SQUEEZING VALUE OF CROSS-DOMAIN LABELS: A DECOUPLED SCORING APPROACH FOR SPEAKER VERIFICATION

Lantian Li, Yang Zhang, Thomas Fang Zheng, Dong Wang

Center for Speech and Language Technologies, Tsinghua University

ABSTRACT

Domain mismatch is often occurred in real applications and causes serious performance reduction on speaker recognition systems. The common wisdom is to collect cross-domain data and train a multi-domain PLDA model, with the hope to learn a domain-independent speaker subspace. In this paper, we firstly present an empirical study to show that simply adding cross-domain data does not help performance in conditions with enroll-test mismatch. Careful analysis shows that this striking result is caused by the incoherent statistics between enroll and test conditions. Based on this analysis, we present a decoupled scoring approach that can maximally squeeze the value of cross-domain labels and obtain optimal verification scores when the enroll and test are mismatched. When the statistics are coherent, the new formulation falls back to the conventional PLDA. Experimental results on cross-channel test show that the proposed approach is highly effective and is a principle solution to domain mismatch.

Index Terms— speaker verification, domain mismatch, decoupled scoring

1. INTRODUCTION

Speaker recognition aims to recognize claimed identities of speakers. After decades of research, current speaker recognition systems have achieved rather satisfactory performance [1], especially with the i-vector model [2], or embedding models based on deep neural nets (DNN), e.g., the x-vector model [3, 4]. The deep embedding models have been significantly improved recently, by employing advanced techniques such as novel architectures [5, 6], attention pooling [4, 7, 8], or max-margin training [9, 10, 11, 12]. As a result, deep learning models have achieved state-of-the-art performance on several benchmark datasets [13], in particular when combined with the PLDA model [14] for scoring.

In spite of the great achievement, a large performance degradation is often observed when current speaker recognition systems are deployed to real applications. A particular problem is the domain mismatch, including the training-deployment domain mismatch and enrollment-test domain mismatch. For example, when the enrollment uses one recording device, and the test uses another device, the performance will be seriously degraded.

A large body of research has been conducted to solve the domain mismatch problem, and the basic idea among these approaches is either normalization or adaptation: the former aims to make the data or model domain-independent [15, 16, 17, 18, 19], while the latter aims to make them more suitable for the target domain [20, 21, 22]. Among all these approaches, training a multi-domain PLDA is the mostly used. This approach collects data from multiple domains and then train the PLDA model, with the hope to learn a domain-independent speaker subspace. This PLDA multi-domain training (MDT) is in particular useful when the training data is too limited to conduct more data-hungry approach, e.g., the domain-invariance feature learning [16, 19]. A key advantage of this MDT approach is that the resultant PLDA model will be suitable for both training-deployment mismatch and enrollment-test mismatch.

Domain-adaption training (DAT) is another approach, which transforms speaker vectors from the source domain to the target domain, and then performs the scoring process in the target domain. The transform can be established by maximizing likelihood of the projected data in the source domain under the PLDA model.

For both MDT and DAT, cross-domain speakers are essentially important. The cross-domain labels provide information regarding the variation related to domains. A common wisdom is to collect as much cross-domain data as possible, with the hope to learn a *true* speaker subspace that is independent of domain change. However, our experimental study shows that this is not true: more cross-domain data does not necessarily lead to better performance in conditions with enroll-test mismatch. We give a careful analysis on this phenomenon, and find that it is related to the statistics incoherence problem caused by domain mismatch. Based on this analysis, we propose a decoupled scoring approach. The key idea is to decouple the scoring process into different phases and utilize its own correct statistics in each phase. By this model, the cross-domain labels are only used to establish the link between the *statistics of different domains*.

Thanks to the National Natural Science Foundation of China No. 61633013 for funding. Dong Wang is the corresponding author (wangdong99@mails.tsinghua.edu.cn).

The rest of the paper is organized as follows. Section 2 presents experimental setups, and Section 3 gives an empirical study on the statistics incoherence problem. Section 4 presets details of our proposed decoupled scoring approach. Section 5 describes experimental results, and the entire paper is concluded in Section refsec:con.

2. EXPERIMENTAL SETUPS

2.1. Data

Two datasets were used in our experiments: VoxCeleb [5, 23] and AIShell-1 [24]. VoxCeleb was used to train the speaker embedding model, which is the x-vector model in our experiment. AIShell-1 was used for performance evaluation under cross-channel domain mismatch scenarios. More information about datasets is presented below.

VoxCeleb[1]: The entire dataset contains $2,000$+ hours of speech signals from $7,000$+ speakers. Data augmentation was applied to improve robustness, with the MUSAN corpus used to generate noisy utterances, and the room impulse responses (RIRS) corpus was used to generate reverberant utterances.

AIShell-1[2]: This is an open-source multi-channel Chinese Mandarin speech dataset published by AISHELL. All the speech utterances are recorded in parallel via three categories of devices, including high fidelity Microphones (Mic), Android phones (AND) and Apple iPhones (iOS). This dataset is used for the cross-channel (domain) test in our experiment. The entire dataset consists of two parts: *AIShell-1.Train*, which covers 3 devices and involves $360,897$ utterances from 340 speakers, was used to implement both the ad-hoc normalization/adaptation methods and our proposed decoupled scoring method. *AIShell-1.Eval*, which also covers 3 devices and involves $64,495$ utterances from 60 speakers, was used for performance evaluation on different cross-channel conditions.

2.2. Embedding models

We built the state-of-the-art x-vector embedding model based on the TDNN architecture [3]. This x-vector model was created using the Kaldi toolkit [25], following the VoxCeleb recipe. The acoustic features are 40-dimensional Fbanks. The main architecture contains three components. The first component is the feature-learning component, which involves 5 time-delay (TD) layers to learn frame-level speaker features. The splicing parameters for these 5 TD layers are: $\{t\text{-}2, t\text{-}1, t, t\text{+}1, t\text{+}2\}$, $\{t\text{-}2, t, t\text{+}2\}$, $\{t\text{-}3, t, t\text{+}3\}$, $\{t\}$, $\{t\}$. The second component is the statistical pooling component, which computes the mean and standard deviation of the frame-level features from a speech segment. The final one is the speaker-classification component, which discriminates different speakers. This component has 2 full-connection (FC) layers and the size of its output is $7,185$, corresponding to the number of speakers in the training set. Once trained, the 512-dimensional activations of the penultimate FC layer are read out as an x-vector.

[1] http://www.robots.ox.ac.uk/ vgg/data/voxceleb/

[2] http://openslr.org/33/

3. EMPIRICAL ANALYSIS

We firstly employ the MDT approach to train a domain-independent PLDA. In detail, data of speakers collected from two domains are labeled in two ways. One is the domain-independent labeling, where data from different domains share the same speaker label. The other one is the domain-dependent labeling, where data from different domains are relabeled as different speakers even if they are from the same speaker. The PLDA model will be trained in a controlled experiment, where we control the proportion of domain-independent labels and domain-dependent labels. The expectation was that more domain-independent labels will lead to better performance at lest in cross-domain test.

In our experiments, the MDT is conducted using *AIShell-1.Dev*, and the resultant PLDA is evaluated on *AIShell-1.Eval*. The results are shown in Table 1. Note that in the first column, the case name 'A-B' means enroll on condition A and test on condition B. The 'Base' column presents results using PLDA trained with Voxceleb. A key observation is that the best performance is obtained when the PLDA model is trained with 40%-60% domain-independent labels plus 60%-40% domain-dependent labels, rather with 100% domain-independent labels.

Table 1. EER(%) results with MDT under different cross-channel labels.

Cases	Base	Proportion of domain-independent labels					
		0%	20%	40%	60%	80%	100%
AND-AND	0.797	-	-	-	-	-	-
AND-Mic	2.146	3.329	1.410	1.273	**1.066**	1.259	1.151
AND-iOS	1.425	1.642	1.104	**0.930**	1.029	1.170	1.161
Mic-Mic	2.175	3.675	1.953	1.746	1.184	1.307	**1.161**
Mic-AND	0.778	-	-	-	-	-	-
Mic-iOS	2.251	3.732	1.883	1.675	**1.255**	1.349	1.293
iOS-iOS	1.599	2.024	1.472	1.241	**1.156**	1.274	**1.156**
iOS-AND	2.216	3.697	1.651	1.476	**1.061**	1.236	1.137
iOS-Mic	0.920	-	-	-	-	-	-

This somewhat striking result can be intuitively explained by the statistical nature of the PLDA model. Basically, PLDA conducts inference based on the between-class and within-class distributions that are learned from data, and if the distributions match the data perfectly, PLDA will derive optimal scores in terms of Bayes risk. Unfortunately, MDT learn distributions of the pool data, which are not ideal for any domain. For instance, the within-class variance tends to be large by MDT even if it is small in each of the single domain, which

may lead to unreliable scores. This analysis is just intuitive; more convincing explanation requires more theoretical study, as we will present in the next section.

4. DECOUPLED SCORING

We present a theoretical analysis and solution for the domain mismatch problem sampled in the previous section. This analysis and solution are based on the normalized likelihood (NL) scoring framework that we proposed recently [26].

4.1. Revisit NL scoring

We shall assume a simple linear Gaussian as follows:

$$p(\boldsymbol{\mu}) = N(\boldsymbol{\mu}; \mathbf{0}, \epsilon\mathbf{I}) \tag{1}$$

$$p(\boldsymbol{x}|\boldsymbol{\mu}) = N(\boldsymbol{x}; \boldsymbol{\mu}, \sigma\mathbf{I}), \tag{2}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{x}$ represent speaker means and speaker vectors, respectively; ϵ and σ are the between-speaker variance and the within-speaker variance, respectively.

With this model, it is easy to derive the marginal probability $p(\boldsymbol{x})$ and the posterior probability $p(\boldsymbol{\mu}|\boldsymbol{x})$ as follows:

$$p(\boldsymbol{x}) = N(\boldsymbol{x}; \mathbf{0}, (\epsilon + \sigma)\mathbf{I}) \tag{3}$$

$$p(\boldsymbol{\mu}|\boldsymbol{x}) = N(\boldsymbol{\mu}; \frac{\epsilon}{\epsilon+\sigma}\boldsymbol{x}, \frac{\epsilon\sigma}{\epsilon+\sigma}\mathbf{I}). \tag{4}$$

If the observations are more than one, the posterior probability is:

$$p(\boldsymbol{\mu}|\boldsymbol{x}_1, ..., \boldsymbol{x}_n) = N(\boldsymbol{\mu}; \frac{n\epsilon}{n\epsilon+\sigma}\bar{\boldsymbol{x}}, \frac{\epsilon\sigma}{n\epsilon+\sigma}\mathbf{I}), \tag{5}$$

where $\bar{\boldsymbol{x}}$ is the average of the observations $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$.

The task of speaker verification is to test the following two hypotheses regarding to a speaker vector $\boldsymbol{x}$ and check which one is more probable: $\{H_0$: $\boldsymbol{x}$ belongs to class k; H_1: $\boldsymbol{x}$ belongs to any class other than k $\}$. We therefore define the normalized likelihood (NL) as:

$$NL(\boldsymbol{x}|k) = \frac{p(\boldsymbol{x}|H_0)}{p(\boldsymbol{x}|H_1)} = \frac{p_k(\boldsymbol{x})}{p(\boldsymbol{x})}. \tag{6}$$

where $p(\boldsymbol{x}|H_0)$ is the likelihood of $\boldsymbol{x}$ generated by class k, denoted by $p_k(\boldsymbol{x})$. The posterior for H_0 is essentially the likelihood $p_k(\boldsymbol{x})$ normalized by the evidence $p(\boldsymbol{x})$.

The likelihood $p_k(\boldsymbol{x})$ can be computed by marginalizing over $\boldsymbol{\mu}_k$ according to Eq.(5), where $\boldsymbol{\mu}_k$ can be estimated from the enrollment samples $\{\boldsymbol{x}_1^k, ..., \boldsymbol{x}_n^k\}$. Following Eq.(3), we have:

$$\begin{aligned} p_k(\boldsymbol{x}) &= p(\boldsymbol{x}|\boldsymbol{x}_1^k, ..., \boldsymbol{x}_n^k) \\ &= \int p(\boldsymbol{x}|\boldsymbol{\mu}_k)p(\boldsymbol{\mu}_k|\boldsymbol{x}_1^k, ..., \boldsymbol{x}_n^k)\mathrm{d}\boldsymbol{\mu}_\mathrm{k} \\ &= N(\boldsymbol{x}; \frac{n_k\epsilon}{n_k\epsilon+\sigma}\bar{\boldsymbol{x}}_k, (\sigma + \frac{\epsilon\sigma}{n_k\epsilon+\sigma})\mathbf{I}). \end{aligned} \tag{7}$$

A simple computation shows that:

$$\begin{aligned} \log NL(\boldsymbol{x}|k) &= \log p_k(\boldsymbol{x}) - \log p(\boldsymbol{x}) \\ &\propto -\frac{1}{\sigma + \frac{\epsilon\sigma}{n_k\epsilon+\sigma}}||\boldsymbol{x} - \tilde{\boldsymbol{\mu}}_k||^2 + \frac{1}{\epsilon+\sigma}||\boldsymbol{x}||^2, \end{aligned} \tag{8}$$

where we have defined:

$$\tilde{\boldsymbol{\mu}}_k = \frac{n_k\epsilon}{n_k\epsilon+\sigma}\bar{\boldsymbol{x}}_k. \tag{9}$$

4.2. Three-phase perspective

A simple arrangement on the NL score shows that:

$$NL = \frac{p(\boldsymbol{x}|\boldsymbol{x}_1, ..., \boldsymbol{x}_n)}{p(\boldsymbol{x})} = \frac{\int p(\boldsymbol{x}|\boldsymbol{\mu})p(\boldsymbol{\mu}|\boldsymbol{x}_1, ..., \boldsymbol{x}_n)\mathrm{d}\boldsymbol{\mu}}{p(\boldsymbol{x})}. \tag{10}$$

By this NL form, the score is computed based on three phases:

- The enrollment phase $p(\boldsymbol{\mu}|\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$ that produces the posterior of the class mean $\boldsymbol{\mu}$.
- The prediction phase $p(\boldsymbol{x}|\boldsymbol{\mu})$ that computes the probability of a test sample belonging to a class represented by the class mean $\boldsymbol{\mu}$.
- The normalization phase $p(\boldsymbol{x})$ that computes the probability that $\boldsymbol{x}$ is produced by all potential classes.

Usually these three phases are based on the same statistical model, under the assumption that the statistics of the enrollment and test conditions are the same. In this case, the NL score is mathematically equal to PLDA [26]. In conditions where the enroll and test are in different domains, the statistics of the enrollment and test conditions are different, and so the three phases should use different statistics, in order to obtain an optimal score. This phenomenon is called statistics coherence (between scoring phases).

The concept of three-phase scoring provides a clear explanation of the strange behavior we observed in Section 4. Firstly notice that although domain-independent labels help learn a better between-class distribution, it also destroys the within-class distribution, by making it unnecessarily large. In contrast, domain-dependent labels help learn the within-class distribution, but is less useful for learning the between-class distribution (this is particularly the case if we assume that

the cross-channel effect is mostly a feature shift). Moreover, note that a domain-robust between-class distribution is only requested in the enrollment phase[3], however a good within-class distribution is requested for all the three phases. Therefore, with a limited amount of data, we need balance the contribution to the between-class and within-class distributions, and so have to reserve a proportion of domain-dependent labels.

4.3. Decoupled scoring

Following the three-phase perspective, a principle solution to domain mismatch is to use the respective statistical model for each phase, which we call **domain statistics decomposition (DSD)**. In this paper, we propose a simple yet effective DSD implementation based on a linear transform. Firstly, for the enrollment phase, the statistical model of the enrollment condition $\{\epsilon\mathbf{I}, \sigma\mathbf{I}\}$ should be used. Secondly, for the normalization phase, the statistical model of the test condition $\{\hat{\epsilon}\mathbf{I}, \hat{\sigma}\mathbf{I}\}$ should be used. Finally, for the prediction phase, a transformation is applied to connect the conditional probability $p(\boldsymbol{x}|\boldsymbol{\mu})$ and the posterior $p(\boldsymbol{\mu}|\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$. For simplicity, we assume this transformation is linear:

$$\boldsymbol{x} = \mathbf{M}\hat{\boldsymbol{x}} + \boldsymbol{b}, \tag{11}$$

where $\hat{\boldsymbol{x}}$ represents the observation in the test condition, and $\boldsymbol{x}$ is the transformed data in the enrollment condition. If we assume that the transformed data can be well represented by the statistical model of the enrollment condition, the NL score can be derived.

Firstly, the posterior $p(\boldsymbol{\mu}_k|\boldsymbol{x}_1^k, ..., \boldsymbol{x}_n^k)$ is computed using the model of the enrollment condition:

$$p(\boldsymbol{\mu}_k|\boldsymbol{x}_1^k, ..., \boldsymbol{x}_n^k) = N(\boldsymbol{\mu}_k; \frac{n_k\epsilon}{n_k\epsilon + \sigma}\bar{\boldsymbol{x}}_k, \frac{\epsilon\sigma}{n_k\epsilon + \sigma}\mathbf{I}). \tag{12}$$

Secondly, transform the test sample $\hat{\boldsymbol{x}}$ by the linear transformation, and perform the prediction using the model of the enrollment condition:

$$\begin{aligned} p_k(\hat{\boldsymbol{x}}; \mathbf{M}, \boldsymbol{b}) &= p(\mathbf{M}\hat{\boldsymbol{x}} + \boldsymbol{b}|\boldsymbol{x}_1^k, ..., \boldsymbol{x}_n^k) \\ &= \int p(\mathbf{M}\hat{\boldsymbol{x}} + \boldsymbol{b}|\boldsymbol{\mu}_k)p(\boldsymbol{\mu}_k|\boldsymbol{x}_1^k, ..., \boldsymbol{x}_n^k)\mathrm{d}\boldsymbol{\mu}_\mathrm{k} \\ &= N(\mathbf{M}\hat{\boldsymbol{x}} + \boldsymbol{b}; \frac{n_k\epsilon}{n_k\epsilon + \sigma}\bar{\boldsymbol{x}}_k, (\sigma + \frac{\epsilon\sigma}{n_k\epsilon + \sigma})\mathbf{I}). \end{aligned} \tag{13}$$

According to Eq.(3), the normalization term $p(\hat{\boldsymbol{x}})$ is computed based on the model of the test condition:

$$p(\hat{\boldsymbol{x}}) = N(\hat{\boldsymbol{x}}; \mathbf{0}, (\hat{\epsilon} + \hat{\sigma})\mathbf{I}). \tag{14}$$

The NL score is then computed as follows:

$$\log NL(\hat{\boldsymbol{x}}|k) \propto -\frac{1}{\sigma + \frac{\epsilon\sigma}{n_k\epsilon+\sigma}}||\mathbf{M}\hat{\boldsymbol{x}} + \boldsymbol{b} - \tilde{\boldsymbol{\mu}}_k||^2 + \frac{1}{\hat{\epsilon} + \hat{\sigma}}||\hat{\boldsymbol{x}}||^2. \tag{15}$$

The optimal parameters $\{\mathbf{M}, \boldsymbol{b}\}$ for the linear transformation can be estimated by maximum likelihood (ML) training, which maximizes $p_k(\hat{\boldsymbol{x}}; \mathbf{M}, \mathbf{b})$ with respect to $\mathbf{M}$ and $\mathbf{b}$, using samples in the test condition and the samples from the same speaker in the enrollment condition. The objective function for the optimization can be written by:

$$\mathcal{L}(\mathbf{M}, \boldsymbol{b}) = \sum_{k=1}^{K}\sum_{i=1}^{N} p_k(\hat{\boldsymbol{x}}_i; \mathbf{M}, \boldsymbol{b}), \tag{16}$$

where K denotes the number of speakers, and N denotes the number of test samples in each speaker.

5. EXPERIMENTAL RESULTS

5.1. Methods

- **DSD**: Domain statistics decomposition with linear transformation. This model is a linear transformation $\boldsymbol{x} = \mathbf{M}\hat{\boldsymbol{x}} + \boldsymbol{b}$ that maps the test sample $\hat{\boldsymbol{x}}$ to the enrollment condition so that $\hat{\boldsymbol{x}}$ can be represented by the statistical model of the enrollment condition. Importantly, the transformation is only used to compute the likelihood term $p_k(\boldsymbol{x})$; and the normalization term $p(\boldsymbol{x})$ is still computed using the statistical model of the test condition, based on the original test data $\hat{\boldsymbol{x}}$. We implemented it based on the MLE criterion with the training objective shown in Eq.(16). In our experiment, the Adam optimizer [27] was used to optimize the parameters $\{\mathbf{M}, \boldsymbol{b}\}$.

- **MDT**: Multi-domain training (MDT). A commonly adopted domain normalization approach. In this method, data from both the enrollment and test conditions are pooled together, and then are used to retrain the PLDA model.

- **DAT**: Domain-adaptation training (DAT). A simple domain adaptation approach that transforms speaker vectors from the source domain to the target domain and then performs the scoring in the target domain. For better comparison, the transformation is also linear and the same as in DSD, defined by $\boldsymbol{x} = \mathbf{M}\hat{\boldsymbol{x}} + \boldsymbol{b}$, and the parameters $\{\mathbf{M}, \boldsymbol{b}\}$ are estimated using the same ML training as the DSD approach.

[3]For the prediction phase, there is no between-class distribution required; and for the normalization phase, we need a between-class distribution that *matches* the within-class distribution in order to represent the marginal distribution of the test data.

5.2. Basic results

The basic result with these three methods are reported in Table 2. It can be observed that the proposed DSD approach consistently outperforms MDT and DAT, demonstrating that DSD is a more effective approach in dealing with domain mismatch. The comparison between DSD and DAT is especially interesting, as the two methods look very similar and only differ in the normalization term $p(\boldsymbol{x})$. The clear advantage of DSD demonstrated the importance of a principle solution based on a solid theory.

Table 2. EER(%) results with three methods on the cross-channel test.

Cases	Base	Methods		
		MDT	DAT	DSD
AND-AND	0.797	-	-	-
AND-Mic	2.146	1.151	1.245	**0.981**
AND-iOS	1.425	1.161	1.312	**0.623**
Mic-AND	2.175	1.161	1.189	**0.712**
Mic-Mic	0.778	-	-	-
Mic-iOS	2.251	1.293	1.481	**0.812**
iOS-AND	1.599	1.156	1.184	**0.755**
iOS-Mic	2.216	1.137	1.231	**1.052**
iOS-iOS	0.920	-	-	-

5.3. Further analysis

We perform a further analysis to better understand the properties of these three methods. The number of speakers are sampled from *AIShell-1.Dev* ranging from 68 to 340, and then we test the performance of MDT, DAT and DSD methods with different amounts of cross-domain data, and investigate which method has the best capability to squeeze the cross-domain data. The results are reported in Table 3.

It can be observed that with the increasing of speaker numbers, the performance of DSD is consistently improved and gradually outperforms both MDT and DAT. We attribute its success to the decoupled scoring scheme. In this scheme, $p_k(\boldsymbol{x})$ and $p(\boldsymbol{x})$ are individually computed. The $p_k(\boldsymbol{x})$ is optimized based on the statistics in the enrollment conditions, while the $p(\boldsymbol{x})$ is estimated based on the statistics in the test conditions. Therefore, the decoupled NL score has more capability to squeeze the cross-domain data. In a word, compared to MDT and DAT, DSD can obtain an optimal NL score, and can be regarded as a principle solution to domain mismatch.

6. CONCLUSIONS

This paper investigated the issue of domain mismatch in speaker recognition, and found that the statistics incoherence is the essential problem associated with this mismatch. To deal with this problem, we presented a decoupled scoring approach. Specifically, we decouple the scoring process to three separated phases according to the normalized likelihood (NL) framework, and statistics from different conditions are used in different phases. Besides, a simple linear transformation is applied to implement this decoupling scoring approach. Experimental results demonstrated that the proposed decoupling scoring approach is highly effective to squeeze the value of cross-domain data and obtains the best performance among all the competitive methods. Future work will extend this approach to many other mismatch scenarios, e.g., dynamic speaker enrollment, multi-genre test.

Table 3. EER(%) results with three methods under different numbers of cross-domain data.

Methods	Cases	# of speakers				
		68	136	204	272	340
MDT	AND-Mic	5.093	1.896	1.264	1.283	1.151
	AND-iOS	5.185	1.628	1.250	1.250	1.161
	Mic-AND	5.586	2.284	1.274	1.194	1.161
	Mic-iOS	5.732	2.213	1.491	1.420	1.293
	iOS-AND	5.213	1.807	1.236	1.151	1.156
	iOS-Mic	5.296	1.900	1.165	1.165	1.137
DAT	AND-Mic	1.174	1.137	1.245	1.287	1.245
	AND-iOS	1.184	1.175	1.274	1.307	1.312
	Mic-AND	1.458	1.364	1.189	1.227	1.189
	Mic-iOS	1.548	1.434	1.420	1.496	1.481
	iOS-AND	1.222	1.123	1.146	1.208	1.184
	iOS-Mic	1.264	1.127	1.151	1.250	1.236
DSD	AND-Mic	2.976	2.108	0.863	1.080	0.981
	AND-iOS	1.241	1.080	0.528	0.646	0.623
	Mic-AND	5.926	5.086	0.967	0.750	0.712
	Mic-iOS	5.577	4.902	1.062	0.830	0.812
	iOS-AND	1.840	1.632	0.646	0.816	0.755
	iOS-Mic	2.735	1.882	0.816	1.094	1.052

7. REFERENCES

[1] John HL Hansen and Taufiq Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[2] 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, 2011.

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings

for speaker recognition," in *ICASSP*. IEEE, 2018, pp. 5329–5333.

[4] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, "Attentive statistics pooling for deep speaker embedding," in *INTERSPEECH*, 2018, pp. 2252–2256.

[5] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "VoxCeleb2: Deep speaker recognition," in *INTERSPEECH*, 2018, pp. 1086–1090.

[6] Jee weon Jung, Hee-Soo Heo, Ju ho Kim, Hye jin Shim, and Ha-Jin Yu, "RawNet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," in *INTERSPEECH*, 2019, pp. 1268–1272.

[7] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *ICASSP*, 2019, pp. 5791–579.

[8] Nanxin Chen, Jess Villalba, and Najim Dehak, "Tied mixture of factor analyzers layer to combine frame level representations in neural speaker embeddings," in *INTERSPEECH*, 2019, pp. 2948–2952.

[9] Wenhao Ding and Liang He, "MTGAN: Speaker verification through multitasking triplet generative adversarial networks," in *INTERSPEECH*, 2018, pp. 3633–3637.

[10] Jixuan Wang, Kuan-Chieh Wang, Marc T Law, Frank Rudzicz, and Michael Brudno, "Centroid-based deep metric learning for speaker recognition," in *ICASSP*. IEEE, 2019, pp. 3652–3656.

[11] Zhongxin Bai, Xiao-Lei Zhang, and Jingdong Chen, "Partial AUC optimization based deep speaker embeddings with class-center learning for text-independent speaker verification," in *ICASSP*. IEEE, 2020, pp. 6819–6823.

[12] Zhifu Gao, Yan Song, Ian McLoughlin, Pengcheng Li, Yiheng Jiang, and Li-Rong Dai, "Improving aggregation and loss function for better embedding learning in end-to-end speaker verification system," in *INTERSPEECH*, 2019, pp. 361–365.

[13] Seyed Omid Sadjadi, Craig Greenberg, Elliot Singer, Douglas Reynolds, Lisa Mason, and Jaime Hernandez-Cordero, "The 2018 NIST speaker recognition evaluation," in *INTERSPEECH*, 2019, pp. 1483–1487.

[14] Sergey Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision (ECCV)*. Springer, 2006, pp. 531–542.

[15] Md Hafizur Rahman, Ahilan Kanagasundaram, Ivan Himawan, David Dean, and Sridha Sridharan, "Improving PLDA speaker verification performance using domain mismatch compensation techniques," *Computer Speech & Language*, vol. 47, pp. 240–258, 2018.

[16] Qing Wang, Wei Rao, Sining Sun, Lei Xie, Eng Siong Chng, and Haizhou Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *ICASSP*, 2018, pp. 4889–4893.

[17] Jennifer Williams and Simon King, "Disentangling style factors from speaker representations," in *INTERSPEECH*, 2019, pp. 3945–3949.

[18] Gautam Bhattacharya, Jahangir Alam, and Patrick Kenny, "Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training," in *ICASSP*. IEEE, 2019, pp. 6041–6045.

[19] Woo Hyun Kang, Sung Hwan Mun, Min Hyun Han, and Nam Soo Kim, "Disentangled speaker and nuisance attribute embedding for robust speaker verification," *IEEE Access*, 2020.

[20] Suwon Shon, Seongkyu Mun, Wooil Kim, and Hanseok Ko, "Autoencoder based domain adaptation for speaker recognition under insufficient channel information," in *INTERSPEECH*, 2017, pp. 1014–1018.

[21] Chunlei Zhang, Shivesh Ranjan, and John HL Hansen, "An analysis of transfer learning for domain mismatched text-independent speaker verification," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2018, pp. 181–186.

[22] Kong Aik Lee, Qiongqiong Wang, and Takafumi Koshinaka, "The CORAL+ algorithm for unsupervised domain adaptation of plda," in *ICASSP*. IEEE, 2019, pp. 5821–5825.

[23] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "VoxCeleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[24] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *O-COCOSDA*. IEEE, 2017, pp. 1–5.

[25] 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 EPFL-CONF-192584.

[26] Dong Wang, "Remarks on optimal scores for speaker recognition," 2020.

[27] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.