. Hence, the non-Gaussian distribution of speaker embeddings should not be held responsible for explaining the performance difference between the PLDA and cosine back-ends. Instead, it should be attributed to the dimensional independence assumption made by the cosine, as evidenced in our experimental results and analysis. Further improvements on PLDA need to take this assumption into consideration.

6. References

[1] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Twelfth annual conference of the international speech communication association*, 2011.

[2] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

[3] 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.

[4] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "But system description to voxceleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[5] Q. Wang, K. A. Lee, and T. Liu, "Scoring of large-margin embeddings for speaker verification: Cosine or plda?" *arXiv preprint arXiv:2204.03965*, 2022.

[6] N. Brümmer, A. Swart, L. Mošner, A. Silnova, O. Plchot, T. Stafylakis, and L. Burget, "Probabilistic spherical discriminant analysis: An alternative to plda for length-normalized embeddings," *arXiv preprint arXiv:2203.14893*, 2022.

[7] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[8] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1652–1656.

[9] L. Li, Z. Tang, Y. Shi, and D. Wang, "Gaussian-constrained training for speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6036–6040.

[10] Y. Zhang, L. Li, and D. Wang, "Vae-based regularization for deep speaker embedding," *arXiv preprint arXiv:1904.03617*, 2019.

[11] Y. Cai, L. Li, A. Abel, X. Zhu, and D. Wang, "Deep normalization for speaker vectors," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 733–744, 2020.

[12] L. Li, D. Wang, and T. F. Zheng, "Neural discriminant analysis for deep speaker embedding," *arXiv preprint arXiv:2005.11905*, 2020.

[13] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[14] S. Balakrishnama and A. Ganapathiraju, "Linear discriminant analysis-a brief tutorial," *Institute for Signal and information Processing*, vol. 18, no. 1998, pp. 1–8, 1998.

[15] A. Sizov, K. A. Lee, and T. Kinnunen, "Unifying probabilistic linear discriminant analysis variants in biometric authentication," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 2014, pp. 464–475.

[16] M. I. Jordan, "An introduction to probabilistic graphical models," 2003.

[17] N. Brümmer and E. De Villiers, "The speaker partitioning problem." in *Odyssey*, 2010, p. 34.

[18] 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 *Proc. of ASRU*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[19] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[20] L. Li, R. Liu, J. Kang, Y. Fan, H. Cui, Y. Cai, R. Vipperla, T. F. Zheng, and D. Wang, "Cn-celeb: multi-genre speaker recognition," *Speech Communication*, 2022.

[21] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," *arXiv preprint arXiv:2003.11982*, 2020.

[22] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.