

# Scoring of Large-Margin Embeddings for Speaker Verification: Cosine or PLDA?

Qiongqiong Wang, Kong Aik Lee, Tianchi Liu

Institute for Infocomm Research (I<sup>2</sup>R), A\*STAR, Singapore

{wang-qiongqiong; lee-kong.aik; liu.tianchi}@i2r.a-star.edu.sg

## Abstract

The emergence of *large-margin softmax cross-entropy losses* in training deep speaker embedding neural networks has triggered a gradual shift from parametric back-ends to a simpler cosine similarity measure for speaker verification. Popular parametric back-ends include the *probabilistic linear discriminant analysis* (PLDA) and its variants. This paper investigates the properties of margin-based cross-entropy losses leading to such a shift, and aims to find scoring back-ends best suited for speaker verification. In addition, we revisit the pre-processing techniques which have been widely used in the past and assess their effectiveness on large-margin embeddings. Experiments on the state-of-the-art ECAPA-TDNN networks trained with various large-margin softmax cross-entropy losses show a substantial increment in intra-speaker compactness making the conventional PLDA superfluous. In this regard, we found that constraining the within-speaker covariance matrix could improve the performance of the PLDA. It is demonstrated through a series of experiments on the VoxCeleb-1 and SITW core-core test sets with 40.8% equal error rate (EER) reduction and 35.1% minimum detection cost (minDCF) reduction. It also outperforms cosine scoring consistently with reductions in EER and minDCF by 10.9% and 4.9%, respectively.

**Index Terms:** speaker verification, large-margin softmax, cosine similarity, PLDA, ECAPA-TDNN

## 1. Introduction

Automatic speaker verification (ASV) is the process to verify whether a given speech utterance is from a specific speaker or not. I-vector embedding [1] followed by *probabilistic linear discriminant analysis* (PLDA) [2, 3] was dominant in ASV for a long time until recent years when ASV started to benefit from deep learning. The use of deep neural networks (DNNs) has been investigated to replace individual components along the ASV pipeline, including the front-end feature extraction [4, 5], back-end modeling [6], and the entire pipeline in an end-to-end manner [7, 8]. Among these, using DNNs to extract discriminative speaker embeddings has been shown to be the most viable and effective. Therefore, recent works in ASV have focused on building network architectures that produce embedding vectors with improved speaker representations [4, 9–11].

A DNN for extracting an utterance-level speaker embedding consists of three modules: (1) a frame-level feature encoder, (2) a pooling layer, and (3) utterance-level representations. The input to the first module is a sequence of acoustic features, e.g., *mel-frequency cepstral coefficients* (MFCCs) and filter-bank coefficients. After considering relatively short-term acoustic features, this module outputs intermediate representa-

tions. Various neural network architectures have been used as the encoder, e.g., the *time-delay neural network* (TDNN) [4], convolutional neural network (CNN) [12], LSTM [13], the incorporation of LSTM to TDNN [5] or gated recurrent unit (GRU) [8]. The goal of this module is to extract more comprehensive speaker information. The second module converts variable-length frame-level intermediate features into a single fixed-dimensional vector by a temporal pooling. In addition to the most basic average statistics pooling, attention mechanism [14–16] is commonly used to form weighted statistics focusing on essential frames and in turn become more speaker discriminative. The third module stacks several fully-connected layers including one bottleneck layer used to extract utterance-level speaker embeddings with a fixed dimension in the testing phase. During training, the output nodes correspond to the set of speaker IDs in the training data. A softmax function is commonly used to constrain the predicted outputs so that they sum to one, and a cross-entropy (CE) loss is used to measure the network performance.

Good speaker embeddings should be discriminative between different speakers and compact within the same speaker. Embeddings learned using the conventional softmax CE loss, however, are optimized for only inter-speaker discrepancy. To address this issue, margin penalties have been introduced to the so-called large-margin softmax CE loss [17–19]. In this paper, we refer to the embeddings extracted from networks trained with margin penalties as the large-margin embeddings.

The emergence of large-margin embeddings has triggered a gradual shift from parametric back-ends, such as the PLDA, to a simpler cosine similarity measure [20, 21]. One possible reason is that a PLDA model decomposes the total variability into within and between-speaker covariance matrices [2, 3]. The intra-speaker compactness of the large-margin embeddings makes the within-speaker variability modeling no longer essential. However, as we noted, there is no prior experimental analysis. The goal of this paper is three-fold: (1) to study the properties of large-margin embeddings with respect to their predecessors, and to find (2) suitable scoring back-ends and (3) pre-processing techniques best suited for large-margin embeddings.

The paper is organized as follows. Section 2 reviews the large-margin softmax CE loss, as well as cosine similarity and PLDA back-ends. Section 3 introduces our investigations and motivations. Section 4 shows the experimental setup and results. Section 5 provides a summary of our work.

## 2. Large-Margin Embeddings for ASV

### 2.1. Softmax and Large-Margin Softmax

#### 2.1.1. Softmax Cross-Entropy Loss

The softmax function is often used as an activation function to calculate the relative probabilities to target classes in multi-way

Tianchi Liu is also with Department of Electrical and Computer Engineering, National University of Singapore, Singapore.

classification tasks. The cross-entropy (CE) loss could be calculated as:

$$L_1 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{\mathbf{W}_{y_i}^T \mathbf{x}_i + b_{y_i}}}{\sum_{j=1}^C e^{\mathbf{W}_j^T \mathbf{x}_i + b_j}} \quad (1)$$

where  $N$  is the batch size,  $C$  is the number of speakers in the training set,  $\mathbf{x}_i \in \mathbb{R}^d$  is the embedding representation of the  $i$ -th utterance, belonging to  $y_i$ -th class. The vector  $\mathbf{W}_j \in \mathbb{R}^d$  denotes  $j$ -th column of the weight matrix  $\mathbf{W} \in \mathbb{R}^{d \times C}$  while  $b_j \in \mathbb{R}^n$  is the corresponding bias term. The softmax function constrains the total probabilities to all the classes as 1, which helps training converge more quickly than it otherwise would. The expression  $\mathbf{W}_{y_i}^T \mathbf{x}_i$  in the numerator of (1) is equal to  $\|\mathbf{W}_{y_i}\| \|\mathbf{x}_i\| \cos(\theta_{y_i})$ , with the angle  $\theta_{y_i}$  between the vectors  $\mathbf{W}_{y_i}$  and  $\mathbf{x}_i$ . A modified softmax CE loss [22, 23] further normalizes the individual weight vector  $\|\mathbf{W}_j\| = 1$ , normalizes the embedding vector  $\|\mathbf{x}_i\|$  and re-scales to  $s$ , and discards the bias term:

$$L_2 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cdot \cos(\theta_{y_i})}}{\sum_{j=1}^C e^{s \cdot \cos(\theta_j)}}. \quad (2)$$

The modification enables the network to directly optimize angles and learn angularly distributed features, but not necessarily more discriminative ones [23].

### 2.1.2. Large-Margin Softmax Cross Entropy Loss

Since angles are used as the distance metric in (2), various techniques were introduced to incorporate margin penalties in order to enhance the speaker-discriminative power. They can be summarized with an angular function [19]

$$\psi(\theta_{y_i}) = \cos(m_1 \theta_{y_i} + m_2) - m_3 \quad (3)$$

where  $m_1$ ,  $m_2$  and  $m_3$  are the three margin penalties. Therefore, the larger margin softmax cross-entropy (CE) loss is

$$L_3 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cdot \psi(\theta_{y_i})}}{e^{s \cdot \psi(\theta_{y_i})} + \sum_{j=1, j \neq i}^C e^{s \cdot \cos(\theta_j)}}. \quad (4)$$

The margins  $\{m_1, m_2, m_3\}$  can be used simultaneously [19] or individually [17–19, 23], in which (4) is denoted, respectively, as the angular softmax (A-Softmax) [23]

$$\psi(\theta_{y_i}) = \cos(m_1 \theta_{y_i}), \quad (5)$$

the additive angular margin softmax (AAM-Softmax or ArcFace) [19]

$$\psi(\theta_{y_i}) = \cos(\theta_{y_i} + m_2), \quad (6)$$

and the additive margin softmax (AM-Softmax) [17]

$$\psi(\theta_{y_i}) = \cos(\theta_{y_i}) - m_3. \quad (7)$$

The margin penalties enforce intra-class compactness and inter-class discrepancy. This corresponds to a reduced within-speaker variability and a larger between-speaker variability in speaker recognition terminology. We refer to this class of representation as large-margin embeddings in this paper.

## 2.2. Speaker Verification

Speaker verification can be accomplished by calculating the similarity between the two speaker embeddings corresponding to an enrollment and test speech. To this end, a simple cosine distance measurement can be used. Alternatively, a more sophisticated scoring back-end can be trained such as the *probabilistic linear discriminant analysis* (PLDA).

### 2.2.1. Cosine Similarity

Cosine similarity scoring is a computationally efficient method in many verification tasks. When it is applied to speaker verification, the cosine of the angle between the enrollment ( $\phi_e$ ) and test ( $\phi_t$ ) embeddings is used as the decision score

$$s(\phi_e, \phi_t) = \frac{\langle \phi_e, \phi_t \rangle}{\|\phi_e\| \|\phi_t\|}. \quad (8)$$

This technique has an advantage that no training is required. Scoring is performed directly in the speaker embedding space.

### 2.2.2. PLDA

As opposed to cosine similarity measure, PLDA is a supervised method where speaker labels are necessary to train a PLDA model. There are multiple PLDA variants [2, 3, 24, 25]. Here we focus on the formulation reported in [2], which is widely used in speaker recognition [26, 27].

Let  $\phi$  be an embedding vector which we assume follows a Gaussian distribution [2, 3, 28]:

$$p(\phi | \mathbf{h}, \mathbf{x}) = \mathcal{N}(\phi | \mu + \mathbf{F}\mathbf{h} + \mathbf{G}\mathbf{x}, \Sigma), \quad (9)$$

where  $\mu \in \mathbb{R}^d$  is the global mean. The matrices  $\mathbf{F} \in \mathbb{R}^{D \times d}$  and  $\mathbf{G} \in \mathbb{R}^{D \times D}$  are, respectively, the speaker and channel loading matrices, and  $\Sigma$  models the residual variances and is constrained to be a diagonal matrix. The vectors  $\mathbf{h}$  and  $\mathbf{x}$  are the latent speaker and channel variables, respectively. Integrating out the latent variables, we arrive at the following marginal density

$$p(\phi) = \mathcal{N}(\phi | \mu, \Phi_B + \Phi_w) \quad (10)$$

where  $\{\Phi_B, \Phi_w\}$  are the between and within-speaker covariance matrices given by

$$\Phi_B = \mathbf{F}\mathbf{F}^T, \Phi_w = \mathbf{G}\mathbf{G}^T + \Sigma. \quad (11)$$

In the testing phase, the log-likelihood score between the enrollment ( $\phi_e$ ) and test ( $\phi_t$ ) embeddings is calculated as

$$s(\phi_e, \phi_t) = \log \frac{p(\phi_e, \phi_t)}{p(\phi_e)p(\phi_t)}. \quad (12)$$

Here, the joint likelihood in the numerator can be computed as

$$p(\phi_e, \phi_t) = \mathcal{N}\left(\begin{bmatrix} \phi_e \\ \phi_t \end{bmatrix} \middle| \begin{bmatrix} \mu \\ \mu \end{bmatrix}, \begin{bmatrix} \Phi_B + \Phi_w & \Phi_B \\ \Phi_B & \Phi_B + \Phi_w \end{bmatrix}\right), \quad (13)$$

while the likelihood  $p(\phi_e)$  and  $p(\phi_t)$  in the denominator are evaluated using (10). It is evident that PLDA scoring involves the explicit use of between and within covariance matrices, which is absent in cosine scoring.

## 3. Diagonalizing Within-Class Covariance Matrix

PLDA [2, 3] was originally introduced in ASV to work with i-vector framework [1]. Despite the i-vector front-end being replaced with more effective deep speaker embeddings, PLDA continues to be a promising back-end [1, 26, 29].

We study empirically the between and within-speaker covariance of the conventional x-vector embeddings [4] and large-margin embeddings from an ECAPA-TDNN [9]. Figure 1 shows that the within-speaker covariance of the conventional x-vector embeddings is larger than the between-speaker covariance in most of the dimensions in (a). In contrary, the between-speaker covariance is larger than the within-speaker covariance

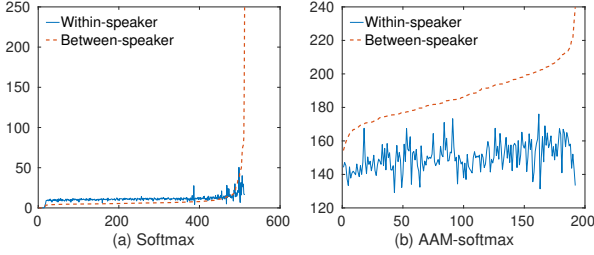


Figure 1: *Diagonal elements of the between and within-speaker covariance matrices of (a) conventional x-vector embeddings derived from a TDNN trained with softmax CE loss and (b) large-margin embeddings derived with an ECAPA-TDNN trained with AAM-Softmax CE loss. Values are sorted according to the between-speaker covariance matrices.*

for the large-margin embeddings in all the dimensions in (b). It indicates that the use of large-margin softmax CE loss efficiently reduces the intra-speaker variability (enhanced intra-speaker compactness) in the embedding space. We also observe that the between-speaker covariance of the large-margin embeddings are distributed more evenly across all of the dimensions, while in the conventional embeddings, high covariance values concentrate in certain dimensions only. This motivates us to constrain PLDA models to match the reduced within-speaker variability in large-margin embeddings. In our implementation, we set the within-speaker covariance as a diagonal matrix in each iteration of the expectation-maximization (EM) [30] steps in PLDA training. For the *linear discriminant analysis* (LDA) pre-processing technique, we also use a constrained variant which keeps only the diagonal elements in the within-speaker covariance matrix calculated from the data in the calculation of the LDA transformation matrix. In this paper, they are referred to as LDA-diag and PLDA-diag.

## 4. Experiments

### 4.1. Experimental settings

In order to verify the effectiveness of back-end techniques, the experiments are conducted on both VoxCeleb1 [12] and the Speaker in the Wild (SITW) core-core [31] test sets. For VoxCeleb1, we have exploited the original test set Vox1-o and the hard test set Vox1-h. All of our front-ends and parametric back-ends are trained on VoxCeleb2 dataset [32]. Approximately 2% of the train set is reserved for validation. Between our training and evaluation sets, there are no overlapping speakers. We employ augmentation techniques to produce a variety of the training data for the embedding networks, including random drops of audio chunks and frequency bands [33], speed perturbation [34], environmental corruptions with a collection of room impulse responses (RIRs) and noise [35]. For the parametric back-end training, a subset of VoxCeleb2 that consists of 300k utterances from 5,985 speakers is used with no augmentation, considering the training and testing data are in similar conditions.

We study several systems of state-of-the-art TDNN, ECAPA-TDNN and MFA-TDNN backbones with softmax, AAM-Softmax and AM-Softmax cross-entropy (CE) losses for comparisons [4, 9–11]. The pooling options are average and attentive statistics pooling and posterior inference pooling [10].

The details of combinations are shown in Table 1. We use SpeechBrain open-source toolkit [36] to implement all the front-ends and extract speaker embeddings. At the input of the neural networks, our systems utilize 80-dimensional filterbank features.

We evaluate three scoring methods: cosine similarity, PLDA and PLDA-diag, and also the effect of length normalization (LN) [24] and LDA as pre-processing steps for PLDA, as well as LDA-diag. The dimensions of LDA and LDA-diag are set to 150. Results are reported in terms of equal error rate (EER) and the minimum normalized detection cost function (MinDCF) at  $P_{target} = 10^{-2}$  and  $C_{FA} = C_{Miss} = 1$ .

### 4.2. Results and analysis

We first investigate the intra-speaker compactness in the conventional softmax embeddings (S6-S7 in Table 1) and the large-margin embeddings (S1-S5), respectively. Only LN is used before scoring as the pre-processing step. As shown in Table 1, for both S6 and S7, PLDA outperforms cosine similarity measure, while for the five systems (S1-S5) with different types of large margin softmax CE losses, cosine similarity measure achieves better performance than PLDA. These observations are consistent on all three evaluation sets. This indicates that the within-speaker variability in the conventional softmax embeddings are effectively reduced by channel compensation in PLDA, while the channel compensation is no longer essential for large-margin embeddings and even deteriorates the ASV performance. Figure 1 depicts the difference in the covariance plots between different embeddings. Both the results in Table 1 and the covariance plots show that the use of large-margin softmax CE loss efficiently reduces the intra-speaker variability in the embeddings. Comparing the front-ends, the large-margin embeddings (S1-S5) achieve much better performance than the conventional embeddings (S6, S7), which also confirms the efficiency of large-margin softmax in learning speaker-discriminative embeddings.

Next, we investigate the effectiveness using diagonal within-class covariance matrix (denoted as PLDA-diag) in Table 1. The use of the diagonalized within-speaker covariance in the PLDA model on the large-margin embeddings (S1-S5) reduces EER and minDCF on average by 40.8% and 35.1%, respectively, compared with the conventional PLDA with full within-class covariance matrix. Additionally, it outperforms cosine similarity consistently, reducing EER and minDCF on average by 10.9% and 4.9%, respectively. For conventional embeddings (S6, S7), on the contrary, PLDA-diag degrades both EER and minDCF compared with the conventional PLDA.

Taking ECAPA-TDNN as a front-end example of large-margin embeddings (S1 vs. S2), we further investigate the effect of embedding dimensions on ASV performance. Cosine similarity gives similar performance across the two dimensional embeddings, while the degradation produced by using the conventional PLDA in the 512-d embedding system S2 is almost double that of the 192-d embedding system S1. If we use PLDA-diag instead of PLDA, the performance improves and both systems have similar performance again.

Next we investigate the feasibility of the pre-processing techniques of PLDA on the large-margin embeddings. Since Vox1-h shows the same trend in the performance as Vox1-o, we exclude it considering the page limit. Figure 2 shows the effect of length normalization (LN) on the large-margin embeddings (S1) and the conventional embeddings (S6) with both PLDA and PLDA-diag back-ends on the Vox1-o and SITW core-core

Table 1: *EER and minDCF of the evaluations of three back-ends: cosine similarity, PLDA and PLDA-diag with five sets of large-margin embeddings (S1-S5), and two sets of the conventional softmax embeddings (S6, S7), on the three test sets: vox1-o, vox1-h and SITW core-core. The large-margin softmax includes AM- and AAM-Softmax CE losses. The backbones of the networks include: TDNN, ECAPA-TDNN and MFA-TDNN [11]. The pooling options are average and attentive statistics pooling and posterior inference pooling [10].*

ID	Backbone		Pooling	Loss	Dim	Vox1-o			Vox1-h			SITW		
	TDNN	MFA ECAPA				Cos	PLDA	PLDA- diag	Cos	PLDA	PLDA- diag	Cos	PLDA	PLDA- diag
S1 [9]	✓	✓	✓	✓	✓	1.28/0.177	1.91/0.261	<b>1.18/0.157</b>	2.47/0.241	3.75/0.331	<b>2.29/0.243</b>	1.83/0.167	2.35/0.240	<b>1.42/0.160</b>
S2		✓	✓	✓	✓	1.21/0.145	2.53/0.247	<b>1.08/0.129</b>	2.47/0.248	4.54/0.418	<b>2.32/0.249</b>	1.61/0.175	2.95/0.297	<b>1.48/0.169</b>
S3		✓	✓	✓	✓	1.22/0.127	1.75/0.193	<b>1.16/0.125</b>	2.57/0.251	3.74/0.348	<b>2.44/0.252</b>	1.92/0.184	2.49/0.242	<b>1.56/0.174</b>
S4 [10]		✓	✓	✓	✓	1.25/0.150	1.93/0.208	<b>1.16/0.136</b>	2.43/0.238	3.75/0.334	<b>2.27/0.239</b>	1.78/0.167	2.45/0.235	<b>1.39/0.158</b>
S5 [11]		✓	✓	✓	✓	1.14/0.132	1.56/0.208	<b>1.02/0.114</b>	2.26/0.225	3.35/0.299	<b>2.09/0.218</b>	1.56/0.156	2.07/0.229	<b>1.28/0.145</b>
S6		✓	✓	✓	✓	3.41/0.389	<b>2.46/0.283</b>	2.76/0.298	5.88/0.497	<b>4.36/0.394</b>	4.73/0.428	3.77/0.367	3.08/0.315	<b>2.71/0.313</b>
S7 [4]	✓		✓	✓	✓	6.86/0.637	<b>3.23/0.368</b>	5.81/0.505	12.42/0.778	<b>5.87/0.505</b>	9.97/0.643	13.72/0.892	<b>5.60/0.609</b>	13.22/0.964

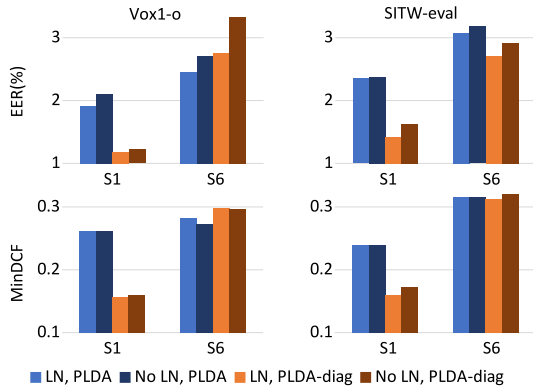


Figure 2: *Comparisons of EER and minDCF when using length normalization or not in embedding pre-processing*

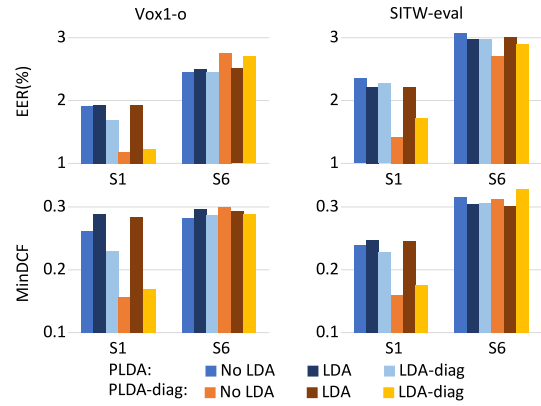


Figure 3: *Comparisons of EER and minDCF when using LDA or not in embedding pre-processing*

test sets. We observe that applying LN reduces both EER and minDCF in almost all systems. The performance improvement in EER is larger than that in minDCF. Therefore, we conclude that LN is still effective for large-margin speaker embeddings. We also note that with or without LN, PLDA-diag outperforms PLDA significantly. We have validated all the large-margin embeddings in Table 1 and obtained the same results.

Figure 3 shows the effect of LDA pre-processing technique on the same front-ends and back-ends. For the large-margin embeddings (S1), the use of the conventional LDA does not help in conventional PLDA systems, but drastically increases errors when applying to PLDA-diag systems. Applying LDA-diag to the PLDA systems improves the performance, however, much less than the improvement brought by using PLDA-diag directly. Applying it to the PLDA-diag system degrades the performance slightly. We conclude that for large-margin embeddings, removing the off-diagonal elements in the within speaker-covariance matrix in either LDA or PLDA improves speaker modeling. Using only PLDA-diag without LDA is sufficient to achieve good performance. For the conventional embeddings (S6), applying both LDA and LDA-diag does not greatly affect the performance. LDA helps when there is a slight mismatch between the SITW test set and the model training set.

## 5. Conclusions

This paper, for the first time, experimentally investigated the reasons of the shift from parametric back-ends to a simpler cosine similarity measure for the scoring of large-margin speaker embedding in speaker verification. Our experiments on the state-of-the-art ECAPA-TDNN networks with AAM-Softmax and AM-Softmax cross-entropy losses on VoxCeleb1 and SITW core-core test sets showed substantial increment in intra-speaker compactness making the conventional PLDA superfluous, while the cosine similarity scoring seems to be sufficient. We found that simply discarding off-diagonal elements in the within-speaker covariance matrix of the PLDA model improved the performance significantly with an average of 40.8% EER reduction and 35.1% minDCF reduction. It also outperformed cosine scoring consistently with reductions in EER and minDCF by 10.9% and 4.9%, respectively. In addition, this paper revisited the pre-processing techniques which have been widely used in the ASV back-ends in the past, and assessed their effects. In the future, we will investigate the evaluations in mismatch domains.

## 6. Acknowledgements

This project is supported Agency of Science, Technology and Research (A\*STAR), Singapore, through its Core Project Scheme (Project No. CR-2021-005).

## 7. References

- [1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," in *Proc. IEEE Transactions on Audio, Speech and Language Processing*, 2011, pp. 788–798.
- [2] S. Prince and J. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. ICCV*, 2007, pp. 1–8.
- [3] S. Ioffe, "Probabilistic linear discriminant analysis," in *Proc. ECCV*, 2006, pp. 531–542.
- [4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE ICASSP*, 2018, pp. 5329–5333.
- [5] Y. Tang, G. Ding, J. Huang, X. He, and B. Zhou, "Deep speaker embedding learning with multi-level pooling for text-independent speaker verification," in *Proc. IEEE ICASSP*, 2019, pp. 6116–6120.
- [6] J. Chien and C. Hsu, "Variational manifold learning for speaker recognition," in *Proc. IEEE ICASSP*, 2017, pp. 4935–4939.
- [7] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," in *arXiv:1705.02304*, 2017.
- [8] J. Rohdin, A. Silnova, M. Diez, O. Plchot, P. Matejka, L. Burget, and O. Glembek, "End-to-end DNN based text-independent speaker recognition for long and short utterances," in *Computer Speech Language*, 2020, pp. 22–35.
- [9] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Interspeech*, 2020, pp. 3830–3834.
- [10] K. A. Lee, Q. Wang, and T. Koshinaka, "Xi-vector embedding for speaker recognition," in *IEEE Signal Processing Letters*, 2021, p. 1385–1389.
- [11] T. Liu, R. Das, K. Lee, and H. Li, "MFA: TDNN with multi-scale frequency-channel attention for text-independent speaker verification with short utterances," in *Proc. IEEE ICASSP*, 2022.
- [12] A. Nagrani, J. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.
- [13] G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," in *Proc. Interspeech*, 2017, pp. 1517–1521.
- [14] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2018, pp. 3573–3577.
- [15] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018, pp. 2252–2256.
- [16] H. Zhu, K. A. Lee, and H. Li, "Serialized multi-layer multi-head attention for neural speaker embedding," in *Proc. Interspeech*, 2021, pp. 106–110.
- [17] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," in *IEEE Signal Processing Letters*, vol. 25, 2018, pp. 926–930.
- [18] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *IEEE CVPR*, 2018, pp. 5265–5274.
- [19] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *arXiv:1801.07698*, 2018.
- [20] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," in *Proc. Interspeech*, 2019, pp. 2873–2877.
- [21] D. Zhou, L. Wang, K. A. Lee, Y. Wu, M. Liu, J. Dang, and J. Wei, "Dynamic margin softmax loss for speaker verification," in *Proc. Interspeech*, 2020, pp. 3800–3804.
- [22] R. Ranjan, C. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," in *arXiv:1703.09507*, 2017.
- [23] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proc. IEEE CVPR*, 2017, pp. 212–220.
- [24] D. Garcia-Romero and C. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. Interspeech*, 2011, pp. 249–252.
- [25] N. Brummer and E. Villiers, "The speaker partitioning problem," in *Odyssey*, 2010, pp. 194–201.
- [26] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Odyssey*, 2010, pp. 14–21.
- [27] K. Lee, Q. Wang, and T. Koshinaka, "The CORAL+ algorithm for unsupervised domain adaptation of PLDA," in *Proc. IEEE ICASSP*, 2019, pp. 5821–5825.
- [28] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [29] P. Matejka, O. Glembek, F. Castaldo, M. J. Alam, O. Plchot, P. Kenny, L. Burget, and J. Cernocky, "Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification," in *Proc. IEEE ICASSP*, 2011, pp. 4828–4831.
- [30] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," in *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, 1977, pp. 1–22.
- [31] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The speakers in the wild (SITW) speaker recognition database," in *Proc. Interspeech*, 2016, pp. 818–822.
- [32] J. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.
- [33] D. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. Cubuk, and Q. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.
- [34] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015, pp. 3586–3589.
- [35] T. Ko, V. Peddinti, D. Povey, M. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. IEEE ICASSP*, 2017, pp. 5220–5224.
- [36] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell *et al.*, "SpeechBrain: A general-purpose speech toolkit," in *arXiv:2106.04624*, 2021.