Data augmentation with moment-matching networks for i-vector based speaker verification

Sayaka Shiota*, Shinnosuke Takamichi† and Tomoko Matsui‡
* Tokyo Metropolitan University, Faculty of System Design, Japan
† Graduate School of Information Science and Technology, The University of Tokyo, Japan
‡ The Institue of Statistical and Mathematics, Japan

***Abstract*—This paper proposes an i-vector generation scheme with conditional generative moment-matching networks (MMNs) for speaker verification. In this scheme, multiple i-vectors for each enrollment speaker are randomly generated from trained MMNs and noise distributions. The randomly generated i-vectors are assumed to represent diverse variations for each enrollment speaker. Since this paper is aim to provide new possibility of the i-vector augmentation with MMNs, i-vector-based preliminary speaker verification evaluation with support vector machine (SVM) are performed. The results of SVM classification show that the generated i-vectors are contributed for estimation of the accurate SVM classifiers of enrollment speakers. From the experimental results, we also compare the distributions of the generated i-vectors with those of the original ones and discuss them.**

I. Introduction

Speaker verification (SV), which offers a natural and flexible system for biometric authentication, has been actively studied in the past decades [1]–[4]. One of the important problems for SV systems is to represent diverse variations such as speaking styles, emotions, for each enrollment speaker. The other one is to estimate reliable speaker models by using insufficient amount of utterances. The state-of-the-art SV systems are mostly based on or compared with i-vector-based framework [2], which is a de-facto standard technique. However, the i-vector-based methods are also suffered from the problem about the amount of enrollment data for training speaker dependent models. By now, in order to avoid the problems and improve its robustness, several variability compensation techniques such as Within-Class Covariance Normalization (WCCN) [5], Linear Discriminant Analysis (LDA), Nuisance Attribute Projection (NAP) [6], and Gaussian Probabilistic LDA (GPLDA) [7] have been proposed as the back-end of the i-vector-based framework. However, the robustness problem of the i-vector representation has not been completely solved. For example, the i-vector scatter diagram via the utterance length variation as shown in [8] renders that it is difficult to find the relation between the i-vector distribution and the utterance lengths. This indicates that the conventional methods based on the i-vector framework cannot deal with the utterance length variation properly, and so sufficient performance has not been obtained.

Meanwhile, deep learning (DL) approaches have proven to be useful for a wide range of speech problems, and recently started to be used also for speaker recognition [4], [9]–[11]. It is known that SV systems are required to work with few utterances which have relatively short lengths both in training and evaluation [12]. In the well-known NIST Speaker Recognition Evaluation (NIST SRE) series [13], [14], such a condition with few utterances has been discussed. Since DL approaches with deep neural networks (DNNs) usually need a large amount of training data, it is not straightforward to utilize DNNs for SV problems.

In speech synthesis area, many techniques with DNNs have been reported and the quality of synthesized speeches remarkably has been improved [15]–[17]. On the other hand, the techniques are required not only naturalness but also flexibility for utterance variation in speaking styles for each speaker. To obtain the flexibility, the sampling-based speech parameter generation by using moment-matching networks (MMNs) has been proposed [18]. In this method, speech parameters are randomly sampled from noise distributions and the synthesized speech can represent diverse variation. It is possible to synthesize various speech close to specific speaker's one. Therefore, the use of synthetic speech can be considered to increment the number of training utterances for each speaker in i-vector based SV systems. However, it is not guaranteed that an i-vector derived from the synthesized speech is close to the speaker's one, that is, the distribution of the former i-vector is similar to that of the latter i-vector.

In this paper, we investigate a scheme of i-vector augmentation by conditional generative MMNs to present diverse i-vector variations. In this scheme, i-vectors are directly generated to minimize the objective function of conditional maximum mean discrepancy (MMD) so that the distribution of the generated i-vectors is similar to that of the original speaker. The proposed scheme is evaluated by means of a speaker verification system based on i-vectors for representation and SVMs for classification. An analysis of the distribution of the generated i-vectors, performed using low-dimensional spaces, is presented.

The outline of this paper is as follows. Section 2 reviews the i-vector paradigm and related topics, and the framework of the MMNs is illustrated in section 3. Section 4 and 5 present the experimental results and conclusions, respectively.

II. I-vector extraction

An i-vector is a fixed low dimensional representation of a speech utterance that preserves the speaker-specific in-

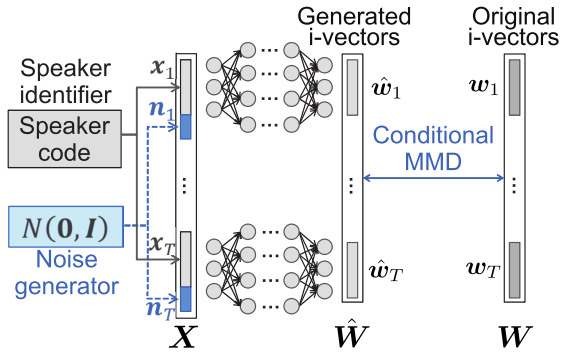

Fig. 1. Sampling-based i-vector generation using moment-matching networks.

formation [2]. In the i-vector paradigm, a speaker-specific Gaussian mixture model (GMM) mean supervector $\boldsymbol{M}_s$ can be represented in terms of a speaker and channel independent supervector $\boldsymbol{m}$, a low rank total variability matrix $\boldsymbol{T}$, and a vector $\boldsymbol{w}_s$ as

$$\boldsymbol{M_s} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{w_s}, \tag{1}$$

where s indicates a speaker index. In Eq. 1, $\boldsymbol{w}_s$ is called i-vector and the $\boldsymbol{T}$ matrix is learned using a large amount of training data. An i-vector of an utterance represents its coordinates in the total variability space (i.e. space spanned by the columns of $\boldsymbol{T}$), extracted as the maximum a posteriori (MAP) point estimates of $\boldsymbol{w}_s$ given the utterance. In the back-end process, each extracted i-vectors $\boldsymbol{w}_s$ from all the training utterances is used as a feature vector of support vector machine (SVM) for a classification scheme.

III. Data augmentation with moment-matching networks

A. Deep generative models

A deep generative model is a generative model with DNNs, and the well-known examples are generative adversarial networks (GANs) [19] and generative moment-matching networks (MMNs) [20]. The DNNs that can randomly generate data samples are trained to represent the training data distribution. Input fed to the DNNs is a low-dimensional noise vector that is randomly generated from a prior probabilistic distribution (e.g., Gaussian or uniform distribution), and the trained DNNs work to transform the prior noise distribution to the training data distribution. The data samples are randomly generated via sampling from the prior noise distribution.

GANs are trained with a minimax optimization technique. It is known that the optimization requires tricky implementation [21] and its generation accuracy is difficult to evaluate. On the other hand, generative MMNs are in an easy-to-optimize minimization problem. Also, the networks can be extended to the conditional generative MMNs [22] (Section III-B2) conditioned by the preferred information (e.g., speaker information). The generative MMNs can model unconditional distribution, and the conditional ones can model conditional distribution.

GAN-based data augmentation approaches were proposed in image recognition [23] and audio event detection [24]. GANs that represent complicated data distribution help to generate realistic data newly and to determine boundaries of classification robustly. Its application of the GAN-based data augmentation to speaker verification is straightforward. However, the augmentation performance is strongly affected by the GAN accuracies. Therefore, in this paper, we adopt generative MMNs and investigate the relationship between the data augmentation performance and the training accuracy. In the following sections, we introduce general frameworks of the generative MMNs and propose data augmentation with the networks.

B. Generative moment-matching networks

1) Minimizing maximum mean discrepancy (MMD): Let $\boldsymbol{W} = \{\boldsymbol{w}_t\}_{t=1}^T$ be a training data set. $\boldsymbol{w}_t$ is the t-th feature vector and T is the total number of the training data. A set of low-dimensional noise vectors $\boldsymbol{N} = \{\boldsymbol{n}_t\}_{t=1}^T$, which is fed to DNNs $\boldsymbol{G}(\cdot)$, is randomly sampled from a standard multivariate Gaussian distribution. The dimensionality of $\boldsymbol{n}_t$ is often smaller than that of $\boldsymbol{w}_t$. Let $\hat{\boldsymbol{W}} = \{\hat{\boldsymbol{w}}_t\}_{t=1}^T$ be a set of data generated from DNNs, i.e., $\hat{\boldsymbol{w}}_t = \boldsymbol{G}(\boldsymbol{n}_t)$. $\boldsymbol{G}(\cdot)$ is trained to minimize squared error between moments of $\boldsymbol{W}$ and those of $\hat{\boldsymbol{W}}$. Using a kernel trick, the criterion is represented using gram matrices, and is known as the square of kernelized Maximum Mean Discrepancy (MMD) [20] as follows:

$$\begin{aligned} L_{\mathrm{MMD}} =& \frac{1}{T^2}\Big\{\mathrm{tr}\left(\boldsymbol{1}_T \boldsymbol{K}_{\boldsymbol{W},\boldsymbol{W}}\right) + \mathrm{tr}\left(\boldsymbol{1}_T \boldsymbol{K}_{\hat{\boldsymbol{W}},\hat{\boldsymbol{W}}}\right) \\ & -2\mathrm{tr}\left(\boldsymbol{1}_T \boldsymbol{K}_{\boldsymbol{W},\hat{\boldsymbol{W}}}\right)\Big\}, \end{aligned} \tag{2}$$

where $\mathrm{tr}(\cdot)$ denotes matrix trace. $\boldsymbol{1}_T$ is a T-by-T matrix whose all components are 1. $\boldsymbol{K}_{\boldsymbol{W},\hat{\boldsymbol{W}}}$ is a T-by-T gram matrix between $\boldsymbol{W}$ and $\hat{\boldsymbol{W}}$. $\{t,\tau\}$-th component of $\boldsymbol{K}_{\boldsymbol{W},\hat{\boldsymbol{W}}}$ is an arbitrary kernel function between $\boldsymbol{w}_t$ and $\hat{\boldsymbol{y}}_\tau$. 0th-through-infinite order moments are considered in training if the Gaussian kernel is chosen.

2) Minimizing conditional MMD: The conditional generative MMNs [22] are trained in the same manner as in Section III-B1. Let $\boldsymbol{S} = \{\boldsymbol{s}_t\}_{t=1}^T$ be a set of *conditioning vectors*, where $\boldsymbol{s}_t$ is a *conditioning vector* corresponding to $\boldsymbol{w}_t$. Here, the conditioning vector means given parameters for a conditional distribution. A set of concatenated vectors, $\tilde{\boldsymbol{S}} = \{\tilde{\boldsymbol{s}}_t\}_{t=1}^T$, is used for conditioning $\boldsymbol{G}(\cdot)$. $\tilde{\boldsymbol{s}}_t = \left[\boldsymbol{s}_t^\top, \boldsymbol{n}_t^\top\right]^\top$ is fed to $\boldsymbol{G}(\cdot)$ and $\hat{\boldsymbol{w}}_t$ is given as $\boldsymbol{G}(\tilde{\boldsymbol{x}}_t)$. The training criterion to be minimized is

$$\begin{aligned} L_{\mathrm{CMMD}} =& \frac{1}{T^2}\Big\{\mathrm{tr}\left(\boldsymbol{L}_{\tilde{\boldsymbol{S}}} \boldsymbol{K}_{\boldsymbol{W},\boldsymbol{W}}\right) + \mathrm{tr}\left(\boldsymbol{L}_{\tilde{\boldsymbol{S}}} \boldsymbol{K}_{\hat{\boldsymbol{W}},\hat{\boldsymbol{W}}}\right) \\ & -2\mathrm{tr}\left(\boldsymbol{L}_{\tilde{\boldsymbol{S}}} \boldsymbol{K}_{\boldsymbol{W},\hat{\boldsymbol{W}}}\right)\Big\}, \end{aligned} \tag{3}$$

$$\boldsymbol{L}_{\tilde{\boldsymbol{S}}} = (\boldsymbol{K}_{\tilde{\boldsymbol{S}}} + \lambda \boldsymbol{I}_T)^{-1} \boldsymbol{K}_{\tilde{\boldsymbol{S}}} (\boldsymbol{K}_{\tilde{\boldsymbol{S}}} + \lambda \boldsymbol{I}_T)^{-1}, \tag{4}$$

where $\boldsymbol{I}_T$ is the T-by-T identity matrix, λ is the regularization coefficient, and $\boldsymbol{K}_{\tilde{\boldsymbol{S}}}$ is a gram matrix of $\tilde{\boldsymbol{S}}$, where $\{t,\tau\}$-th component of $\boldsymbol{K}_{\tilde{\boldsymbol{S}}}$ is an arbitrary kernel function of $\boldsymbol{s}_t$ between $\boldsymbol{x}_\tau$. Note that the kernel function of $\boldsymbol{K}_{\tilde{\boldsymbol{S}}}$ is different from that

of $\boldsymbol{K}_{\boldsymbol{W},\hat{\boldsymbol{W}}}$. When $L_{\mathrm{CMMD}} = 0$, the conditional distribution of $\hat{\boldsymbol{w}}_t$ given $\boldsymbol{s}_t$ is completely match with that of $\boldsymbol{w}_t$ given $\boldsymbol{s}_t$.

C. Data augmentation with generative moment-matching networks

We apply the conditional generative MMNs to data augmentation for i-vector-based SV. Figure 1 shows the diagram of the MMNs for i-vectors. $\boldsymbol{s}_t$ and $\boldsymbol{w}_t$ are the speaker vector (i.e., a vector representing speaker information) and corresponding i-vector of the s-th speaker. $\boldsymbol{s}_t$ is given as a one-hot vector in this paper. Namely, the dimensionality of $\boldsymbol{s}_t$ is the same to the number of pre-stored speakers, and the s-th component of $\boldsymbol{s}_t$ takes 1 and otherwise are 0.

In training, mini-batch learning method can be adopted. A set of speaker vectors $\boldsymbol{S}$ and a set of corresponding i-vectors $\boldsymbol{W}$ are randomly selected from a whole training dataset, and $\boldsymbol{N}$ is randomly generated from the prior noise distribution. The model parameters of DNNs are updated by the stochastic gradient descent method. In generation, given a speaker vector $\boldsymbol{s}_t$ of the s-th speaker and randomly generated $\boldsymbol{n}_t$, i-vector for the speaker, $\hat{\boldsymbol{w}}_t$, is randomly generated from the trained DNNs. After generating a number of i-vectors for all pre-stored speakers, the i-vectors are used as an augmented training datasets for speaker verification.

The one-hot vector is a discrete representation for the speaker information. Another choice of representations is an averaged i-vectors as a continuous representation. A mean vector of i-vectors is calculated speaker by speaker in advance, and is used as $\boldsymbol{s}_t$. We compared current one-hot vector-based and this i-vector mean-based methods in terms of SV accuracies, and the former one was chosen since it exhibited a better performance.

IV. Experiments

To evaluate the performance of the proposed i-vector augmentation scheme, i-vector-based SV systems with SVMs were conducted. These experiments were regarded as a preliminary investigation for i-vector augmentation scheme with the MMNs.

A. Experimental conditions

Table I shows the experimental conditions of the i-vector-based SV system. In this experiments, since it assumed that the amount of training data for each enrollment speaker was limited, the duration of each data for extracting one i-vector was segmented to about one second. Consequently about 130 i-vectors were prepared for each enrollment speaker. These i-vectors were regarded as original i-vectors. 100 i-vectors were used for enrollment data and 30 i-vectors were used for evaluation data. The baseline system was used the original i-vectors only for estimating the back-end classifier.

To construct feed-forward DNN architectures for i-vector augmentation MMNs, several parameters of DNNs were used, and the details were shown in Table II, where λ was a regularization coefficient at Eq. (4). Thanks to [27], the DNN parameters were optimized. The input of MMNs was a 8-dimensional one-hot-vector[1] and a randomly sampled noise vector per i-vector. Rectified Linear Unit (ReLU) [28] was used as an activation function of hidden layers, and the noise vectors are sampled from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The MMNs were estimated from about 100 original i-vectors per enrollment speaker. The parameter sets were obtained by combining the parameters in Table II and over 700 parameter sets were used to train the MMNs. For each enrollment speaker, 100 i-vectors were sampled from the MMNs estimated from each parameter sets. To calculate conditional MMD, a Gaussian kernel was used as the kernel function for speech parameters $\boldsymbol{y}$, i.e., $\exp\{-\|\boldsymbol{y}_t - \hat{\boldsymbol{y}}_\tau\|^2/\sigma^2\}$. σ was set so that $\|\boldsymbol{y}_t - \hat{\boldsymbol{y}}_\tau\|^2 < 1$ for all the training data [22].

The original and sampled i-vectors were used as feature vectors for linear kernel SVM classification in the back-end of i-vector-based SVM systems. In this experiment, the sampled i-vectors from the MMN of speaker "A" were regarded as the enrollment data of speaker "A." For evaluation, the accuracies, false rejection rate (FRR) and false acceptance rate (FAR) were calculated with the estimated SVM classifier. To investigate the tendencies between the evaluation scores and the MMD value of final epoch in training the MMNs, each parameter set was called with the MMD ranking. For example, "min1" meant that the parameter set obtained the minimum MMD among all parameter sets. "max5" meant that 5 parameter sets which obtained the maximum MMD in the first to fifth were combined into one set. "rand10" represented that 10 parameter

TABLE I
Experimental conditions for i-vector based SV system

Feature extraction	
Sampling rate	16kHz
Frame length	25msec
Frame shift	10msec
Feature vector	19-order MFCCs+Δ +$\Delta\Delta$
UBM, TV matrix	
Database	JNAS [25]
# of sentences	23,657 (female only)
mixture	1,024
i-vector dimension	200
Enrollment and evaluation	
Database	VLD database [26]
# of speakers	8

TABLE II
Parameters of DNN architectures

# of hidden layers	2, 3
# of hidden unit	128, 256, 512
Noise vector dim.	3, 5, 10
Regularization coef. (λ)	0.01, 0.001, 0.0001
Mini-batch size	300, 500
# of epoch	300, 500
Learning rate	0.001, 0.01, 0.05

[1] As explained in Section III-C, one-hot-vector encoding of speaker information had better performance for training data including eight speakers. We expect that another choice of encoding will be required for training data including a larger number of speakers.

TABLE III
FARS AND FRRS OF EACH PARAMETER SET

Param. set	FAR(%)	FRR(%)	Acc(%)
Baseline	3.0	22.0	95.2
min1	4.2	5.4	95.5
min5	**2.5**	**8.3**	**96.7**
min10	2.5	12.0	96.2
min20	6.6	15.0	92.3
min30	3.1	14.5	95.4
max1	14.2	0.0	87.5
max5	11.7	1.6	89.5
max10	8.6	7.5	91.5
max20	2.5	22.0	95.0
max30	2.5	25.4	94.5
rand5	5.5	17.5	92.9
rand10	2.6	15.8	95.6
rand20	2.9	13.3	95.7
rand30	3.3	15.0	95.1
min5+max5	2.8	11.6	96.0
min10+max10	2.7	15.4	95.6
min15+max15	3.2	15.4	95.2

sets were randomly selected.

B. Experimental results

Table III shows FARs, FRRs and accuracies for each parameter set. It mentioned that X of "minX," "maxX" and "randX" was the combination number of parameter sets. For example, "min5" represented that 100 i-vectors were generated with 5 parameter sets, thus, totally 500 generated i-vectors were used for the enrollment speakers. From the table, "min5" achieved the highest accuracy and outperformed the baseline system, and the FARs and FRRs of "minX" tended to obtain lower scores than those of "maxX." The accuracies of "minX" were also increased than those of "maxX." The results showed that the parameter sets which obtained lower MMD values tend to obtain the lower FARs. It indicated that these parameter sets represented the high speaker similarity of enrollment speakers. However, it was noted when the number of combinations was over 10, the evaluation scores tend to be worse. By using the MMNs for i-vector augmentation and adequate parameter sets, the reliable classifier was able to be estimated. Moreover, we used other combinations; "min+max" was combination of "min" and "max" parameter sets. Even though "rand" and "min+max" obtained the lowest values of FARs, their corresponding FRRs were too high. It implied that an incorrect choice of the parameter sets caused SVM performance to degrade.

To discuss the characteristics of generated i-vectors, Fig. 2 plots the values of each i-vector dimension for 150 original and 50 generated i-vectors. The upper and lower figures of Fig. 2 show the case of the same speaker and different speakers, respectively. From these figures, at the same speaker case, the distribution of generated i-vectors were close to that of original ones. And at the different speaker case, the distributions of original and generated i-vectors were difference.

To visualize the original and generated i-vectors in low dimensional space, t-distributed stochastic neighbor embedding (t-SNE) [29] was performed. t-SNE is a technique that

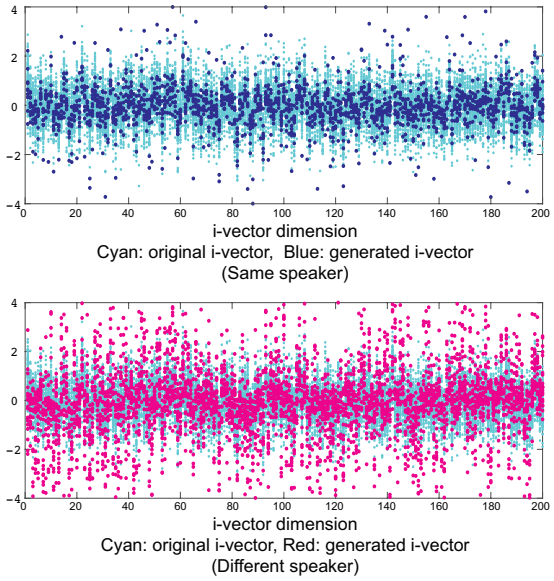

Fig. 2. Comparison of the original and generated i-vectors

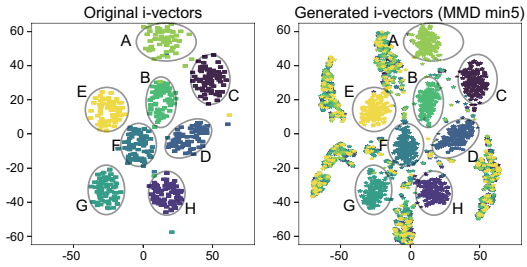

Fig. 3. i-vectors distribution (Original i-vectors and generated i-vectors (MMD min5)).

visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map. Figure 3 depicts the two dimensional map from 100 original and 100 generated i-vectors per speaker. The generated i-vectors were sampled from the MMNs trained with the parameter sets of "min5 which obtained the best accuracy. The alphabets in Fig. 3 denote the speaker ID. The left and right figures were the distributions of original and generated i-vectors, respectively. From the left figure, almost all original i-vectors were distributed in each speakers area. The right figure illustrates that some generated i-vectors were distributed in the originated speaker's distribution. However, the others were placed outside of each speaker's areas, and the outside i-vectors were mixed in the close areas.

In the case of the other parameter sets (i.e., "max 30" and "min15+max15"), the distributions were scattered outside of each speakers area. It denoted that the parameter sets which obtained lower accuracies contained weak speaker dependency. These results suggested that the lower MMD values represented the higher speaker similarity between the original

and generated i-vectors.

V. Conclusion

This paper reported the preliminary investigations about the capability of the generated i-vectors from the MMNs for SV systems. By using MMNs and noise distributions, randomly generated i-vectors were utilized for the i-vector/SVM systems. Although the accuracy of each parameter set was dependent on the parameter sets of the MMNs, the generated i-vectors led to higher accuracies than the baseline system. For the same speaker, the generated i-vectors were shown to be close the original i-vectors, and the distributions of the original and generated i-vectors were similar. Based on these findings, the generated i-vectors were proven to represent the original speaker characteristics.

Our future work includes evaluation of our scheme with a large database and investigation of a method to effectively estimate the parameters of i-vector augmentation. A practical use of the generated i-vectors will be discussed.

VI. Acknowledgements

This work was supported in part by Grant-in-Aid for scientific Research (B), 16757733 and SECOM Science and Technology Foundation.

References

[1] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using gmm supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.

[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. 14, no. 4, pp. 788–798, 2011.

[3] J. Zhong, W. Hu, F.K. Soong, and H. Meng, "Dnn i-vector speaker verification with short, text-constrained test utterances," *in Proc. Interspeech*, pp. 1507–1511, 2017.

[4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *in Proc. ICASSP*, 2018.

[5] A. Hatch, S. Kajarekar, and A. Stolcke, "Withinclass covariance normalization for svm-based speaker recognition," *in Proc. Interspeech*, pp. 1471–1474, 2006.

[6] N. Dehak, R. Dehak, J. Glass, D. Reynolds, and P. Kenny, "Cosine similarity scoring without score normalization techniques," *in Proc. Odyssey*, 2010.

[7] P. Kenny, "Bayesian speaker verification with heavy tailed priors," *in Proc. Odyssey*, 2010.

[8] A. Kanagasundaram, D. Dean, S. Sridharan, J. Gonzalez-Dominguez, J. Gonzales-Rodrigues, and D. Ramos, "Improving short utterance i-vector speaker verification using utterance variance modelling and compensation techniques," *Speech Communication*, vol. 59, pp. 69–82, 2014.

[9] K. Audhkhasi, A. Rosenberg, A. Sethy, B. Ramabhadran, and B. Kingsbury, "End-to-end asr-free keyword search from speech," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1351–1359, 2017.

[10] A.v.d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A.W. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

[11] E. Variani, X. Lei, E. McDermott, I.L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," *in Proc. ICASSP*, pp. 4080–4084, 2014.

[12] J. Guo, U.A. Nookala, and A. Alwan, "Cnn-based joint mapping of short and long utterance i-vectors for speaker verification using short utterances," *in Proc. Interspeech*, pp. 3712–3716, 2017.

[13] "The nist year 2016 speaker recognition evaluation plan," https://www.nist.gov/sites/default/files/documents/2016/10/07/sre16evalplanv1.3.pdf.

[14] "The nist year 2012 speaker recognition evaluation plan," https://www.nist.gov/sites/default/files/documents/itl/iad/mig/NIST_SRE12_evalplan-v17-r1.pdf.

[15] K. Tokuda and H. Zen, "Directly modeling speech waveforms by neural networks for statistical parametric speech synthesis," *in Proc. ICASSP*, pp. 4215–4219, 2015.

[16] Y. Wang, R.J. Skerry-Ryan, D. Stanton, Y. Wu, R.J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q.V. Le, Y. Agiomyrgiannakis, R. Clark, and Rif A. Saurous, "Tacotron: Towards end-to-end speech synthesis," *CoRR*, vol. abs/1609.03499, 2017.

[17] S. Takamichi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "A statistical sample-based approach to GMM-based voice conversion using tied-covariance acoustic models," *IEICE Transactions on Information and Systems*, vol. E99-D, no. 10, pp. 2490–2498, 2016.

[18] S. Takamichi, K. Tomoki, and H. Saruwatari, "Sampling-based speech parameter generation using moment-matching network," *in Proc. Interspeech*, pp. 3961–3965, 2017.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *in Proc. NIPS*, pp. 2672–2680, 2014.

[20] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," *in Proc. ICML*, pp. 1718–1727, 2015.

[21] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," *in Proc. NIPS*, 2016.

[22] Y. Ren, J. Li, Y. Luo, and J. Zhu, "Conditional generative moment-matching networks," *in Proc. NIPS*, pp. 2928–2936, 2016.

[23] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," 2018.

[24] S. Mun, S. Park, D.K. Han, and H. Ko, "Generative adversarial network based acoustic scene training set augmentation and selection using svm hyper-plane," *Detection and Classification of Acoustic Scenes and Events*, 2017.

[25] K. Ito, M. Yamamoto, K. Takeda, T. Takezawa, T. Matsuoka, T. Kobayashi, K. Shikano, and S. Itahashi, "Jnas: Japanese speech corpus for large vocabulary continuous speech recognition research," *the Acoustical Society of Japan (E)*, vol. 20, no. 3, pp. 199–206, 1999.

[26] S. Shiota, F. Villavicencio, J. Yamagishi, N. Ono, I. Echizen, and T. Matsui, "Voice liveness detection algorithms based on pop noise caused by human breath for automatic speaker verification," *in Proc. Interspeech*, pp. 239–243, 2015.

[27] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *EURASIP Journal on Applied Signal Processing*, vol. 12, pp. 2121–2159, 2011.

[28] L.A. Maas, Y.A. Hannun, and Y.A. Ng, "Rectifier nonlinearities improve neural network acoustic models," *in Proc. ICML*, vol. 30, no. 1, 2013.

[29] L.v.d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, pp. 2579–2605, 2008.