INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Adversarial Separation and Adaptation Network for Far-Field Speaker Verification

Lu Yi, Man Wai Mak

Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China

lu-louisa.yi@connect.polyu.hk, enmwmak@polyu.edu.hk

Abstract

Typically, speaker verification systems are highly optimized on the speech collected by close-talking microphones. However, these systems will perform poorly when the users use far-field microphones during verification. In this paper, we propose an adversarial separation and adaptation network (ADSAN) to extract speaker discriminative and domain-invariant features through adversarial learning. The idea is based on the notion that speaker embedding comprises domain-specific components and domain-shared components, and that the two components can be disentangled by the interplay of the separation network and the adaptation network in the ADSAN. We also propose to incorporate a mutual information neural estimator into the domain adaptation network to retain speaker discriminative information. Experiments on the VOiCES Challenge 2019 demonstrate that the proposed approaches can produce more domain-invariant and speaker discriminative representations, which could help to reduce the domain shift caused by different types of microphones and reverberant environments.

Index Terms: Far field speaker verification; domain adaptation; adversarial learning; domain mismatch

1. Introduction

Today's speaker recognition systems have achieved remarkable performance under controlled environments such as quiet offices and clean telephone channels. However, far-field speaker recognition is still challenging because the environmental noise and reverberation effect are hard to control. Due to the difference in microphone characteristics, there is a domain mismatch between near-field microphone speech and far-field microphone speech. The mismatch can make a speaker recognition system that is trained on near-field microphone speech to perform poorly on far-field microphone speech. Domain adaptation (DA) can be applied to address this problem.

Domain adaptation algorithms for speaker verification (SV) either adapt the probabilistic linear discriminant analysis (PLDA) models to fit the target data [1] or find a common embedding space in which the feature distributions have low discrepancy across multiple domains. For the former, a recent approach is to use the concept of correlation alignment [2] to estimate a pseudo-in-domain covariance matrix. The matrix is then interpolated with the out-of-domain covariance matrix of the PLDA model [3]. Finding a common embedding space is a more general approach in that it is independent of the backend. Research in this direction has focused on compensating for the inter-data set variability [4], projecting out the inter-dataset variability [5], and normalizing the covariance of in- and out-domain data [6]. Huang and Bocklet [7] applied invariant representation learning to find noise robust speaker representations. The idea is to minimize the cosine distance and mean squared error at the embedding layer of the x-vector network across clean and noisy utterances. In [8], the maximum mean discrepancy (MMD) was added to the objective function of an autoencoder. After training, domain-invariant features were extracted from the middle layer of the autoencoder.

More recent research on domain adaptation is based on adversarial learning. Wang *et al.* [9] applied domain adversarial training (DAT) to learn speaker discriminative and domain-invariant representations. The proposed approach outperforms other traditional unsupervised domain adaptation techniques on the 2013 Domain Adaptation Challenge. Tu *et al.* [10] imposed adversarial learning on a variational autoencoder to extract domain-invariant and Gaussian-like speaker features. The Gaussianized feature vectors meet the Gaussianity requirement of the PLDA backend, which helps to improve performance. In [11], adversarial domain adaptation was utilized to reduce language mismatch. Nidadavolu *et al.* [12] investigated the effectiveness of cycle-consistent generative adversarial networks (CycleGAN) when a limited amount of target domain data are available. Their experiment on far-field microphones reveals that this unsupervised domain adaptation technique can help to reduce the mismatch between the reverberant speech and the clean speech.

Many studies in domain adaptation focused on finding a common feature space for all domains [13, 14]. They assumed that the speaker representation contains both domain-specific components and domain-invariant components and attempted to disentangle the two types of components. In the context of transfer learning, a negative transfer occurs if the learned common features contain many domain-specific properties. Domain separation networks (DSNs) [13] can be used to alleviate this problem. Through the use of autoencoders and divergence measures, a DSN can capture the common representation shared by different domains. The domain-specific representations can be obtained by finding subspaces that are orthogonal to the common representation. It has been shown that DSNs can help extract domain-invariant features in image recognition tasks [13] and speech recognition tasks [14].

In this work, we propose applying the idea of domain separation networks for speaker verification. Domain-invariant speaker embeddings can be obtained by disentangling the shared properties of the source and target domains from the domain-specific properties. Similar to DSNs, our proposed network aims to obtain a domain-invariant representation through a shared encoder and acquire domain-specific representations through domain-dependent encoders. Instead of forcing the domain-specific and domain-invariant embeddings to be or-

This work was in part supported by the National Natural Science Foundation of China (NSFC), Grant No. 61971371 and Huawei, Contract No. YBN2019095008.

http://dx.doi.org/10.21437/Interspeech.2020-2372

thogonal to each other, a discriminator is used to maximize the differences between the shared properties and domain-specific properties. It is assumed that the representations are different if they can always be classified correctly by an optimal discriminator.

One potential risk of DSNs is that the features produced by the shared encoder may not contain enough task-related information (in our case, the speaker information), especially when the decoder is flexible enough [15, 16]. Therefore, we further propose to maximize the mutual information between the input and output of the shared encoder to retain as much speaker information as possible in the common features ($\mathbf{z}_h^s$ and $\mathbf{z}_h^t$ in Fig. 1). Using labelled closed-talking utterances and unlabelled far-field utterances, the proposed network can be trained to reduce the domain mismatch caused by different types of microphones. The resulting network can improve the performance of speaker verification systems that are trained on near-field microphone speech but evaluated on far-field microphone speech.

2. Adversarial Separation and Adaptation Network

2.1. Network Structure and Loss Functions

Let $\mathbf{X}^s = \left\{\left(\mathbf{x}_i^s, \mathbf{y}_{\text{spk}}^i\right)\right\}_{i=1}^{N_s}$ represents a labelled dataset of N_s samples from the source domain $\mathcal{D}_s$ and $\mathbf{X}^t = \left\{\mathbf{x}_i^t\right\}_{i=1}^{N_t}$ represents an unlabelled dataset of N_t samples from the target domain $\mathcal{D}_t$. Apart from applying the original DSN proposed by Bousmlis *et al.* [13], we further propose a variant of DSN called adversarial separation and adaptation network (ADSAN), as Fig. 1 shows.

The proposed network comprises three encoders, three discriminators, and a decoder. The shared encoder G_h is trained adversarially to extract the common features $\{\mathbf{z}_h^s, \mathbf{z}_h^t\}$ from both source and target domains. The speaker discriminator is applied to make the common features to be speaker discriminative. Because only the source data have speaker labels, the speaker discriminator is trained to minimize the classification loss on the source domain. The domain-dependent encoders, G_s and G_t, are trained to extract domain-specific features $\{\mathbf{z}_s^s, \mathbf{z}_t^t\}$, which are assumed to comprise as much domain-specific information as possible. The shared decoder $R\left(\mathbf{z}_c\right)$ is trained to reconstruct the original features ($\mathbf{x}^s$ or $\mathbf{x}^t$) from the concatenation of the domain-specific features and shared features ($(\mathbf{z}_s^s, \mathbf{z}_h^s)$ or $(\mathbf{z}_t^t, \mathbf{z}_h^t)$). The reconstruction loss backpropagated to the source and target encoders can force the extracted features to maintain the information of the original data. To ensure that the common features can be disentangled from the domain-dependent information, the separation discriminator D_{sep} is applied to discriminate the vectors $\mathbf{z}_s^s, \mathbf{z}_h^s, \mathbf{z}_h^t$ and $\mathbf{z}_t^t$ into three groups: *shared, source*, and *target*. The adaptation discriminator D_{adt} is utilized to distinguish whether the shared features are from the source domain or from the target domain. Meanwhile, the shared encoder tries to produce shared features that make the adaptation discriminator into believing that there is no difference between $\mathbf{z}_h^s$ and $\mathbf{z}_h^t$. As a result, domain-invariant and speaker discriminative features can be extracted from the output of the shared encoder, which are expected to improve the speaker verification performance on the target domain.

To achieve the objectives mentioned above, we need to reduce the classification error on the source domain by minimizing the cross-entropy loss:

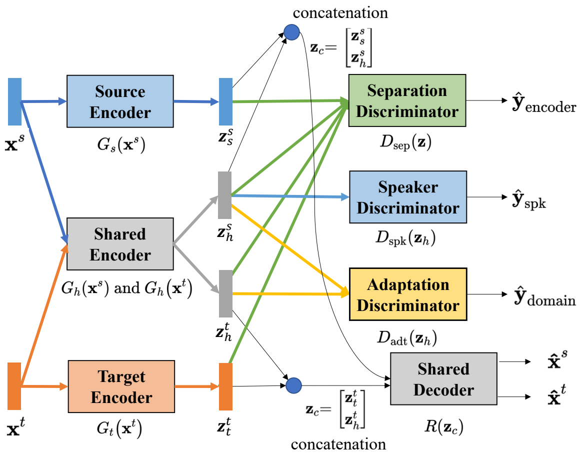
concatenation
$\mathbf{z}_c = \begin{bmatrix}\mathbf{z}_s^s \\ \mathbf{z}_h^s\end{bmatrix}$
Source Encoder
$G_s(\mathbf{x}^s)$
$\mathbf{x}^s$
$\mathbf{z}_s^s$
Separation Discriminator
$D_{\text{sep}}(\mathbf{z})$
$\hat{\mathbf{y}}_{\text{encoder}}$
Speaker Discriminator
$D_{\text{spk}}(\mathbf{z}_h)$
$\hat{\mathbf{y}}_{\text{spk}}$
Shared Encoder
$G_h(\mathbf{x}^s)$ and $G_h(\mathbf{x}^t)$
$\mathbf{z}_h^s$
$\mathbf{z}_h^t$
Adaptation Discriminator
$D_{\text{adt}}(\mathbf{z}_h)$
$\hat{\mathbf{y}}_{\text{domain}}$
Target Encoder
$G_t(\mathbf{x}^t)$
$\mathbf{x}^t$
$\mathbf{z}_t^t$
Shared Decoder
$R(\mathbf{z}_c)$
$\hat{\mathbf{x}}^s$
$\hat{\mathbf{x}}^t$
$\mathbf{z}_c = \begin{bmatrix}\mathbf{z}_t^t \\ \mathbf{z}_h^t\end{bmatrix}$
concatenation

Figure 1: *The architecture of the adversarial separation and adaptation network.*

$$\mathcal{L}_{\text{task}} = \mathbb{E}_{\mathbf{x}^s \sim \mathcal{D}_s}\left[-\sum_{k=1}^{K} y_{\text{spk}}^{(k)} \log D_{\text{spk}}\left(G_h\left(\mathbf{x}^s\right)\right)_k\right], \quad (1)$$

where the subscript k denotes the k-th output of the speaker discriminator.

The shared decoder is trained to minimize the mean squared error:

$$\begin{aligned}\mathcal{L}_{\text{recon}} = \mathbb{E}_{\mathbf{x}^s \sim \mathcal{D}_s}&\left\|\mathbf{x}^s - R\left(\left[G_s\left(\mathbf{x}^s\right)^\mathsf{T}, G_h\left(\mathbf{x}^s\right)^\mathsf{T}\right]^\mathsf{T}\right)\right\|_2^2 \\ + \mathbb{E}_{\mathbf{x}^t \sim \mathcal{D}_t}&\left\|\mathbf{x}^t - R\left(\left[G_t\left(\mathbf{x}^t\right)^\mathsf{T}, G_h\left(\mathbf{x}^t\right)^\mathsf{T}\right]^\mathsf{T}\right)\right\|_2^2. \quad (2)\end{aligned}$$

The separation discriminator has three outputs, which correspond to $\hat{y}_{\text{sep}}^{(1)} = P(\text{shared}|\mathbf{z})$, $\hat{y}_{\text{sep}}^{(2)} = P(\text{source}|\mathbf{z})$, and $\hat{y}_{\text{sep}}^{(3)} = P(\text{target}|\mathbf{z})$, respectively. Therefore, it is trained to minimize the following cross-entropy loss:

$$\begin{aligned}\mathcal{L}_{\text{sep}} = \mathbb{E}_{\mathbf{x}^s \sim \mathcal{D}_s}&\left[-\log D_{\text{sep}}\left(G_s(\mathbf{x}^s)\right)_2 - \log D_{\text{sep}}\left(G_h\left(\mathbf{x}^s\right)\right)_1\right] \\ + \mathbb{E}_{\mathbf{x}^t \sim \mathcal{D}_t}&\left[-\log D_{\text{sep}}\left(G_t\left(\mathbf{x}^t\right)\right)_3 - \log D_{\text{sep}}\left(G_h\left(\mathbf{x}^t\right)\right)_1\right], \\ &\quad (3)\end{aligned}$$

where the subscripts 1, 2, and 3 correspond to shared features, source-domain dependent features, and target-domain dependent features, respectively.

The adaptation discriminator has a sigmoid output to determine whether the shared features come from the source domain or target domain. Its loss function is

$$\begin{aligned}\mathcal{L}_{\text{adt}} = &- \mathbb{E}_{\mathbf{x}^s \sim \mathcal{D}_s} \log D_{\text{adt}}\left(G_h\left(\mathbf{x}^s\right)\right) \\ &- \mathbb{E}_{\mathbf{x}^t \sim \mathcal{D}_t} \log\left[1 - D_{\text{adt}}\left(G_h\left(\mathbf{x}^t\right)\right)\right]. \quad (4)\end{aligned}$$

The total loss for training the ADSAN is

$$\mathcal{L}_{\text{ADSAN}} = \mathcal{L}_{\text{task}} + \alpha\mathcal{L}_{\text{sep}} - \beta\mathcal{L}_{\text{adt}} + \gamma\mathcal{L}_{\text{recon}}. \quad (5)$$

And the minimax optimization can be summarized as:

$$\min_{\theta_c, \theta_s, \phi_s, \phi_t, \phi_c, \phi_r} \max_{\theta_d} \mathcal{L}_{\text{ADSAN}}\left(\theta_c, \theta_s, \theta_d, \phi_s, \phi_t, \phi_c, \phi_r\right), \quad (6)$$

where $\theta_c, \theta_s, \theta_d, \phi_s, \phi_t, \phi_c, \phi_r$ are the parameters of the speaker discriminator, separation discriminator, adaptation discriminator, source encoder, target encoder, shared encoder, and shared decoder, respectively.

The difference between the DSN and our proposed ADSAN is the definition of $\mathcal{L}_{\text{sep}}$. For the DSN, the difference between domain-specific representations and the common representations is determined by orthogonality, i.e.,

$$\mathcal{L}_{\text{sep}}^{\text{DSN}} = \left\|\mathbf{z}_s^s \cdot \mathbf{z}_h^s\right\|_F^2 + \left\|\mathbf{z}_t^t \cdot \mathbf{z}_h^t\right\|_F^2 . \quad (7)$$

For ADSAN, on the other hand, the difference is defined by the classification loss in Eq. 3.

2.2. Mutual Information Neural Estimator

As mentioned previously, a flexible decoder may prevent the shared features $\left\{\mathbf{z}_h^s, \mathbf{z}_h^t\right\}$ from retaining task-related information. One possible solution is to maximize the mutual information between the encoder's output and the encoder's input.

Mutual information is a measure of information shared between random variables. If two variables are independent, the mutual information is zero and high mutual information indicates high dependency between two random variables. However, mutual information is tractable only for discrete random variables or continuous random variables with known probability distributions. Traditional approaches are non-parametric [17, 18, 19, 20] or rely on the approximate Gaussianity of data distributions [21]. These approaches cannot scale well with sample size or dimension [22]. To address this issue, Belghazi *et al.* [23] proposed a mutual information neural estimator (MINE) to estimate mutual information. They demonstrated that MINE can help to improve the training of adversarial networks. Therefore, we applied the MINE to estimate the mutual information.

MINE utilizes a deep neural network with parameters $\theta \in \Theta$ to find a lower bound of the mutual information:

$$I(X;Z) \geq I_\Theta(X,Z), \quad (8)$$

where $I_\Theta(X,Z)$ is defined as

$$I_\Theta(X,Z) = \sup_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}_{XZ}}\left[T_\theta\right] - \log\left(\mathbb{E}_{\mathbb{P}_X\otimes\mathbb{P}_Z}\left[e^{T_\theta}\right]\right). \quad (9)$$

In Eq. 9, $T_\theta : \mathcal{X} \times \mathcal{Z} \rightarrow \mathbb{R}$ is a function parametrized by the deep neural network. The expectations in Eq. 9 are estimated using empirical samples from $\mathbb{P}_{XZ}$ and $\mathbb{P}_X \otimes \mathbb{P}_Z$, where $\mathbb{P}_{XZ}$ is the joint distribution and $\mathbb{P}_X \otimes \mathbb{P}_Z$ is the product of the marginal distributions. Alternatively, it can be done by shuffling the samples from the joint distribution along the batch axis.

The training of MINE is realized by minimizing the loss

$$\mathcal{L}_{\text{MINE}}(\mathbf{x},\mathbf{z}) = -I_\Theta\left(\mathbf{x}^s; G_h\left(\mathbf{x}^s\right)\right). \quad (10)$$

By incorporating MINE into the ADSAN, we can calculate the lower bound of the mutual information between $\mathbf{x}^s$ and $\mathbf{z}_h^s$ as well as $\mathbf{x}^t$ and $\mathbf{z}_h^t$. Therefore, the shared encoder is trained to maximize the estimated mutual information:

$$I_\Theta(\mathbf{x},\mathbf{z}) = \sigma_1 I_\Theta\left(\mathbf{x}^s; G_h\left(\mathbf{x}^s\right)\right) + \sigma_2 I_\Theta\left(\mathbf{x}^t; G_h\left(\mathbf{x}^t\right)\right), \quad (11)$$

where

$$\begin{aligned} I_\Theta\left(\mathbf{x}^s; G_h\left(\mathbf{x}^s\right)\right) = \sup_{\theta\in\Theta} & \mathbb{E}_{p(\mathbf{x}^s,\mathbf{z}_h^s)}\left[T_\theta\left(\mathbf{x}^s, G_h\left(\mathbf{x}^s\right)\right)\right] \\ & - \log \mathbb{E}_{p(\mathbf{x}^s)p(\mathbf{z}_h^s)}\left[e^{T_\theta\left(\mathbf{x}^s, G_h\left(\mathbf{x}^s\right)\right)}\right] \end{aligned} \quad (12)$$

$$\begin{aligned} I_\Theta\left(\mathbf{x}^t; G_h\left(\mathbf{x}^t\right)\right) = \sup_{\theta\in\Theta} & \mathbb{E}_{p(\mathbf{x}^t,\mathbf{z}_h^t)}\left[T_\theta\left(\mathbf{x}^t, G_h\left(\mathbf{x}^t\right)\right)\right] \\ & - \log \mathbb{E}_{p(\mathbf{x}^t)p(\mathbf{z}_h^t)}\left[e^{T_\theta\left(\mathbf{x}^t, G_h\left(\mathbf{x}^t\right)\right)}\right]. \end{aligned} \quad (13)$$

And the total loss of the adversarial separation and adaptation network with a mutual information neural estimator is

$$\mathcal{L}_{\text{ADSAN}} = \mathcal{L}_{\text{task}} + \alpha\mathcal{L}_{\text{sep}} - \beta\mathcal{L}_{\text{adt}} + \gamma\mathcal{L}_{\text{recon}} - I_\Theta(\mathbf{x},\mathbf{z}). \quad (14)$$

The parameters of components in the networks are trained with the following minimax optimization:

$$\min_{\theta_c,\theta_s,\phi_s,\phi_t,\phi_c,\phi_r,\theta} \max_{\theta_d} \mathcal{L}_{\text{ADSAN}}\left(\theta_c,\theta_s,\theta_d,\phi_s,\phi_t,\phi_c,\phi_r,\theta\right). \quad (15)$$

3. Experimental Setting

X-vectors extracted from near-field microphone speech and far-field microphone speech were respectively used as the source and target domain data to train the ADSAN to produce domain-invariant features. The domain-invariant features from the source domain were used to train a PLDA backend. Then the trained PLDA backend was used to test the domain-invariant features from the target domain.

3.1. Datasets

The source domain comprises utterances from VoxCeleb1 [24] and VoxCeleb2 [25] to conform to the fixed condition of the VOiCES Challenge 2019 [26]. The combination of VoxCeleb1, VoxCeleb2, and their augmented sets were used to train an x-vector extractor. The augmented sets were obtained using MUSAN [27] based on the Kaldi's receipts. There are ∼2.2M utterances spoken by 7,323 speakers from the source domain. The trained x-vector extractor was then used to extract x-vectors of the utterances from the target domain.

The target domain comprises utterances from the VOiCES Challenge 2019, which consists of a development set and an evaluation set. There are 15,904 utterances from 196 speakers in the development set, and 11,392 utterances in the evaluation set.

When training the adaptation networks, the source domain training data were selected from the original VoxCeleb1 & 2 without augmentation. Therefore, the utterances from the source domain can be considered as "clean", while the utterances from the target domain are "noisy". The number of utterances per speaker in VoxCeleb1 & 2 ranges from 20 to 500, which may cause imbalance training in the speaker discriminator (D_{spk}). To address this issue, at least M_{min} and at most M_{max} utterances with the highest signal-to-noise ratios (SNRs) for each speaker from the source domain were selected as source training data. This selection also helps to aggravate the mismatch between the source and target domains, thereby making the results more meaningful. Inspired by [12], we estimated the SNRs using the WADA-SNR algorithm [28]. In our experiments, M_{min} is the minimum number of utterances of valid speakers, which is 21; M_{max} is the median number of utterances of valid speakers, which is 124. For the target domain, we used the utterances in the development set to adapt the ADSAN model and tested the performance on the evaluation set.

3.2. Baseline Systems

In addition to comparing with a baseline system that uses raw x-vectors, i.e., without any transformations, we also compared our proposed network with some other networks, including a domain adaptation neural network (DANN) [29], a DANN with MINE, and a domain separation network (DSN) [13]. Referring to Eq. 14 and Eq. 11, if the weight parameters α, γ, σ_1, and σ_2 are set to zero, then the proposed network becomes a DANN; if setting σ_1 and σ_2 to zero and replacing $\mathcal{L}_{\text{sep}}$ (Eq. 3) with Eq. 7, a DSN can be obtained. These three systems were used as the baseline systems in this paper, and the configurations of the subnetworks in the DANN and the DSN are the same as those in the proposed network.

3.3. Networks without MINE

To exclude the MINE in the training, σ_1 and σ_2 in Eq. 11 were set to 0. The remaining weight parameters for the ADSAN and the DSN were set as follows: $\alpha = 1.0, \beta = 0.1$, and $\gamma = 1.0$; for DANN, $\alpha = \gamma = \sigma_1 = \sigma_2 = 0$. The sub-networks are fully connected neural networks with two hidden layers, each with 1024 neurons; the separation discriminator is the exception, which has one hidden layers and 100 neurons. The learning rate for all sub-networks is 0.0001 and batch normalization was applied to the three encoders and the shared decoder. The activation function in all sub-networks is the leaky ReLU.

3.4. Networks with MINE

Systems improved upon the DSN and the ADSAN were created by incorporating a MINE into the loss function. The weight parameters in the loss function (Eq. 14 and Eq. 11) were set to $\alpha = 1.0, \beta = 0.1, \gamma = 1.0, \sigma_1 = 0.2$, and $\sigma_2 = 0.4$. The MINE is a deep neural network whose size depends on the size of data being estimated. In our experiments, the MINE has two hidden layers, each with 100 neurons. Before training the adaptation models, the MINE was pre-trained with the whole training data from the source domain for 5 epochs. Then, for each epoch in the main training loop, the adaptation and separation network (comprising G_h, G_s, G_t, R, D_{spk}, D_{adt}, and D_{sep}) and the MINE were trained consecutively, i.e., when the adaptation and the separation networks were trained, the MINE is frozen, and vice versa when the MINE was trained.

Because mutual information is unbounded, it could be infinitely large. We applied gradient clipping to prevent infinite gradient. This strategy can ensure that the networks can be updated normally. The initial learning rate of the MINE is 0.0001, which decays every 1000 steps at a decay rate of 0.96. The learning rate for the other sub-networks is 0.00005.

3.5. PLDA Training and Scoring

As a pre-processing step, we projected the raw x-vector to a 200-dimensional space by linear discriminant analysis followed by length normalization. Then, we used the pre-processed x-vectors derived from Voxceleb1 & 2 (with and without data augmentation) to train the PLDA models of our baseline systems (the first row of Tables 1 and 2). We performed the same pre-processing on the transformed x-vectors obtained by various domain adaptation networks (DANN, DSN, and ADSAN) to train other sets of PLDA models. For scoring, the pre-processed x-vectors and the network-transformed pre-processed x-vectors were derived from VOiCES data and were fed to the respective PLDA models. When computing the PLDA scores, the x-vectors were centered by using the mean x-vector of the enrollment data in the development set of VOiCES.

4. Results and Discussions

For fair comparisons, the same preprocessing was applied to all the systems. In addition, the label information of the target domain data was not used during the entire training process.

Table 1 and Table 2 show the performance on the development set and the evaluation set obtained by different systems, respectively. It can be observed that the domain adaptation networks are more effective for the evaluation set in which the utterances are more noisy and are subject to more severe reverberation. The DANN with MINE achieves the worst performance, which indicates a negative transfer. Compared with the DANN, the DSN and ADSAN can achieve better results, and both the EER and minDCF are further reduced when the MINE was incorporated into the systems. When training the MINE in the DANN, the mutual-information loss always converges to a very small value (closed to 0). This indicates that the MINE cannot play a useful role in the DANN.

For the evaluation set, the DSN with MINE achieves the lowest EER while the ADSAN with MINE achieves the lowest minDCF. Different from the DSN, the ADSAN applies a moderate constraint on the difference between domain-specific components and domain-shared components (Eq. 3 vs. Eq. 7). Investigations via t-SNE plots (not shown) produced by the DSN- and ADSAN-transformed vectors suggest that the distributions of the ADSAN-transformed vectors are closer to the distributions of the original vectors; they also exhibit a smaller distance between the source domain and the target domain. These results reveal that the proposed ADSAN can produce domain-invariant and speaker discriminative representations, which are beneficial for speaker verification.

Table 1: *Performance on the VOiCES development set using Voxceleb data (with or without augmentation) for training the PLDA.*

System	VoxCeleb1&2		VoxCeleb1&2+aug	
	EER(%)	minDCF	EER(%)	minDCF
x-vector	3.07	**0.3595**	3.00	**0.3342**
DANN	3.08	0.4519	3.34	0.4502
DANN+MINE	4.87	0.5301	4.88	0.5291
DSN	3.15	0.3886	3.21	0.3831
DSN+MINE	3.03	0.4194	3.15	0.4155
ADSAN	2.98	0.4545	3.00	0.4377
ADSAN+MINE	**2.91**	0.3789	**2.90**	0.3700

Table 2: *Performance on the VOiCES evaluation set using Voxceleb data (with or without augmentation) for training the PLDA.*

System	VoxCeleb1&2		VoxCeleb1&2+aug	
	EER(%)	minDCF	EER(%)	minDCF
x-vector	9.03	0.8157	7.36	0.6125
DANN	7.37	0.7291	6.89	0.6794
DANN+MINE	10.03	0.8163	9.4	0.7908
DSN	7.24	0.6892	6.7	0.6307
DSN+MINE	**6.76**	0.6588	**6.51**	0.6309
ADSAN	7.16	0.7939	6.58	0.7021
ADSAN+MINE	6.98	**0.5934**	6.76	**0.5989**

5. Conclusions

In this paper, we propose an adversarial separation and adaptation network motivated by the domain separation network. We also propose to incorporate a mutual information neural estimator into these two domain separation networks. The proposed approaches can enforce the shared encoder to disentangle the domain-invariant features from the domain-specific properties. Our experiments on the VOiCES corpus show that the proposed approaches can outperform the DANN. And training the networks with the consideration of mutual information can further increase the speaker information in the extracted features.

6. References

[1] D. Garcia-Romero and A. McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *Proc. ICASSP*, 2014, pp. 4047–4051.

[2] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. of AAAI Conference on Artificial Intelligence*, vol. 6, no. 7, 2016.

[3] K. A. Lee, Q. Wang, and T. Koshinaka, "The CORAL+ algorithm for unsupervised domain adaptation of PLDA," in *Proc. ICASSP*, 2019, pp. 5821–5825.

[4] A. Kanagasundaram, D. Dean, and S. Sridharan, "Improving out-domain PLDA speaker verification using unsupervised interdataset variability compensation approach," in *Proc. ICASSP*, 2015, pp. 4654–4658.

[5] H. Aronowitz, "Inter dataset variability compensation for speaker recognition," in *Proc. ICASSP*, 2014, pp. 4002–4006.

[6] M. H. Rahman, A. Kanagasundaram, D. Dean, and S. Sridharan, "Dataset-invariant covariance normalization for out-domain PLDA speaker verification," in *Proc. Interspeech*, 2015, pp. 1017–1021.

[7] J. Huang and T. Bocklet, "Intel far-field speaker recognition system for voices challenge 2019," in *Proc. Interspeech 2019*, 2019, pp. 2473–2477.

[8] W. W. Lin, M. W. Mak, and J. T. Chien, "Multisource i-vectors domain adaptation using maximum mean discrepancy based autoencoders," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 12, pp. 2412–2422, 2018.

[9] Q. Wang, W. Rao, S. Sun, L. Xie, E. S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *Proc. ICASSP*, 2018, pp. 4889–4893.

[10] Y. Tu, M. W. Mak, and J. T. Chien, "Variational Domain Adversarial Learning for Speaker Verification," in *Proc. Interspeech*, 2019, pp. 4315–4319.

[11] J. Rohdin, T. Stafylakis, A. Silnova, H. Zeinali, L. Burget, and O. Plchot, "Speaker verification using end-to-end adversarial language adaptation," in *Proc. ICASSP*, 2019, pp. 6006–6010.

[12] P. S. Nidadavolu, S. Kataria, J. Villalba, and N. Dehak, "Low-resource domain adaptation for speaker recognition using cycle-gans," *arXiv preprint arXiv:1910.11909*, 2019.

[13] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 343–351.

[14] Z. Meng, Z. Chen, V. Mazalov, J. Li, and Y. Gong, "Unsupervised adaptation with domain separation networks for robust speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 214–221.

[15] A. Makhzani and B. J. Frey, "PixelGAN autoencoders," in *Advances in Neural Information Processing Systems*, 2017, pp. 1975–1985.

[16] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, "Variational lossy autoencoder," *arXiv preprint arXiv:1611.02731*, 2016.

[17] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical Review A*, vol. 33, no. 2, p. 1134, 1986.

[18] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E*, vol. 69, no. 6, p. 066138, 2004.

[19] S. Gao, G. Ver Steeg, and A. Galstyan, "Efficient estimation of mutual information for strongly dependent variables," in *Artificial Intelligence and Statistics*, 2015, pp. 277–286.

[20] W. Gao, S. Oh, and P. Viswanath, "Demystifying fixed k-nearest neighbor information estimators," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5629–5661, 2018.

[21] M. M. V. Hulle, "Edgeworth approximation of multivariate differential entropy," *Neural Computation*, vol. 17, no. 9, pp. 1903–1910, 2005.

[22] S. Gao, G. Ver Steeg, and A. Galstyan, "Efficient estimation of mutual information for strongly dependent variables," in *Artificial Intelligence and Statistics*, 2015, pp. 277–286.

[23] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, D. Hjelm, and A. Courville, "Mutual information neural estimation," in *International Conference on Machine Learning*, 2018, pp. 530–539.

[24] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[25] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018.

[26] M. K. Nandwana, J. Van Hout, M. McLaren, C. Richey, A. Lawson, and M. A. Barrios, "The voices from a distance challenge 2019 evaluation plan," *arXiv preprint arXiv:1902.10828*, 2019.

[27] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[28] C. Kim and R. M. Stern, "Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis," in *Proc. Interspeech*, 2008.

[29] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.