INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Variational Domain Adversarial Learning for Speaker Verification

Youzhi Tu[1], *Man-Wai Mak*[1], *Jen-Tzung Chien*[2]

[1]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR
[2]Department of Electrical and Computer Engineering, National Chiao Tung University, Taiwan

918tyz@gmail.com, enmwmak@polyu.edu.hk, jtchien@nctu.edu.tw

Abstract

Domain mismatch refers to the problem in which the distribution of training data differs from that of the test data. This paper proposes a variational domain adversarial neural network (VDANN), which consists of a variational autoencoder (VAE) and a domain adversarial neural network (DANN), to reduce domain mismatch. The DANN part aims to retain speaker identity information and learn a feature space that is robust against domain mismatch, while the VAE part is to impose variational regularization on the learned features so that they follow a Gaussian distribution. Thus, the representation produced by VDANN is not only speaker discriminative and domain-invariant but also Gaussian distributed, which is essential for the standard PLDA backend. Experiments on both SRE16 and SRE18-CMN2 show that VDANN outperforms the Kaldi baseline and the standard DANN. The results also suggest that VAE regularization is effective for domain adaptation.

Index Terms: speaker verification, domain adaptation, domain adversarial training, variational autoencoder

1. Introduction

Speaker verification (SV) is to determine whether the identity of a test utterance matches that of a target speaker. To achieve the best performance, SV systems are trained on data sharing the same distribution with that of the test data (in the target domain or in-domain). In practice, however, due to the high cost of data labeling, usually only a small amount of labeled data from the target domain are available. When the distribution of the source-domain training data differs from that of the target-domain data, domain mismatch will occur, which poses a great challenge to speaker verification. To overcome this problem, domain adaptation (DA) is applied to transfer the knowledge extracted from the source domain to the target domain.

Earlier DA methods are implemented in a supervised manner, which require speaker labels in the target-domain data [1, 2]. More recently, research in DA has been focusing on the unsupervised situation where only some unlabeled target-domain data are available besides large amount of labeled source domain data. One approach to achieving unsupervised DA is to hypothesize the speaker labels through clustering [3, 4, 5]. With these hypothesized labels, we may adapt the probabilistic linear discriminant analysis (PLDA) model to the target domain by computing the in-domain covariance matrices and then interpolating them with the out-of-domain covariance matrices. Another category is to learn a domain-invariant space for transforming the source domain i-vectors [6] and then use the transformed i-vectors to train a PLDA model. For example, Aronowitz [7] introduced the inter dataset variability compensation (IDVC) based on nuisance attribute projection (NAP) [8] to reduce the domain mismatch in the i-vector space. Instead of directly capturing the mismatch between the out-domain data and in-domain data as in inter dataset variability (IDV) [9], Rahman *et al.* [10] measured the mismatch relative to the global mean of i-vectors and relocated both in-domain and out-domain i-vectors into a dataset-invariant space. In [11], autoencoder-based domain adaptation was proposed to transfer channel information from the source domain to the target domain. In [12, 13], Lin *et al.* applied maximum mean discrepancy to measure the degree of domain mismatch across multiple domains and incorporated the measure into the objective function for training autoencoders. The bottleneck features extracted from the autoencoders are shown to be less domain dependent, resulting in performance gain in SRE16 data.

With the emergence of generative adversarial networks (GANs) [14], adversarial learning has been applied for DA to create a domain-invariant space [15, 16, 17]. For instance, in [18], an encoder and a discriminator network are adversarially trained to produce bottleneck features that are robust against noise. Wang *et al.* [19] applied domain adversarial training (DAT) [16] to generate speaker discriminative and domain-invariant feature representations by incorporating a speaker classifier into an adversarial autoencoder, which outperforms traditional DA approaches on the 2013 domain adaptation challenge. Rohdin *et al.* [20] also followed the DAT framework but implemented the adversarial learning in an end-to-end fashion to minimize language mismatch while retaining speaker discrimination capability.

Although adversarial learning based unsupervised DA [18, 19] has greatly boosted the performance of SV systems under domain mismatch scenarios, the adversarial training may lead to non-Gaussian latent vectors, which do not meet the Gaussianity requirement of the PLDA backend. This problem can be solved by using heavy-tailed PLDA [21, 22] or applying i-vector length normalization [23]. However, the former is more computationally expensive than the Gaussian PLDA and the latter is not really a Gaussianization procedure but a sub-optimal compromise. Recently, there have been some work trying to Gaussianize the distribution of speaker embeddings obtained by neural networks, e.g., [24] applied Gaussian-constrained training by incorporating an l_2-regularizer into the standard cross-entropy loss. Kingma and Welling [25] proposed the variational autoencoders (VAEs) as a solution to performing inference in directed probabilistic models whose latent variables have intractable posterior distributions. One desirable property of a VAE is that its KL-divergence term can be considered as a regularizer that constrains the encoder to produce latent vectors that follow a desired distribution. Our method leverages this property to encourage the encoder to produce Gasussian latent vectors, which will be amenable to PLDA modeling. A similar approach using VAE for Gaussian regularization for speaker embeddings was proposed in [26]; however, it is not targeted to

Copyright © 2019 ISCA

http://dx.doi.org/10.21437/Interspeech.2019-2168

address the domain mismatch problem which is the main focus of this paper.

In this paper, we adopt the idea of the domain adversarial neural network (DANN) in [16, 19] for unsupervised DA, aiming to produce both speaker discriminative and domain-invariant features. Simultaneously, we incorporate variational regularization into DAT to ensure that the produced representations follow a Gaussian distribution so that we can directly apply the Gaussian PLDA model. The resulting network is referred to as variational domain adversarial neural network (VDANN). To our best knowledge, VDANN is the first to combine both DAT and VAE for domain-invariant speaker verification.

2. Methodology

Suppose we have a training set $\mathcal{X} = \{\mathcal{X}^{(r)}\}_{r=1}^{R}$ comprising samples from R domains, where $\mathcal{X}^{(r)} = \{\mathbf{x}_1^{(r)}, \ldots, \mathbf{x}_{N_r}^{(r)}\}$ contains N_r samples from the r-th domain. Also we denote $\mathbf{y}$ and $\mathbf{d}$ as the one-hot speaker and domain labels, respectively.

2.1. Variational Domain Adversarial Neural Network (VDANN)

Although DANN has been shown to be superior to conventional DA [19], there is no guarantee that the learned features follow a Gaussian distribution, which is essential for the Gaussian PLDA backend. To alleviate this limitation, we incorporate a VAE into the DAT so that the learned features are not only speaker discriminative and domain-variant but also Gaussian distributed.

Originally, VAE was proposed to solve the variational approximate inference problem by maximizing the evidence lower bound (ELBO) [25]:

$$L_{\text{ELBO}}(\theta, \phi) = \sum_{r=1}^{R}\sum_{i=1}^{N_r}\Big\{-\text{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}_i^{(r)})\|p_\theta(\mathbf{z})\right) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i^{(r)})}\left[\log p_\theta\left(\mathbf{x}_i^{(r)}|\mathbf{z}\right)\right]\Big\}, \quad (1)$$

where ϕ and θ are parameters of the encoder and decoder, respectively. $q_\phi(\mathbf{z}|\mathbf{x})$ is an approximate posterior to the intractable true posterior $p_\theta(\mathbf{z}|\mathbf{x})$, which represents a recognition model encoding input samples into the latent space, while $p_\theta(\mathbf{x}|\mathbf{z})$ denotes a generative model which decodes latent representations back to the original data space.

One desirable property of VAE is that the first term on the right-hand side of Eq. 1 can be considered as a regularizer that constrains the variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$ to be close to the desired prior $p_\theta(\mathbf{z})$. Therefore, if we constrain $p_\theta(\mathbf{z})$ to be a multivariate Gaussian distribution, the encoder will likely to produce Gasusian latent vectors, which is amenable to PLDA modeling.

Assume that $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ and the true posterior also follows a Gaussian distribution with an approximate diagonal covariance matrix. Then the approximate posterior will take the following form

$$\log q_\phi(\mathbf{z}|\mathbf{x}_i) = \log \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2\mathbf{I}\right), \quad (2)$$

where the mean $\boldsymbol{\mu}_i$ and standard derivation $\boldsymbol{\sigma}_i$ are outputs of the encoder given input $\mathbf{x}_i$, and they are parameterized by ϕ. Applying the reparameterization trick in sampling latent variables from the variational approximate posterior, we obtain the l-th latent sample $\mathbf{z}_{il} = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}_l$, where $\boldsymbol{\epsilon}_l \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ is the Hadamard product. Substitute these terms into Eq. 1, we have the Gaussian VAE loss [25]:

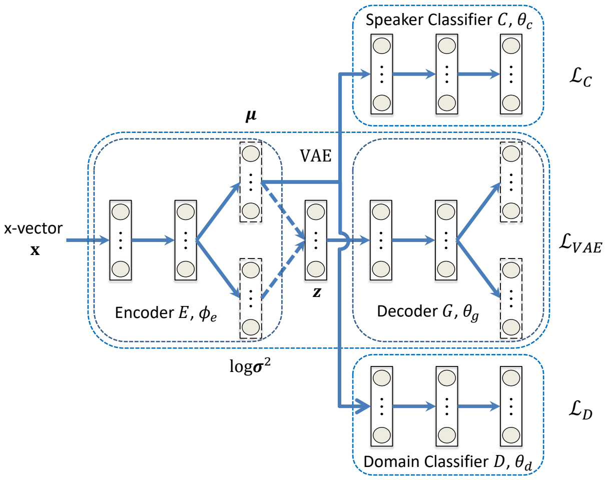

Figure 1: *Schematic of VDANN. The solid and dashed arrows represent network connections and stochastic sampling, respectively.*

$$\mathcal{L}_{\text{VAE}}(\theta, \phi) \simeq -\sum_{r=1}^{R}\sum_{i=1}^{N_r}\Bigg\{\frac{1}{2}\sum_{j=1}^{J}\left[1 + \log\left(\sigma_{ij}^{(r)}\right)^2 - \left(\mu_{ij}^{(r)}\right)^2 - \left(\sigma_{ij}^{(r)}\right)^2\right] + \frac{1}{L}\sum_{l=1}^{L}\log p_\theta\left(\mathbf{x}_i^{(r)}|\mathbf{z}_{il}\right)\Bigg\}, \quad (3)$$

where J is the dimension of $\mathbf{z}$ and L denotes the number of latent samples. In practice, we set $L = 1$.

As shown in Figure 1, the proposed VDANN consists of a speaker predictor C, a domain classifier D and a VAE. The latter contains an encoder E and a decoder G. The network parameters are denoted as θ_c, θ_d, ϕ_e and θ_g, respectively. Through adversarial training, the VDANN learns a domain-invariant space across multiple domains. Specifically, applying adversarial training on E while keeping θ_d fixed together with minimizing the cross-entropy loss of C with respect to ϕ_e will make E to produce a domain-invariant but speaker discriminative representation through the nodes denoted by $\boldsymbol{\mu}$ in Figure 1. While it is also possible to use the concatenation of $\boldsymbol{\mu}$ and $\log\boldsymbol{\sigma}^2$ as the representation, our preliminary experiments suggest that this gives almost the same performance as using $\boldsymbol{\mu}$ only.

To train this network, we define the loss of VDANN as:

$$\mathcal{L}_{\text{VDANN}}(\theta_c, \theta_d, \phi_e, \theta_g) = \mathcal{L}_C(\theta_c, \phi_e) - \alpha\mathcal{L}_D(\theta_d, \phi_e) + \beta\mathcal{L}_{\text{VAE}}(\phi_e, \theta_g), \quad (4)$$

where

$$\mathcal{L}_C(\theta_c, \phi_e) = \sum_{r=1}^{R}\mathbb{E}_{p_{\text{data}}(\mathbf{x}^{(r)})}\left\{-\sum_{k=1}^{K}y_k^{(r)}\log C\left(E\left(\mathbf{x}^{(r)}\right)\right)_k\right\}, \quad (5)$$

$$\mathcal{L}_D(\theta_d, \phi_e) = \sum_{r=1}^{R}\mathbb{E}_{p_{\text{data}}(\mathbf{x}^{(r)})}\left\{-\log D\left(E\left(\mathbf{x}^{(r)}\right)\right)_r\right\}, \quad (6)$$

and $\mathcal{L}_{\text{VAE}}$ takes similar form as in Eq. 3 except that the parameters of the encoder and decoder change to ϕ_e and θ_g, respectively. The subscript k in the categorical cross-entropy loss of the speaker classifier C in Eq. 5 indexes the speakers and represents the k-th output of the classifier. The hyperparameters α and β control the contribution of individual losses that shape the features produced by E.

During training, for each mini-batch, we first optimize D by minimizing the domain classification loss. Parameters of D are then fixed while training the remaining parts of the VDANN. To incorporate speaker information into E, speaker prediction loss is minimized; simultaneously we maximize the domain classification loss so that we can learn a domain-invariant space for E. Moreover, the VAE loss is minimized to regularize the learned features to be Gaussian. To summarize, we optimize the VDANN as follows:

$$\min_{\theta_c,\phi_e,\theta_g} \max_{\theta_d} \mathcal{L}_{\text{VDANN}}(\theta_c,\theta_d,\phi_e,\theta_g). \tag{7}$$

Eq. 7 can be divided into the following min-max procedure:

$$\hat{\theta}_d = \underset{\theta_d}{\operatorname{argmax}}\, \mathcal{L}_{\text{VDANN}}(\hat{\theta}_c,\theta_d,\hat{\phi}_e,\hat{\theta}_g), \tag{8}$$

$$\left(\hat{\theta}_c,\hat{\phi}_e,\hat{\theta}_g\right) = \underset{\theta_c,\phi_e,\theta_g}{\operatorname{argmin}}\, \mathcal{L}_{\text{VDANN}}(\theta_c,\hat{\theta}_d,\phi_e,\theta_g), \tag{9}$$

where symbols with a hat (e.g., $\hat{\theta}_c$) on the right-hand side of Eq. 8 and Eq. 9 mean that they are fixed when optimizing the target parameters. After training, we may extract the transformed features from the mean layer of the encoder E (denoted by $\boldsymbol{\mu}$ in Figure 1). Since the variational approximate posterior is regularized to follow a Gaussian distribution, features produced from the encoder will also likely to be Gaussian.

2.2. Relationship with Domain Adversarial Neural Network (DANN)

In [19], DANN was applied to produce features that are not only speaker discriminative but invariant to domain shift. As the training data in [19] come from two domains, $R = 2$. A standard DANN consists of three networks: a feature extractor E, a speaker predictor C and a domain classifier D, which is a special case of VDANN. By setting $\beta = 0$ in Eq. 4, we have the loss function of DANN:

$$\mathcal{L}_{\text{DANN}}(\theta_c,\theta_d,\phi_e) = \mathcal{L}_C(\theta_c,\phi_e) - \alpha\mathcal{L}_D(\theta_d,\phi_e), \tag{10}$$

where θ_c, θ_d and ϕ_e are the parameters for C, D, E, respectively. $\mathcal{L}_C$ and $\mathcal{L}_D$ are the same as Eq. 5 and Eq. 6, respectively. α controls the trade-off between the two objectives whose gradient is back-propagated into the feature extractor during training. The parameters are optimized as follows:

$$\hat{\theta}_d = \underset{\theta_d}{\operatorname{argmax}}\, \mathcal{L}_{\text{DANN}}\left(\hat{\theta}_c,\theta_d,\hat{\phi}_e\right), \tag{11}$$

$$\left(\hat{\theta}_c,\hat{\phi}_e\right) = \underset{\theta_c,\phi_e}{\operatorname{argmin}}\, \mathcal{L}_{\text{DANN}}\left(\theta_c,\hat{\theta}_d,\phi_e\right). \tag{12}$$

Since there is no extra constraint on the distribution of features learned from DANN, the adversarial training may lead to non-Gaussian latent vectors, which do not meet the Gaussianity requirement of the PLDA backend.

3. Experimental Setup

The experiments were conducted based on x-vectors [27] and SV performance was evaluated on the NIST SRE16 and SRE18-CMN2.

3.1. Acoustic Features and X-vector Extraction

We used the Kaldi's SRE16 recipe[1] to extract 23-dimensional MFCCs, followed by energy-based VAD. After that, 512-dimensional x-vectors were extracted using the pre-trained DNN available from the Kaldi repository.

3.2. VDANN and DANN Training

We used data from four domains to train the VDANN and DANN. The statistics of the training data are shown in Table 1. Note that each training set is a subset of the original set. For example, the minimum number of x-vectors per speaker is 30 for both SRE04–10 and Voxceleb1. SwitchBoard II was selected from Switchboard 2 Phases I–III to ensure that there are at least 20 x-vectors for each speaker, while each speaker in the SITW set has at least 15 x-vectors.

Table 1: *Statistics of training sets for VDANN*

Dataset	No. of speakers	No. of utterances
SRE04–10	1,806	54,180
Voxceleb1	1,251	37,530
SwitchBoard II	273	6,962
SITW	203	3,700

As shown in Figure 1, there are four sub-networks in the VDANN. The encoder in the VAE part has two hidden layers, and the number of nodes in each hidden layer is 1024. We used ReLU as the activation function in each layer, followed by batch normalization. The dimension of the latent space was set to 400. The configuration of the decoder is the same as that of the encoder. The output layers of both the encoder and decoder are linear. For the speaker classifier, we used a 1024-1024 hidden layer structure with Leaky ReLU activations, and batch normalization and dropout layers were appended after each layer. The output layer has 3533 nodes with a softmax function. The configuration of the domain classifier is similar to that of the speaker classifier except that the number of nodes in the two hidden layers are 128 and 32, respectively. It has 4 nodes in the output layer corresponding to the 4 domains in Table 1.

The DANN has the same structure as the VDANN, except for the missing of the VAE decoder and the sampling procedure. Another difference is that we used the loss function in Eq. 10 rather than the loss function in Eq. 4.

For DANN, we set $\alpha = 0.1$ in Eq. 10. For VDANN, we set $\alpha = 0.1$, $\beta = 0.1$ in Eq. 4 and set the standard deviation of all components in $\boldsymbol{\epsilon}$ in the stochastic sampling to 0.01.

3.3. PLDA Training and Scoring

We used the standard Gaussian PLDA backend for scoring. For the SRE16 evaluation task, the baseline PLDA model was trained on the NIST SRE 2004–2010 datasets and their augmented versions; while for the SRE18 evaluation experiments, the Mixer6 dataset and its augmentation were also added to the training sets. The augmentation step also follows the Kaldi's SRE16 recipe. Before PLDA training, x-vectors or their DNN-transformed versions were projected to a 150 dimensional space by an LDA transformation matrix, followed by length normalization. The LDA projection matrix was trained on the same dataset as in training PLDA models.

[1] http://kaldi-asr.org/

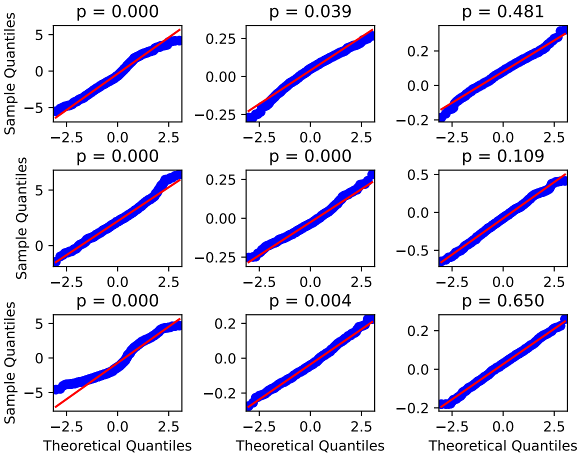

Figure 2: *Quantile-quantile (Q–Q) plots of the 11-th (the 1st row), 111-th (the 2nd row), and 211-th (the 3rd row) components of x-vectors (the 1st column), DANN-transformed x-vectors (the 2nd column), and VDANN-transformed x-vectors (the 3rd column). The vertical and horizontal axes correspond to the samples under test and the samples drawn from a standard normal distribution, respectively. The red line represents the situation of perfectly Gaussian. The p-values above the graphs were obtained from Shapiro–Wilk tests in which* $p > 0.05$ *means failing to reject the null hypothesis that the test samples come from a Gaussian distribution.*

Since there is severe domain mismatch between the PLDA training sets and SRE16/SRE18 evaluation data, development data were used for PLDA adaptation. Specifically, SRE16 unlabeled data were used to adapt the PLDA models for SRE16, while we used SRE18 unlabeled data for SRE18 PLDA adaptation. During adaptation for the baseline, the unlabeled data were used to adapt the out-of-domain PLDA model so that the adapted model better matches the statistics of the in-domain data. The adaptation is detailed in the Kaldi's SRE16 recipe.

The latent vectors extracted from VDANN and DANN systems followed the same pre-processing as the baseline except that the transformed x-vectors ($\boldsymbol{\mu}$ in Figure 1) rather than the original x-vectors were used for centering, LDA training, PLDA training, adaptation and scoring.

4. Results and Discussions

Figure 2 shows the normal Q–Q plots of three randomly selected dimensions of x-vectors and the x-vectors transformed by DANN and VDANN. Evidently, the distribution of VDANN-transformed x-vectors is closer to a Gaussian distribution than the other two. This suggests that the VAE loss can make the latent vectors $\mathbf{z}$'s to follow a Gaussian distribution. The p-values obtained from Shapiro–Wilk tests [28] also suggest that the distribution of VDANN-transformed vectors is the closest to the standard Gaussian.

We followed the Kaldi's SRE16 recipe for SRE16/18 evaluation. For the baseline, the x-vectors were centered, LDA-transformed and length normalized before PLDA scoring. The same preprocessing was applied to the transformed x-vectors for the DANN and VDANN systems.

Table 2 shows the pooled evaluation performance of the systems in SRE16. From the left part of the table, without Kaldi PLDA adaptation, we can see that VDANN reduces the domain mismatch in both EER and minDCF compared to the baseline. Although DANN reduces the minDCF to some extent, it impairs the EER. The right part presents the results using Kaldi's PLDA adaptation as an *extra* adaptation. We see that Kaldi's PLDA adaptation is very powerful because even though the x-vectors have been processed by VDANN, they outperform the baseline by a small margin only. Comparing the results of VDANN with those of DANN reveals that applying variational regularization on the transformed x-vectors is effective for domain adaptation.

Performance on SRE18-CMN2 is shown in Table 3. From the table, we obtain similar conclusions as in SRE16: constraining the transformed x-vector distribution to be Gaussian via VAE is beneficial for overcoming domain mismatch.

The P-values of the McNemar's test [29] between the DANN and VDANN systems are 0 and 4.72×10^{-4} for SRE16 and SRE18 without PLDA adaptation, respectively, while they are 0 and 4.14×10^{-7} with PLDA adaptation. This means that the improvement of VDANN over DANN is statistically significant since we have $P < \alpha$ (typically α is 0.05, 0.01 or 0.001).

Table 2: *Performance on SRE16*

	No PLDA adaptation		PLDA adaptation	
	EER	minDCF	EER	minDCF
Baseline	11.30	0.890	8.27	0.604
DANN	11.62	0.822	8.43	0.599
VDANN	**11.17**	**0.798**	**8.21**	**0.584**

Table 3: *Performance on SRE18-CMN2*

	No PLDA adaptation		PLDA adaptation	
	EER	minDCF	EER	minDCF
Baseline	11.21	0.676	9.60	**0.575**
DANN	10.82	0.678	9.28	0.583
VDANN	**10.25**	**0.667**	**9.23**	0.576

5. Conclusions

In this paper, we proposed a network called VDANN to reduce domain mismatch. VDANN incorporates a VAE into domain adversarial training to impose a constraint on the distribution of the transformed x-vectors so that they are not only speaker discriminative and domain-invariant, but also conform to a Gaussian distribution. Experimental results show that VDANN is capable of reducing domain mismatch. The fact that VDANN consistently outperforms the standard DANN in both EER and minDCF on SRE16 and SRE18-CMN2 suggests that VAE regularization is effective for domain adaptation.

6. Acknowledgements

This work was supported by the RGC of Hong Kong SAR, Grant No. PolyU 152137/17E.

7. References

[1] D. Garcia-Romero, A. McCree, S. Shum, N. Brümmer, and C. Vaquero, "Supervised domain adaptation for i-vector based speaker

recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 4047–4051.

[2] J. Villalba and E. Lleida, "Bayesian adaptation of PLDA based speaker recognition to domains with scarce development data," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2012, pp. 47–54.

[3] S. H. Shum, D. A. Reynolds, D. Garcia-Romero, and A. McCree, "Unsupervised clustering approaches for domain adaptation in speaker recognition systems," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 266–272.

[4] D. Garcia-Romero, X. Zhang, A. McCree, and D. Povey, "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in *Proc. 2014 IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 378–383.

[5] L. Li and M. W. Mak, "Unsupervised domain adaptation for gender-aware PLDA mixture models," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5269–5273.

[6] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[7] H. Aronowitz, "Inter dataset variability compensation for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 4002–4006.

[8] A. Solomonoff, C. Quillen, and W. M. Campbell, "Channel compensation for SVM speaker recognition," in *Odyssey: The Speaker and Language Recognition Workshop*, 2004, pp. 57–62.

[9] A. Kanagasundaram, D. Dean, and S. Sridharan, "Improving out-domain PLDA speaker verification using unsupervised interdataset variability compensation approach," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 4654–4658.

[10] M. H. Rahman, A. Kanagasundaram, D. Dean, and S. Sridharan, "Dataset-invariant covariance normalization for out-domain PLDA speaker verification," in *Proc. INTERSPEECH15*, 2015, pp. 1017–1021.

[11] S. Shon, S. Mun, W. Kim, and H. Ko, "Autoencoder based domain adaptation for speaker recognition under insufficient channel information," in *Proc. INTERSPEECH17*, 2017, pp. 1014–1018.

[12] W. W. Lin, M. W. Mak, and J. T. Chien, "Multi-source i-vectors domain adaptation using maximum mean discrepancy based autoencoders," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 16, no. 12, pp. 2412–2422, 2018.

[13] W. Lin, M. W. Mak, Y. Tu, and J. T. Chien, "Semi-supervised nuisance-attribute networks for domain adaptation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and A. C. Y. Bengio, "Generative adversarial nets," in *Proc. Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[15] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," in *arXiv preprint arXiv:1511.05644*, 2015.

[16] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2130, 2016.

[17] J. Tsai and J. Chien, "Adversarial domain separation and adaptation," in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2017, pp. 1–6.

[18] H. Yu, Z. H. Tan, Z. Ma, and J. Guo, "Adversarial network bottleneck features for noise robust speaker verification," in *Proc. INTERSPEECH17*, 2017.

[19] Q. Wang, W. Rao, S. N. Sun, L. Xie, E. S. Chng, and H. Z. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 4889–4893.

[20] J. Rohdin, T. Stafylakis, A. Silnova, H. Zeinali, L. Burget, and O. Plchot, "Speaker verification using end-to-end adversarial language adaptation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 6006–6010.

[21] P. Kenny, "Bayesian speaker verification with heavy tailed priors," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2010, p. keynote presentation.

[22] A. Silnova, N. Brümmer, D. Garcia-Romero, D. Snyder, and L. Burget, "Fast variational bayes for heavy-tailed PLDA applied to i-vectors and x-vectors," in *Proc. INTERSPEECH18*, 2018, pp. 72–76.

[23] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. INTERSPEECH11*, 2011, pp. 249–252.

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

[25] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

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

[27] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5329–5333.

[28] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

[29] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 1989, pp. 532–535.