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

Data Augmentation using Variational Autoencoder for Embedding based Speaker Verification

Zhanghao Wu, Shuai Wang, Yanmin Qian, Kai Yu

MoE Key Lab of Artificial Intelligence
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{wz.wzh, feixiang121976, yanminqian, kai.yu}@sjtu.edu.cn

Abstract

Domain or environment mismatch between training and testing, such as various noises and channels, is a major challenge for speaker verification. In this paper, a variational autoencoder (VAE) is designed to learn the patterns of speaker embeddings extracted from noisy speech segments, including *i*-vector and *x*-vector, and generate embeddings with more diversity to improve the robustness of speaker verification systems with probabilistic linear discriminant analysis (PLDA) back-end. The approach is evaluated on the standard NIST SRE 2016 dataset. Compared to manual and generative adversarial network (GAN) based augmentation approaches, the proposed VAE based augmentation achieves a slightly better performance for *i*-vector on Tagalog and Cantonese with EERs of 15.54% and 7.84%, and a more significant improvement for *x*-vector on those two languages with EERs of 11.86% and 4.20%.

Index Terms: speaker verification, data augmentation, VAE, PLDA

1. Introduction

Speaker verification (SV) aims to verify a client's identity claim via his spoken segments. Real applications in security systems such as access control demands the system robust enough against various environments.

A major component of a standard speaker verification system is speaker modeling. Gaussian Mixture Model-Universal Background Model (GMM-UBM) proposed in [1] had been the de facto approach before *i*-vector [2]. The *i*-vector system projects the GMM super-vector to a more compact and speaker discriminative embedding space. Since then, speaker embeddings have been the major form of speaker modeling. When deep learning achieves incredible performance improvement in speech processing [3, 4, 5], the speaker verification community also started to leverage this powerful tool. Researchers tried to use Deep Neural Networks (DNN) to extract frame-level [6, 7, 8] or utterance-level speaker embeddings [9, 10, 11]. In recent two years, utterance-level deep speaker embeddings such as *x*-vector [9, 12] have achieved impressive results on many standard datasets and evaluations such as NIST SRE 2016.

Although the SV system has been improved a lot during the last decade, it still suffers from a substantial performance degradation due to the complex environment in real applications. To improve the system's robustness and generalization capability, data augmentation is usually adopted because of its simplicity and effectiveness. Basically, data augmentation generates more training samples by simple transformation (adding noises) on the origin data. Such a method has been applied in image classification [13] and speech recognition [14, 15, 16] to increase the amount and diversity of the training data. For speaker verification, Snyder et al. in [12] manually employ additive noises and reverberation to "clean" speech segments and extract "noisy" embeddings to train a more robust PLDA for both *i*-vector and *x*-vector, which can boost the performance. In our previous work [17], a generative adversarial network [18] (GAN) is used to do data augmentation directly on *x*-vector and further improve the performance for the *x*-vector system.

Besides GAN, variational autoencoder [19] (VAE) is another elegant deep generative model, which is adopted in image processing [20, 21] and speech processing [14, 22]. The basic form, or a vanilla VAE is similar to an autoencoder, while the distribution of the latent variable is restricted to follow the normal distribution. To make the generation process on VAE controllable, the conditional VAE [23, 24] (CVAE) is proposed to generate samples based on specific given knowledge.

In this work, a CVAE based embedding augmentation approach is proposed to boost the SV systems' performance. Specifically, a CVAE model is designed to learn the pattern of embeddings extracted from manually augmented speech segments and generate more embeddings with more diversity. The generated embeddings will be used to train a more robust PLDA. Experiments are carried out on the standard NISE SRE 2016 dataset with two typical speaker embeddings, *i*-vector and *x*-vector. The results in both setups exhibit the superiority of our proposed method compared to the baseline systems.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the *i*-vector and *x*-vector based SV systems. VAE and the CVAE variant will be presented in Section 3. Related work and our proposed CVAE based data augmentation method are described in Section 4. We present the experimental results in Section 5 and conclude the paper in Section 6.

2. Embedding based Speaker Verification

Embedding based speaker modeling has been the dominating approach in speaker verification in recent years, while PLDA is the most used back-end. In this section a brief introduction on the embedding/PLDA based SV framework will be given. Specifically, the classical factor analysis based *i*-vector and deep neural network based *x*-vector will be focused on.

2.1. *i*-vector

To compensate for the non-speaker information in the GMM super-vector space, Joint Factor Analysis (JFA) framework [25] was proposed and models speaker and channel factors in separate sub-spaces. *i*-vector simplifies the JFA framework by

[†] Yanmin Qian and Kai Yu are the corresponding authors

http://dx.doi.org/10.21437/Interspeech.2019-2248

modeling a single total variability subspace [2]. In the *i*-vector framework, the speaker- and session-dependent super-vector $\mathbf{M}$ (derived from UBM) is modeled as

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{x} + \epsilon \tag{1}$$

where $\mathbf{m}$ is a speaker and session-independent super-vector, $\mathbf{T}$ is a low rank matrix which captures speaker and session variability, $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I})$, is a multivariate random variable, *i*-vector is the posterior mean of $\mathbf{x}$. $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, is the residual noise term to account for the variability not captured by $\mathbf{T}$.

2.2. *x*-vector

Deep neural network (DNN) has been heavily investigated for speaker embedding learning in recent two years, different architectures [26, 27] and different loss functions [9, 28, 29] have been investigated heavily. *x*-vector [9, 12] is a typical one and used by many researchers. In the *x*-vector framework, a time-delay neural network (TDNN) is trained for speaker embedding learning. The neural network receives frame-level spectral features as input, after several frame-level layers, a statistics pooling layer computes the mean and standard covariance of deep features from one utterance and output utterance-level features to the following layers. One or more embedding layers can be incorporated in the utterance-level layers to extract speaker embeddings. The extracted embeddings are then scored with PLDA. In our experiments, the *x*-vector extractor is trained following the standard Kaldi recipe [30].

2.3. Probabilistic Linear Discriminant Analysis

Probabilistic Linear Discriminant Analysis (PLDA) is widely used as a compensation method and scoring back-end for *i*-vector or *x*-vector based speaker verification [2, 9]. PLDA has several variants which are usually used in the speaker verification task [31], including the standard PLDA [32], two-cov PLDA [33, 34], heavy-tailed PLDA [35] and Simplified PLDA [34]. In this work, we follow the settings in [9] and use the two-cov variant [33] implemented in Kaldi [30] as the scoring back-end. In the two-cov PLDA, the j-th *i*-vector $\mathbf{x}_j^{(s)}$ from the s-th speaker is assumed to be generated as,

$$\mathbf{x}_j^{(s)} \sim \mathcal{N}(\mathbf{y}^{(s)}, \mathbf{W}^{-1}) \tag{2}$$

$$\mathbf{y}^{(s)} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{B}^{-1}) \tag{3}$$

where the within-speaker and between-speaker covariance matrices are $\mathbf{W}^{-1}$ and $\mathbf{B}^{-1}$. $\boldsymbol{\mu}$ is the global mean and $\mathbf{y}^{(s)}$ represents the speaker vector of s-th speaker, a mean vector of all *i*-vector $x_j^{(s)}$ from the s-th speaker. To deal with the mismatch between different domains, many domain adaptation methods for PLDA have been proposed [36, 7, 37]. In the NIST SRE 2016 evaluation condition, to utilize the provided unlabeled in-domain data, the simple version of unsupervised adaption of PLDA implemented in Kaldi [30] is used. The basic idea is to take the unlabeled *i*-vectors from the target domain and use their mean and variance to adapt the PLDA matrices, implementation details can be referred to in "ivector-adapt-plda.cc" in Kaldi.

3. Variational Autoencoder

3.1. Vanilla Variational Autoencoder

A variational autoencoder [19] (VAE) is a probabilistic generative model containing two parts, an encoder and a decoder. It assumes that data $\mathbf{x}$ from dataset $\mathbf{X}$ are generated by some random process, involved by the random latent variable $\mathbf{z}$. In that random process, a value $\mathbf{z}$ is firstly sampled from the normal distribution, and then a value $\mathbf{x}$ is generated from a conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$ parameterized by θ:

$$\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}) \tag{4}$$

$$\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z}) \tag{5}$$

Generally, $p_\theta(\mathbf{x}|\mathbf{z})$ is considered as a decoder or generator. As the integral of the marginal likelihood $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{z}) p_\theta(\mathbf{x}|\mathbf{z}) d\mathbf{z}$ is intractable, a recognition model $q_\phi(\mathbf{z}|\mathbf{x})$ with parameter ϕ is introduced to approximate the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$. Similar to the decoder, the recognition model is also represented by a neural network, which is considered as an encoder. Therefore, the lower bound of the marginal likelihood can be written as:

$$\begin{aligned} \text{log} p_\theta(\mathbf{x}) &\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[-\log q_\phi(\mathbf{z}|\mathbf{x}) + \log p_\theta(\mathbf{x}, \mathbf{z})] \\ &= -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \end{aligned} \tag{6}$$

where $p_\theta(\mathbf{z})$ is the prior distribution of latent variables, i.e. $\mathcal{N}(0, \mathbf{I})$, and D_{KL} is the Kullback-Leibler (KL) divergence. Maximizing the lower bound (eq. (6)) of the marginal likelihood with reparameterization trick [19] and Stochastic Gradient Variational Bayes [19] (SGVB) estimator, a VAE model can learn how to generate data given a random latent variable $\mathbf{z}$ sampled from the normal distribution.

3.2. Conditional VAE

As mentioned in Section 1, to make the generation process more controllable, Sohn et al. in [23] proposed Conditional VAE (CVAE) based on the vanilla VAE. In addition to the latent variable $\mathbf{z}$, a CVAE generates data $\mathbf{x} \in \mathbb{R}^n$ with some given condition $\mathbf{c}$. And the goal of CVAE is to maximize the log-likelihood of $\mathbf{x}$ given $\mathbf{c}$, whose lower bound can be written as:

$$\begin{aligned} \log p_\theta(\mathbf{x}|\mathbf{c}) \geq& -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{c}) || p_\theta(\mathbf{z}|\mathbf{c})) \\ &+ \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{c})}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c})] \end{aligned} \tag{7}$$

Assume that the $\mathbf{z}$ is independent of $\mathbf{c}$ when $\mathbf{x}$ is unknown [24], the term $p_\theta(\mathbf{z}|\mathbf{c})$ in the lower bound (eq. (7)) can be replaced by the normal distribution $\mathcal{N}(0, \mathbf{I})$.

4. VAE based Data Augmentation for Speaker Embedding

In [9], the authors manually employ additive noises and reverberation to the existing "clean" speech segments in training set, and a robust PLDA is trained with the embeddings extracted from the "clean" and "noisy" utterances. Our previous work [17] adopted a conditional GAN to directly do the data augmentation in the *x*-vector space, which further boosts the *x*-vector/PLDA system's performance. In this paper, a CVAE model is used in a similar way for embedding augmentation and achieve further performance gain for both *x*-vector and *i*-vector systems.

The architecture of the proposed model is shown in Figure 1. A detailed description on the training and generation processes will be given in the following subsections.

4.1. Training

The goal of our model is to maximize the log-likelihood $\log p_\theta(\hat{\mathbf{x}}_u^{(s)}|\mathbf{y}^{(s)})$ of the noisy embedding $\hat{\mathbf{x}}_u^{(s)}$ for the u-th utterance from the s-th speaker, given the corresponding speaker

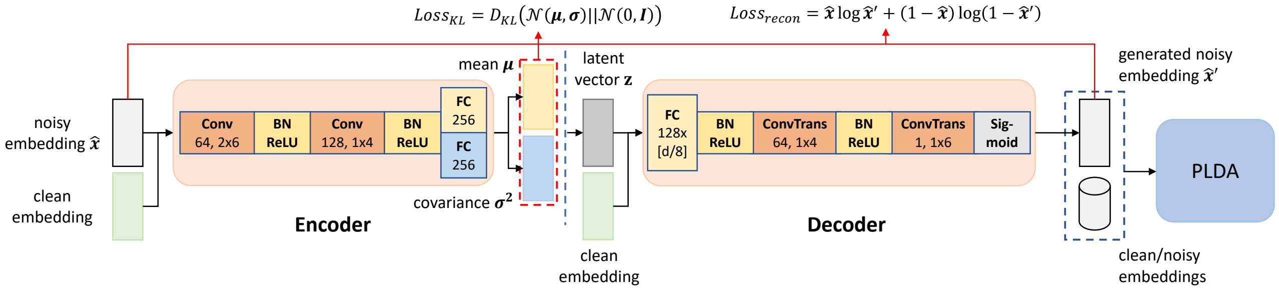
$Loss_{KL} = D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})||\mathcal{N}(0, \boldsymbol{I}))$
$Loss_{recon} = \hat{\mathbf{x}} \log \hat{\mathbf{x}}' + (1 - \hat{\mathbf{x}}) \log(1 - \hat{\mathbf{x}}')$
noisy embedding $\hat{\mathbf{x}}$
clean embedding
Conv 64, 2x6
BN ReLU
Conv 128, 1x4
BN ReLU
FC 256
FC 256
mean $\boldsymbol{\mu}$
covariance $\boldsymbol{\sigma}^2$
latent vector $\mathbf{z}$
Encoder
clean embedding
FC 128x [d/8]
BN ReLU
ConvTrans 64, 1x4
BN ReLU
ConvTrans 1, 1x6
Sig-moid
Decoder
generated noisy embedding $\hat{\mathbf{x}}'$
clean/noisy embeddings
PLDA

Figure 1: *Framework and detailed neural network configuration of the proposed CVAE data augmentation for embedding based speaker verification system.* ***FC*** *means fully connected layer,* ***BN*** *means batch normalization,* ***Conv*** *means convolutional layer,* ***ConvTrans*** *means transposed convolutional layer [38]. The numbers under the layers indicate model configuration, e.g.* $[64, 2 \times 6]$ *means the layer has* 64 *output channels and a kernel size of* 2×6*. The noisy embeddings and clean embeddings are all d-dimensional.*

embedding $\mathbf{y}^{(s)}$, a mean vector of all clean embeddings $\mathbf{x}_u^{(s)}$ from the s-th speaker. With the speaker embeddings as condition, the model no longer needs to learn the distribution of the entire embeddings. Instead, it models the patterns of various noises and reverberation, using $\mathbf{z}$ as a high-level representation of those patterns. As mentioned in section 3, since the type of noises and reverberation is independent with $\mathbf{y}^{(s)}$ when noisy embedding $\hat{\mathbf{x}}_u^{(s)}$ is unknown, the distribution $p_\theta(\mathbf{z}|\mathbf{y}^{(s)}) = \mathcal{N}(0, \mathbf{I})$. Derived from the lower bound (eq. (7)), the model can be trained by minimizing the loss function below:

$$D_{KL}(q_\phi(\mathbf{z}|\hat{\mathbf{x}}, \mathbf{y})||\mathcal{N}(0, \mathbf{I})) + BCELoss(\hat{\mathbf{x}}_u^{(s)}, \hat{\mathbf{x}}_u'^{(s)}) \quad (8)$$

where $\hat{\mathbf{x}}_u^{(s)}, \mathbf{y}^{(s)} \in [0, 1]^d$. $\hat{\mathbf{x}}_u'^{(s)}$ is the generated noisy embedding and $BCELoss(x, x') = x \log x' + (1 - x) \log(1 - x')$. The second term in formula (eq. (8)) is also regarded as a reconstruction loss, which is the binary cross entropy between the input and the generated noisy embeddings. By restricting the distribution of representation $\mathbf{z}$ to normal distribution, the Gaussian space is divided into different parts for the additive noise and reverberation. Therefore, sampling vectors from normal distribution, we can obtain the representation of some nonlinear combination of those attributes.

4.2. Generation

At the generation stage, the encoder of the CVAE model is discarded. We sample multiple $\mathbf{z}$ from normal distribution $\mathcal{N}(0, \mathbf{I})$ and feed them into the trained decoder from the CVAE model. In addition, a clean utterance embedding $\mathbf{x}_u^{(s)}$ is given for each $\mathbf{z}$ as a condition. The noisy embeddings $\hat{\mathbf{x}}'^{(s)}$ generated by the decoder are used as additional data to train a more robust PLDA model.

4.3. Data Augmentation for Adapted PLDA

As mentioned in Section 2, a small amount of unlabeled in-domain embeddings can boost the performance of speaker verification system by applying an unsupervised adaptation to the PLDA back-end. With the data augmentation model above, we can generate some noisy in-domain embeddings from the small set of clean ones. Here, the best way to augment the in-domain data is to train another data augmentation model on that domain. However the in-domain data is limited and unlabeled, so we use the same CVAE model above for simplicity. By combining these clean and noisy in-domain embeddings, the diversity increases. And then we can adapt the PLDA with the combination set and improve the stability of the adaptation process.

5. Experiments

5.1. Datasets

Following the standard settings in the previous work, the training data consists of the SWBD portion and SRE portion, while the former includes Switchboard phases 2,3 and Switchboard Cellular 1,2, and the latter contains the NIST SRE 2004-2010. The *i*-vector and *x*-vector extractors are trained on the SWBD and SRE pooled data, while the PLDA and the VAE model are trained only on the SRE portion. The standard SRE16 evaluation set is used to measure the performance of the proposed system, which consists of Tagalog and Cantonese subsets. The enroll utterances have 60-second length while the test utterances vary from 10-60 seconds. The manual augmentation procedure follows the Kaldi SRE16 recipe [30], additive noises and reverberation are added to the original audios to generate the noisy ones. Details can be referred to in [12]. The number of embeddings to train the VAE model is roughly 90,000, both for *i*-vector and *x*-vector based systems. As a common trick for VAE and GAN training, all the embeddings are scaled to $[0, 1]$.

5.2. *i*-vector and *x*-vector Extractors

The settings of *i*-vector and *x*-vector follow the Kaldi SRE16 recipes v1 and v2, respectively. For the *i*-vector, 20 dimensional MFCCs with delta and double-delta coefficients appended form the 60-dimension input for the system, 2,048 Gaussian components are used for the UBM training and the dimension of *i*-vector is set to 600. For the *x*-vector system, a TDNN is trained on the 30-dimension MFCCs and 512-dimension *x*-vectors are extracted.

5.3. Implementation Details of the Augmentation via CVAE

The CVAE framework mentioned in section 4 is used in the experiments. The model learns to generate data with more diversity from the manually augmented noisy data. The detailed neural network configuration is also shown in Figure 1.

The encoder network consists of two convolutional layers and two fully connected layers, while the decoder network consists of two transposed convolutional layers. 256-dimensional mean vectors $\boldsymbol{\mu}$ and variance vectors $\boldsymbol{\sigma}^2$ are predicted by the encoder network, which is used for reparameterization and computing the KL-Divergence as mentioned in Section 3. Speaker embeddings are fed to both the encoder and decoder networks. A sigmoid function is applied to the output to restrict the generated samples in $[0, 1]$. To stabilize the training process, we also use batch normalization and leaky ReLU with 0.2 negative slope in both the encoder and decoder networks.

Adam optimizer with a learning rate of $3\mathrm{e}{-5}$ and default betas, $(0.9, 0.999)$ is applied for optimizing both the encoder and the decoder networks. With a batch size of 128, we train the model on a single GPU for 10 epochs.

To augment the existing embeddings, we generate 10 noisy embeddings for each speaker with his/her clean embeddings and randomly sampled latent variables $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, providing totally 42,500 augmented noisy embeddings, which is about half of the manually augmented data.

5.4. Results and Analysis

As mentioned in previous sections, the proposed data augmentation approach is evaluated on two typical kinds of embeddings, *i*-vector and *x*-vector. Equal error rate (EER) and the minimum of the normalized detection cost function (minDCF, p_{target} set as 0.01) are used for performance measurement.

Table 1: *Performance comparison for i-vector/PLDA SV system using different data augmentation methods. The amount of augmented data for different methods are comparable.*

	Data	SRE16 Tagalog		SRE16 Cantonese	
	Augmentation	EER (%)	minDCF	EER (%)	minDCF
PLDA	none	18.13	0.7068	9.82	0.3951
+Adaptation		17.84	0.6338	8.82	0.3591
PLDA	manual	17.63	0.6961	9.42	0.3827
+Adaptation		16.94	0.6105	8.30	0.3411
PLDA	VAE	17.45	0.7185	10.14	0.4088
+Adaptation		15.83	0.5981	8.32	0.3461
PLDA	VAE & manual	17.20	0.7106	9.62	0.3940
+Adaptation		**15.54**	**0.5897**	**7.84**	**0.3331**

Table 1 shows the results of different augmentation methods based on the *i*-vector/PLDA based SV system. The VAE augmented system achieves comparable performance with the manually augmented system on both Tagalog and Cantonese evaluation datasets, while further performance enhancement could be obtained by combining the VAE and manually augmented data. The best performance is achieved by the proposed VAE augmentation approach with PLDA adaptation based on the *i*-vector system, obtaining EER of 15.54% and 7.84% for Tagalog and Cantonese, respectively.

Table 2: *Performance comparison of different data augmentation methods for x-vector/PLDA based SV system. Results of GAN based data augmentation are from our previous work [17]*

	Data	SRE16 Tagalog		SRE16 Cantonese	
	Augmentation	EER (%)	minDCF	EER (%)	minDCF
PLDA	none	16.63	0.7121	7.57	0.3451
+Adaptation		14.10	0.5420	5.77	0.2523
PLDA	manual	16.16	0.7248	7.45	0.3368
+Adaptation		12.79	0.5144	5.26	0.2357
PLDA	GAN	16.54	0.7004	7.09	0.3363
+Adaptation		12.42	0.5196	4.66	0.2379
PLDA	GAN & manual	16.59	0.7182	6.85	0.3256
+Adaptation		**11.68**	0.4886	4.43	0.2160
PLDA	VAE	16.44	0.7150	6.705	0.3187
+Adaptation		12.04	0.4844	4.29	0.2051
PLDA	VAE & manual	16.13	0.7114	6.60	0.3082
+Adaptation		11.86	**0.4799**	**4.20**	**0.2032**

For the *x*-vector based systems, the results could be found in Table 2. Compared with the baseline system without data augmentation, all systems trained with augmented data achieve better performance. Moreover, for all the systems, PLDA adaptation consistently improves the performance. It could be noted that our previous GAN system achieves comparable results with the manually augmented system, while the GAN + manual system outperforms the manual one on both Tagalog and Cantonese evaluation set. The pure VAE system outperforms the manual system, while the VAE + manual system further enhances the system, achieving the best results. With the proposed augmentation method, we finally achieve 11.86% and 4.20% EER on Tagalog and Cantonese, respectively.

To give a more intuitive illustration of our proposed VAE data augmentation method, the detection error trade-off (DET) curves of different augmented *x*-vector systems are plotted in Figure 2 (Cantonese). It could be observed that the proposed methods are effective for both non-adapted and adapted PLDA.

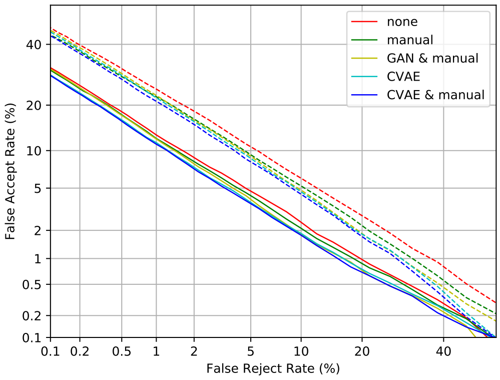

Figure 2: *Detection error trade-off (DET) graph of different data augmentation methods on Cantonese for x-vector based system with PLDA. The dotted and concrete lines represent the non-adapted and adapted PLDA systems respectively. (Better viewed in color)*

6. Conclusions

Speaker embeddings are the dominating modeling methods for speaker verification. Despite the impressive performance by embeddings such as *i*-vector and *x*-vector, the system robustness in different application scenarios is still a problem. In this paper, we proposed to directly perform the data augmentation at the embedding level based on a conditional variational autoencoder. The CVAE-based approach outperforms the manually data augmentation approach and our previous GAN-based data augmentation method on the standard NIST SRE16 evaluation dataset. Combining with manually augmented embeddings, the performance was further boosted, where the best system achieved the EERs of 15.54% and 7.84% with *i*-vector/PLDA based systems and 11.86% and 4.20% with *x*-vector/PLDA based speaker verification for Tagalog and Cantonese, respectively.

7. Acknowledgements

This work has been supported by the China NSFC projects (No. 61603252 and No. U1736202). Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

8. References

[1] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[2] 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.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[5] Z. Chen and K. Yu, "An investigation of implementation and performance analysis of dnn based speech synthesis system," in *Signal Processing (ICSP), 2014 12th International Conference on*. IEEE, 2014, pp. 577–582.

[6] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[7] F. S. Richardson, D. A. Reynolds, and B. Nemsick, "Channel compensation for speaker recognition using map adapted plda and denoising dnns," MIT Lincoln Laboratory Lexington United States, Tech. Rep., 2016.

[8] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014*. IEEE, 2014, pp. 4052–4056.

[9] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *Proc. Interspeech 2017*, pp. 999–1003, 2017.

[10] Z. Huang, S. Wang, and Y. Qian, "Joint i-vector with end-to-end system for short duration text-independent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018*. IEEE, 2018.

[11] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification," in *Proc. InterSpeech*, September 2018.

[12] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018*. IEEE, 2018.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[14] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 16–23.

[15] H. Hu, T. Tan, and Y. Qian, "Generative adversarial networks based data augmentation for noise robust speech recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5044–5048, 2018.

[16] H. T. T. Peiyao Sheng, Zhuolin Yang and Y. Qian, "Data augmentation using conditional generative adversarial networks for robust speech recognition," in *The 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, November 2018.

[17] Y. Yang, S. Wang, M. Sun, Y. Qian, and K. Yu, "Generative adversarial networks based x-vector augmentation for robust probabilistic linear discriminant analysis in speaker verification," in *The 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, November 2018.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[19] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2014.

[20] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taïga, F. Visin, D. Vázquez, and A. C. Courville, "Pixelvae: A latent variable model for natural images," *CoRR*, vol. abs/1611.05013, 2017.

[21] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, "Variational autoencoder for deep learning of images, labels and captions," in *Advances in neural information processing systems*, 2016, pp. 2352–2360.

[22] J. Villalba, N. Brümmer, and N. Dehak, "Tied variational autoencoder backends for i-vector speaker recognition," in *INTERSPEECH*, 2017.

[23] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 3483–3491.

[24] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *ECCV*, 2016.

[25] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *CRIM, Montreal,(Report) CRIM-06/08-13*, 2005.

[26] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[27] N. Li, D. Tuo, D. Su, Z. Li, D. Yu, and A. Tencent, "Deep discriminative embeddings for duration robust speaker verification," *Proc. Interspeech 2018*, pp. 2262–2266, 2018.

[28] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances." in *Interspeech*, 2017, pp. 1487–1491.

[29] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification," *Proc. Interspeech, Hyderabad*, 2018.

[30] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[31] A. Sizov, K. A. Lee, and T. Kinnunen, "Unifying probabilistic linear discriminant analysis variants in biometric authentication," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 2014, pp. 464–475.

[32] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[33] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

[34] N. Brümmer and E. De Villiers, "The speaker partitioning problem." in *Odyssey*, 2010, p. 34.

[35] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." in *Odyssey*, 2010, p. 14.

[36] J. Villalba and E. Lleida, "Unsupervised adaptation of plda by using variational bayes methods," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 744–748.

[37] D. Garcia-Romero, A. McCree, S. Shum, N. Brummer, and C. Vaquero, "Unsupervised domain adaptation for i-vector speaker recognition," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2014.

[38] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 2528–2535.