THE CORAL++ ALGORITHM FOR UNSUPERVISED DOMAIN ADAPTATION OF SPEAKER RECOGNITION

Rongjin Li, Weibin Zhang, Dongpeng Chen

VoiceAI Technologies, Co. Ltd. Shenzhen, China

ABSTRACT

State-of-the-art speaker recognition systems are trained with a large amount of human-labeled training data set. Such a training set is usually composed of various data sources to enhance the modeling capability of models. However, in practical deployment, unseen condition is almost inevitable. Domain mismatch is a common problem in real-life applications due to the statistical difference between the training and testing data sets. To alleviate the degradation caused by domain mismatch, we propose a new feature-based unsupervised domain adaptation algorithm. The algorithm we propose is a further optimization based on the well-known CORrelation ALignment (CORAL), so we call it CORAL++. On the NIST 2019 Speaker Recognition Evaluation (SRE19), we use SRE18 CTS set as the development set to verify the effectiveness of CORAL++. With the typical x-vector/PLDA setup, the CORAL++ outperforms the CORAL by 9.40% relatively on EER.

Index Terms— Speaker recognition, speaker embedding, domain adaptation, unsupervised learning

1. INTRODUCTION

Speaker recognition is the task of recognizing a person's identity based on his or her voice [1]. In recent years, speaker recognition systems based on deep neural networks have achieved state-of-the-art performance in the community. Among them, the embedding-based methods can transform variable-length speech segments into fixed-dimensional vectors for scoring [2, 3]. Neural networks are able to discriminate tiny differences among different speakers by learning from an extensive collection of labeled training set. Various training data provide rich nuisance factors for the model, making it considerably more robust in complex environments.

However, when a speaker recognition system is deployed in real world, it has to face the cross-domain problem where the domains in which the system is deployed differ from that it was trained. And during testing, the speech of enrollment and verification may be collected from different domains that were not presented during training. In this case, the testing data is called 'in-domain' (inD) data while the training data is called 'out-of-domain' (ooD) data. Unexpected cross-domain problems, such as cross-channels, cross-lingual, cross-devices, cross-codecs, duration shift and time-drifting, damage the performance of conventional algorithms. Since it is impossible to enumerate all cross-domain situations and collect all corresponding labeled data into the training set, speaker recognition systems suffer from performance degradation when encountering new challenges. Furthermore, labelling in-domain data for system retraining is expensive and time-consuming. Therefore, this issue has attracted lots of attention from both the academic and the industry [4, 5]. Many experts and researchers focus on closing the gap between inD data and ooD data by using unlabeled inD data sets.

To compensate for the performance degradation, many unsupervised domain adaptation strategies for back-end modeling have been proposed. For the x-vector (or i-vector) /PLDA pipeline [6, 7, 8, 9], there are several main directions. For the model-based adaptation approaches, researchers aim to adapt the hyper-parameters of back-end models. In [10], the authors proposed to adapt the between-class and within-class covariance matrices of PLDA model. Kong Aik *et al.* proposed CORAL+ to compute the pseudo in-domain within and between class covariance matrices to regularize the corresponding matrices of PLDA [11]. Secondly, the feature-based adaptation algorithm is simple and effective without introducing new models, essentially providing more salient features for subsequent model-based methods. The CORAL algorithm is proposed to align the covariance between out-of-domain and in-domain embeddings via a whitening and re-coloring process [4, 5]. In [12], a feature-Distribution Adaptor (fDA) is proposed to avoid the influence of residual components and inaccurate information during adaptation. In addition, there are other methods to eliminate the domain-mismatch problem, such as neural network fine-tuning, metric learning loss functions in networks [13] and scoring framework [14].

In this work, we present an optimized CORrelation ALignment (CORAL) algorithm that works directly on raw embeddings. Hence, we refer to the new algorithm as CORAL++. CORAL++ focuses on aligning the second-order statistics, i.e., the covariance matrices, through a controllable regularization of residual components and a flooring constraint of normalized eigenvalue-spectrum. The raw covariance matrix estimated from sparse in-domain data is usually unreliable and contains many various nuisance factors in different in-domain data. It is impractical to directly use such raw covariance matrix to make adaptation, so our proposed algorithm makes the estimated in-domain covariance matrix more robust based on the assumption that larger eigenvalues (variances) are crucial to the target domain while small eigenvalues are unreliable. [12] proposed to filter out these worthless eigenvalues through a constant threshold. But it is difficult to find a suitable threshold for different data sets, especially when the in-domain data are gathered from multiple data sources. In order to emphasize the important eigenvalues more effectively and stably, we propose to use Z-score normalization on the eigenvalue spectrum and then apply a flooring constraint to remove those nuisance components. Then the in-domain covariance matrix is reconstructed to recolor the embeddings extracted from the out-of-domain data. Finally, the recolored embeddings are used to train the back-end models, e.g., PLDA. We carried out experiments on the NIST SRE19 CTS Challenge, and the corresponding development set includes the SRE18 CTS Dev and Eval sets [15].

The rest of this paper is organized as follows. Section 2 reports the theory of CORAL and fDA. Both are relevant to our work. In section 3, we discuss the details of domain adaptation and CORAL++. The experimental setup is presented in Section 4 while the results are listed in Section 5. Finally, we conclude in Section 6.

2. RELATED WORK

The typical setting of unsupervised domain adaptation is that there is a universal model and a small amount of unlabeled in-domain data. For the x-vector / PLDA pipeline paradigm, it is simple and effective to adopt a feature-based adaptive strategy on training embeddings.

2.1. Correlation Alignment

Addressing the domain-mismatch problem is critical for computer vision and speaker recognition. The main idea of CORrelation ALignment (CORAL) algorithm is to minimize the distance between the covariance of out-of-domain and in-domain embeddings [4, 5]. Suppose D_I and D_O are the D-dimensional embeddings of inD and ooD data sets respectively, i.e., $D_I \triangleq \{\vec{y_i}\}$, $\vec{y} \in \mathbb{R}^D$, and $D_O \triangleq \{\vec{x_i}\}$, $\vec{x} \in \mathbb{R}^D$. In addition, C_I and C_O are the covariance matrices of D_I and D_O respectively. The CORAL algorithm aims to find a transform matrix A that minimize the Frobenius norm between the transformed out-of-domain covariance matrix and the in-domain covariance matrix, i.e.,

$$\begin{aligned} A^* &= \arg\min_A \left\| C_{\widehat{O}} - C_I \right\|_F^2 \\ &= \arg\min_A \left\| A^\top C_O A - C_I \right\|_F^2 \end{aligned} \tag{1}$$

where $C_{\widehat{O}}$ is the transformed ooD covariance matrix. There is an analytic solution for the above function [4].

$$A^{*\top} = C_I^{\frac{1}{2}} C_O^{-\frac{1}{2}} \tag{2}$$

As can be seen, the optimal transformation matrix A^* can be further decomposed into two parts: the first part $C_O^{-\frac{1}{2}}$ whitens the ooD data while the second part $C_I^{\frac{1}{2}}$ re-colors it with the inD covariance matrix. As also suggested in [4], in practice an identity matrix is usually added to the covariance matrix to make it full rank for the sake of efficiency and stability. Thus we can perform the classical whitening and coloring. This is advantageous since: 1) it is faster as singular value decomposition (SVD) on the original covariance matrix might slow to converge; 2) the process is more stable. That is

$$A^{*\top} = (C_I + \mathbf{I})^{\frac{1}{2}} (C_O + \mathbf{I})^{-\frac{1}{2}} \tag{3}$$

A pseudocode of CORAL algorithm is presented in Algorithm 1.

Algorithm 1: CORAL for Unsupervised Domain Adaptation

Input: out-of-domain data D_O, in-domain data D_I
Output: Adapted out-of-domain data D_O^*
$C_O = cov(D_O) + eye(size(D_O, 2))$
$C_I = cov(D_I) + eye(size(D_I, 2))$
$D_O = D_O * C_O^{-\frac{1}{2}}$ % whitening out-of-domain data
$D_O^* = D_O * C_I^{\frac{1}{2}}$ % re-coloring in-domain data

2.2. Feature-Distribution Adaptor

In paper [12], the authors proposed a method to deal with the problem that the in-domain covariance matrix usually is not reliable in speaker recognition. In a typical x-vector/PLDA setup, generally, the covariance matrix is 512×512 and only several thousand samples are available to train the covariance matrix. The authors argued that only large eigenvalues reflect the true characteristics of in-domain data, and small eigenvalues are noisy and unreliable. A flooring mechanism is thus proposed to keep large components in the in-domain covariance matrix for re-coloring. Moreover, the authors proposed to firstly apply by-domain mean adaptation to inD and ooD embeddings to eliminate mean-shift in cross-domain applications. The whole algorithm is shown in Algorithm 2. The P and Δ are eigenvector and diagonal eigenvalues matrices, respectively.

Algorithm 2: feature-Distribution Adaptor

Apply by-domain mean adaptation to inD and ooD vectors.
Compute covariance matrices C_I, C_O of inD and ooD data.
Compute SVD of $C_O^{-\frac{1}{2}} C_I C_O^{-\frac{1}{2}} = P \Delta P^\top$
Compute matrix $\widehat{\Delta}$ such that $\widehat{\Delta}_{i,i} = max(1, \Delta_{i,i})$
For each ooD vector x do $x \leftarrow (C_O^{\frac{1}{2}} P \widehat{\Delta}^{\frac{1}{2}} P^\top C_O^{-\frac{1}{2}}) x$

For inD covariance, the feature-Distribution Adaptor computes eigenvalue-spectrum in the 'whitened' space, which appears to be efficient to retain the specific information of target domain.

3. THE CORAL++ ALGORITHM

The object of CORAL is to minimize the matrix distance between C_O and C_I, i.e., function (1). Analytically, the transformation matrix A can be decomposed into two parts $C_I^{\frac{1}{2}}$ and $C_O^{-\frac{1}{2}}$, and only the $C_I^{\frac{1}{2}}$ of re-coloring process is decided by in-domain data. Obviously, the effect of re-coloring is so critical. And given limited amount of development sets, the estimation of $C_I^{\frac{1}{2}}$ is probably not reliable, so we need to further optimize $C_I^{\frac{1}{2}}$. The assumption of CORAL++ is that large values of eigenvalue spectrum are reliable in C_I, while those small ones are not reliable and need to be filtered out.

CORAL++ focuses on optimizing the eigenvalues of C_I and then reconstructing C_I using the optimized eigenvalues. Specifically, given the D-dimensional symmetric covariance matrix C_I, we compute the eigenvalues through eigenvalue decomposition. The raw eigenvalues varies largely and it is hard for us to set a universal threshold to filter out those unimportant ones. On the other hand, only the relative importance of eigenvalues is useful for the re-coloring process. Thus, we first propose to normalize the eigenvalues to have zero mean and unit variance through Z-score normalization. Suppose the eigenvalues are s_i where $i = 1, 2, \cdots, D$. We have

$$\widehat{s_i} = \frac{s_i - \mu_s}{\sigma_s}, i = 1, 2, \cdots, D \tag{4}$$

where μ_s and σ_s are the mean and variance of eigenvalues respectively. After normalization, we can compare eigenvalues measured at different scales. It is easy to understand how good a certain eigenvalue is relative to the entire group. Then we set a universal threshold α to filter out those unreliable eigenvalues. That is

$$v_i = max(\alpha, \widehat{s}_i), i = 1, 2, \cdots, D \tag{5}$$

where α is a variable used to determine the retention of eigenvalue components. Those elements with very low or even negative (after

normalization) variances should be discarded by the $max(\cdot)$ operation. This step ensures that only those components with large variance are propagated to the transformation matrix A.

The traditional whitening is adding a small regularization parameter λ to the diagonal elements of covariance matrix to explicitly make it full rank. The authors in [4] argue that the performance of final system is insensitive to the value of λ in computer vision and thus an identity matrix $\mathbf{I}$ is used in [4]. However, we found that the flexible adjusting the hyper-parameter λ is helpful in domain-mismatch speaker recognition task. Adding a λ that is too large will compress the relative differences between variances.

By combining all the points above, the proposed CORAL++ algorithm is shown in Algorithm 3. The α and λ are the hyper-parameters and will be analyzed in Section 5.

Algorithm 3: CORAL++ for Unsupervised Domain Adaptation

Input: out-of-domain data D_O, in-domain data D_I
Output: Adapted out-of-domain data D_O^*

$C_O = cov(D_O)$
$C_I = cov(D_I)$
$EVD(C_I) \rightarrow P * diag(s) * P^\top$ % symmetric QR method
$\widehat{s} = Z(s)$ % Z-score on eigenvalues
$v_i = max(\alpha, \widehat{s}_i)$ % applying a flooring constraint
$\widehat{C}_O = C_O + \lambda * eye(size(D_O, 2))$
$\widehat{C}_I = P * diag(v) * P^\top + \lambda * eye(size(D_I, 2))$
$D_O = D_O * \widehat{C}_O^{-\frac{1}{2}}$ % whitening out-of-domain data
$D_O^* = D_O * \widehat{C}_I^{\frac{1}{2}}$ % re-coloring with in-domain data

As can be seen from Figure 1, the usage of CORAL++ is the same as CORAL. Both are used as the first module of back-end system [16]. The raw training embeddings are adapted by CORAL++ (or CORAL, or fDA) firstly, and then centering, principal component analysis (PCA), length normalization (LN), linear discriminant analysis (LDA) and Gaussian probability LDA (GPLDA) are successively trained by the adapted training embeddings. In addition, the CORAL+ algorithm is used to adapt the GPLDA model with the development embeddings and finally an adapted GPLDA is produced by interpolating the GPLDA with the CORAL+ model. The trials scores are then computed with the adapted GPLDA. To thoroughly compare different cross-domain adaptation techniques, we also used cosine distance scoring (CDS) in our experiments.

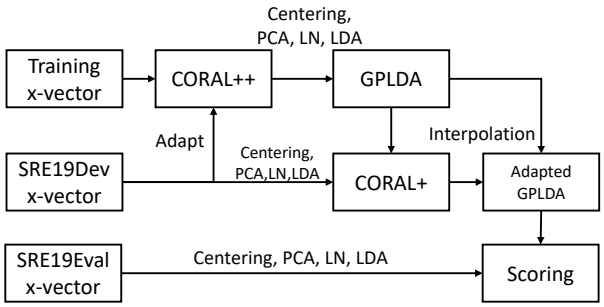

Fig. 1. Flow Diagram of Back-end Optimized Strategies

4. EXPERIMENTAL CONFIGURATION

We carried out our experiments with x-vectors of factorization time-delay neural networks (FTDNN) [17]. The effectiveness of CORAL++ is evaluated on the NIST SRE 2019 CTS Challenge.

4.1. Training setup

We used Switchboard (SWBD), NIST SREs, MIXER 6, Vox-Celeb, CN-Celeb [18] and Librispeech to train the neural networks. The SWBD corpora consist of SWBD 2 Phase 1, 2 and 3, and SWBD Cellular 1 and 2. The NIST SREs corpora consist of 2004 to 2010. For the MIXER6 corpus, we just used the telephone phone calls part. The Vox-Celeb and CN-Celeb corpora consist of 1 and 2, and we concatenated all segments of same utterance into a single one. The Librispeech corpora consists of train-clean and train-other parts. We also did data augmentation with additive noise and reverberation. The noise sets include MUSAN [19] and RIRS_NOISES [20]. Meanwhile, since encoding-decoding of speech is lossy during communication and storage, we used the MP3 codecs to simulate such a process. Finally, utterances that are shorter than 5 seconds and speakers with less than 8 utterances were all discarded.

All speech was down-sampled to 8KHz if the original recordings were 16KHz speech. The dimension of log Mel filter-banks (F-bank) was set to 64. All the features were extracted every 10ms with a 25ms shift window and the valid frequency was limited to 20-7600Hz. We applied the energy-based voiced activity detection (VAD) and cepstral mean-normalization (CMN) with a sliding window of up to 3 seconds on these acoustic features.

In the SRE19 CTS Challenge, the SRE18 data set, which consists of an unlabeled set (SRE18Dev) and labeled enroll-test sets (SRE18Eval), is used as the development set (named as SRE19Dev in Figure 1). We used the entire SRE19Dev as the in-domain data for adaptation and it contains 17,524 utterances. The SRE19Dev and SRE19Eval data are collected from 8KHz PSTN and VOIP and are spoken in Tunisian Arabic [15]. The training x-vectors extracted from the NIST SREs, Vox-Celeb 1 and CN-Celeb 1 datasets were used as the out-of-domain data to train back-end models. The out-of-domain data sets consist of multi-lingual (English, Chinese, etc.), multiple sampling rates (8KHz and 16KHz), multiple data sources (landline phone, web video, etc.) and so on. All the above make the domain mismatch problem in this task very challenging.

Meanwhile, in order to verify the consistency of hyper-parameters, we set two independent experimental groups where we just used SRE18Dev as the in-domain data to adapt the training x-vectors and it contains 4,073 utterances. And then we analyzed the corresponding results on the SRE18Eval.

4.2. Models setup

The FTDNN was trained by PyTorch [21], while acoustic features and back-end models were implemented with Kaldi [22]. The FTDNN we used is described in [16]. It reduces parameters by factorizing weight matrices in a semi-orthogonal manner. Skip connections are introduced between low-rank interior layers, where prior layers are concatenated to form the input of current layer. As for the objective function, we used both Softmax and AM-Softmax loss functions. AM-Softmax was used to minimize intra-speaker variation and maximize inter-speaker discrepancy [23, 24]. The margin was set to 0.35 and the scale was 64 for the AM-Softmax.

All network models were trained using the SGD optimizer and the cyclical learning rate (CLR) strategy based on the triangular2 policy. The weight decay of SGD was $3e^{-4}$, and the max and min learning rates were set at $1e^{-2}$ and $1e^{-4}$, respectively. We trained the network models for 4 epochs with a batch-size of 128. Then the 512-dimensional x-vectors were extracted for scoring. The PCA and

Table 3. Performance comparison of FTDNN x-vector systems with different domain adaptation methods on the NIST SRE19 CTS Challenge. The two evaluation metrics are EER(%)/min-Cost. The ratio is the percentage of random subset of available data. The fDA is the feature-Distribution Adaptor method. λ and α were set to 0.1 and 0.5 respectively for the CORAL++ algorithm.

Scoring		PLDA				Cosine			
Method	Ratio	raw	CORAL	fDA	CORAL++	raw	CORAL	fDA	CORAL++
Softmax	100%	6.47/0.453	6.47/0.454	7.01/0.486	**5.73/0.421**	7.66/0.494	7.64/0.493	8.56/0.548	**6.24/0.433**
AM-Softmax	100%	5.16/0.375	5.21/0.380	5.50/0.402	**4.72/0.354**	5.93/0.402	6.20/0.415	7.20/0.466	**4.99/0.366**
	50%	5.20/0.377	5.24/0.381	5.48/0.404	**4.75/0.355**	6.01/0.404	6.16/0.414	7.28/0.471	**4.98/0.364**
	10%	5.26/0.380	5.42/0.390	5.58/0.404	**4.80/0.359**	6.01/0.409	6.57/0.429	7.33/0.470	**5.07/0.369**

LDA reduced the dimension of x-vectors from 512 to 200, and from 200 to 100, respectively. There are no fine-tuning of network and no score normalization for all scoring results.

5. RESULTS

All systems were evaluated on the SRE18 and SRE19 evaluation set. Metrics used for performance measurement is equal error rate (EER) and minimum detection cost (min-Cost). We used the same parameters to calculate EER and min-Cost as in [15].

CORAL++ includes two parameters, i.e., λ and α. We carried out experiments to see how the system performance is influenced by these hyper-parameters. There are some constraints on both λ and α. Firstly, λ must be positive to ensure that both $C_I + \lambda\mathbf{I}$ and $C_O + \lambda\mathbf{I}$ are of full rank. Secondly, α must be non-negative since negative components of eigenvalue spectrum are not allowed.

In Table 1, we chose one of x-vector systems for analysis, i.e., the FTDNN-AMSoftmax/PLDA system. We set the flooring constraint α to be a constant 0 and varied λ. We could see that larger regularization parameter λ results to better performance when λ is smaller than 3.0. λ is added to the diagonal elements of covariance matrices to explicitly make them full rank, and it improve the generalization ability of model.

Table 1. Sensitivity of Covariance Regularization Parameter λ on the NIST SRE18 and SRE19 CTS Challenge. The metric is EER(%). The α is fixed to 0.

λ	0.1	0.5	1.0	1.5	2.0	2.5	3.0
SRE18Eval	6.30	5.61	5.58	**5.56**	5.58	5.63	5.66
SRE19Eval	5.80	5.00	4.83	4.81	**4.80**	4.83	4.86

We continued to carry out experiments where λ was fixed. The results are shown in Table 2. We found that the best value for α is much smaller than that of λ. α is used to filter out unreliable components from the eigenvalue spectrum. If α is too large, it would filter out some meaningful components. We recommend to set $\alpha = 0.5$. We also found that after the proposed Z-score normalization, the eigenvalues follow the normal distribution and the adjustment of α becomes easier and more stable across of different datasets.

Table 2. Sensitivity of Flooring constraint Parameter α on the NIST SRE18 and SRE19 CTS Challenge. The metric is EER(%). The λ is fixed to 0.1.

α	0.1	0.5	1.0	1.5	2.0	2.5	3.0
SRE18Eval	5.81	**5.53**	5.56	5.71	5.85	5.98	6.06
SRE19Eval	5.20	**4.72**	4.73	4.81	4.88	4.97	5.06

Following the above experiments, we chose $\lambda = 0.1$ and $\alpha = 0.5$ for the CORAL++ algorithm. In Table 3, we compared the x-vector systems with different domain adaptation algorithms on the SRE19 CTS Challenge. The "raw" method means that we did not use any adaptation method for the training embeddings in Figure 1. As can be seen, no matter the PLDA scoring or the cosine scoring are used, we can almost draw the same conclusion for the four adaptation techniques in comparison. The PLDA is superior to the cosine distance scoring in cross-domain problems for four different approaches. The PLDA can compensate for channel differences and it can be combined with other model-based adaptation algorithms to further enhance performance, such as the CORAL+. Therefore, we will focus on the PLDA scoring below. The performance of fDA is disappointing. We found that the fixed parameter flooring constraint (i.e., 1) in Algorithm 2 cannot effectively highlight the vital components. The AM-Softmax comparison group is much better than the Softmax group. We might draw a conclusion that increasing the inter-speaker variance and decreasing the intra-speaker variance help overcome the cross-domain problem.

As for the CORAL++, it achieves the best performance in different settings. When the Softmax is used, the CORAL++ is better than the CORAL by 11.44% and 18.32% relatively and respectively on EER when PLDA and CDS are used. When the AM-Softmax is used, the CORAL++ outperforms the "raw" method by 8.53% and 15.85% relatively and respectively on EER when PLDA and CDS are used. Furthermore, we randomly sampled 50% and 10% of SRE19Dev to compare the performance. For each percentage ratio, we randomly did three experiments and use the median of outputs. We found that only a subset of in-domain data is used can improve the performance by CORAL++, which proves to be stable.

6. CONCLUSION

In this study, we focused on how to effectively align the second order statistics of in-domain and out-of-domain data through unsupervised adaptation on back-end models. The proposed CORAL++ algorithm is highly efficient and reliable to handle complex cross-domain speaker recognition tasks without requiring labeled data.

CORAL++ has been examined on the well-known NIST SRE19 CTS challenge and yields excellent results consistently. Suppressing the smaller eigenvalues of covariance matrix and highlighting the larger ones helps alleviate the cross-domain problem of sparse in-domain data. We believe that the core idea of CORAL++ could also provide hints for the improvement of neural network training and adaptation in future works.

7. ACKNOWLEDGEMENT

This work is partially supported by the key research and development program of Guangdong Province (2019B010154003), the fundamental research funds of IFS, China(2021JB019).

8. REFERENCES

[1] Douglas A Reynolds, "An overview of automatic speaker recognition technology," in *2002 IEEE international conference on acoustics, speech, and signal processing*. IEEE, 2002, vol. 4, pp. IV–4072.

[2] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[3] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Interspeech*, 2017, pp. 999–1003.

[4] Baochen Sun, Jiashi Feng, and Kate Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016, vol. 30.

[5] Md Jahangir Alam, Gautam Bhattacharya, and Patrick Kenny, "Speaker verification in mismatched conditions with frustratingly easy domain adaptation.," in *Odyssey*, 2018, vol. 2018, pp. 176–180.

[6] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev 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] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[8] Simon JD Prince and James 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] Sergey Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

[10] Daniel Garcia-Romero, Alan McCree, Stephen Shum, Niko Brummer, and Carlos Vaquero, "Unsupervised domain adaptation for i-vector speaker recognition," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2014, vol. 8.

[11] Kong Aik Lee, Qiongqiong Wang, and Takafumi Koshinaka, "The coral+ algorithm for unsupervised domain adaptation of plda," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5821–5825.

[12] Pierre-Michel Bousquet and Mickael Rouvier, "On robustness of unsupervised domain adaptation for speaker recognition," in *InterSpeech*, 2019.

[13] Raphaël Duroselle, Denis Jouvet, and Irina Illina, "Metric learning loss functions to reduce domain mismatch in the x-vector space for language recognition," in *INTERSPEECH 2020*, 2020.

[14] Lantian Li, Dong Wang, Jiawen Kang, Renyu Wang, Jing Wu, Zhendong Gao, and Xiao Chen, "A principle solution for enroll-test mismatch in speaker recognition," *arXiv preprint arXiv:2012.12471*, 2020.

[15] Seyed Omid Sadjadi, Craig Greenberg, Elliot Singer, Douglas Reynolds, Lisa Mason, Jaime Hernandez-Cordero, et al., "The 2019 nist speaker recognition evaluation cts challenge," in *Speaker Odyssey*, 2020, vol. 2020, pp. 266–272.

[16] Rongjin Li, Dongpeng Chen, and Weibin Zhang, "Voiceai systems to nist sre19 evaluation: Robust speaker recognition on conversational telephone speech," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6459–6463.

[17] Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks.," in *Interspeech*, 2018, pp. 3743–3747.

[18] Yue Fan, JW Kang, LT Li, KC Li, HL Chen, ST Cheng, PY Zhang, ZY Zhou, YQ Cai, and Dong Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7604–7608.

[19] David Snyder, Guoguo Chen, and Daniel Povey, "MUSAN: A Music, Speech, and Noise Corpus," *arXiv:1510.08484v1*, 2015.

[20] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[22] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number CONF.

[23] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[24] Rongjin Li, Na Li, Deyi Tuo, Meng Yu, Dan Su, and Dong Yu, "Boundary discriminative large margin cosine loss for text-independent speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6321–6325.