The following publication Q. Yao and M. -W. Mak, "SNR-Invariant Multitask Deep Neural Networks for Robust Speaker Verification," in IEEE Signal Processing Letters, vol. 25, no. 11, pp. 1670-1674, Nov. 2018 is available at https://doi.org/10.1109/LSP.2018.2870726.

SNR-Invariant Multi-Task Deep Neural Networks for Robust Speaker Verification

Qi YAO and Man-Wai MAK, *Senior Member, IEEE*

***Abstract*—A major challenge in speaker verification is to achieve low error rates under noisy environments. We observed that background noise in utterances will not only enlarge the speaker-dependent *i*-vector clusters but also shift the clusters, with the amount of shift depending on the signal-to-noise ratio (SNR) of the utterances. To overcome this SNR-dependent clustering phenomenon, we propose two deep neural network (DNN) architectures: hierarchical regression DNN (H-RDNN) and multi-task DNN (MT-DNN). The H-RDNN is formed by stacking two regression DNNs in which the lower DNN is trained to map noisy *i*-vectors to their respective speaker-dependent cluster means of clean *i*-vectors and the upper DNN aims to regularize the outliers that cannot be denoised properly by the lower DNN. The MT-DNN is trained to denoise *i*-vectors (main task) and classify speakers (auxiliary task). The network leverages the auxiliary task to retain speaker information in the denoised *i*-vectors. Experimental results suggest that these two DNN architectures together with the PLDA backend significantly outperform the multi-condition PLDA model and mixtures of PLDA, and that multi-task learning helps to boost verification performance.**

***Index Terms*—Deep learning; speaker verification; *i*-vectors, multi-task learning; noise robustness**

I. Introduction

SPeaker verification (SV) aims to verify the identity of a claimant through analysing his/her voice. Despite the success of the *i*-vector/PLDA framework [1]–[3] in text-independent SV, applying the standard *i*-vector/PLDA framework to real-world noisy environments is still challenging. To tackle the noise robustness problem, a promising approach is to confine the noisy acoustic features to a low-dimensional subspace through feature-domain factor analysis [4]. Another approach is to consider *i*-vectors as features and apply transformation techniques, such as linear discriminant analysis (LDA) [5] and within-class covariance normalization (WCCN) [6], to transform the *i*-vectors to a subspace that is less sensitive to the noise. As for the backend PLDA modeling, multi-condition PLDA training [7], [8] and soft-aligned mixtures of PLDA [9], [10] have also demonstrated noise robustness.

The relationship between acoustic noise and *i*-vectors is rather complex and possibly nonlinear. Because LDA, WCCN, and PLDA are linear models, their capability in dealing with the nonlinear noise effect on the *i*-vectors is limited. Recent studies have shown that deep neural networks (DNNs) are better candidates for addressing this nonlinearity. For example, in [11], speaker vectors are extracted from *i*-vectors by disentangling the latent dependence between speaker and channel components. In [12], [13], the PLDA model parameters are replaced by the weights learned from stacked restricted Boltzmann machines (RBM). In [14], a discriminative denoising autoencoder (DDAE) is used to map the noisy *i*-vectors to their clean counterparts directly by explicitly using clean *i*-vector information and speaker identity information.

This paper analyzes how noisy speech affects the distribution of *i*-vectors. The analysis suggests that channel effects could cause large intra-speaker variability of speaker clusters and that the variability depends on the SNR levels. This SNR-dependent noise effect, referred to as SNR variability, makes the backend PLDA model difficult to separate the speaker variability and the channel variability from noisy *i*-vectors. To address this issue, we propose two DNN-based model to suppress both channel and SNR variabilities directly in the *i*-vector space. The first model, referred to as hierarchical regression DNNs (H-RDNNs), comprises two denoising regression DNNs stacked hierarchically. The second model is a multi-task DNN (MT-DNN) trained to perform both *i*-vector denoising (regression) and speaker classification.

II. Variability Analyses and Proposed Models

I-vectors capture not only the speaker characteristics but also other characteristics in the utterances. In [15], the authors observed that noise-contaminated *i*-vectors with similar SNRs tend to form clusters in the *i*-vector space. This SNR-grouping phenomenon motivates the use of PLDA mixture models in [9], [10] so that each SNR group can be handled by a PLDA model. We argue that instead of tackling the SNR-dependent *i*-vector groups as in [9], it is more effective to compensate for such variability in the *i*-vector space directly.

A. Hierarchical Regression DNN

The structure of a hierarchical regression DNN (H-RDNN) is shown in Fig. 1. Denote $\mathbf{x}_n$ as training *i*-vectors preprocessed by WCCN and length normalization (LN), and $\mathbf{t}_n$ as target *i*-vectors obtained by averaging speaker-dependent *i*-vectors from clean utterances. Given a training set $\mathcal{S}$ composed of N utterances: $\mathcal{S} = \{\mathbf{x}_n, \mathbf{t}_n; n = 1, \ldots, N\}$, in the first stage, the regression network $f^{reg}_{\boldsymbol{\Theta}}(\cdot)$ aims to minimize the MSE and the Frobenius norm of weight paramters:[1]

$$\min_{\boldsymbol{\Theta}} \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}||f^{reg}_{\boldsymbol{\Theta}}(\mathbf{x}_n) - \mathbf{t}_n||_2^2 + \frac{\beta_{reg_1}}{2}||\boldsymbol{\Theta}||_2^2, \qquad (1)$$

where $f^{reg}_{\boldsymbol{\Theta}}(\mathbf{x}_n)$ is the output of the top regression layer of the first (left) DNN in Fig. 1; $\boldsymbol{\Theta}$ comprises all of the weights in the

Q. Yao and M. W. Mak are with Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR (Email: michael.yao@connect.polyu.hk; enmwmak@polyu.edu.hk).

This work was in part supported by RGC of Hong Kong (Grant No. PolyU 1521371/17E) and HKPolyU (Grant No. 4-ZZGT).

[1] If $\mathbf{x}_n$ and $\mathbf{t}_n$ are obtained from a noise-contaminated utterance and its clean counterpart, Eq. 1 leads to the denoising autoencoder (DAE) [16].

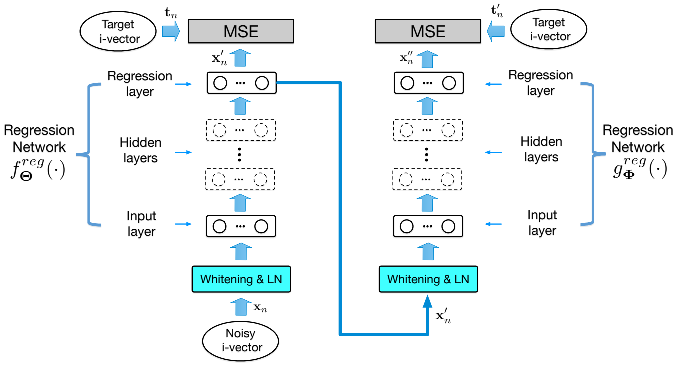

Fig. 1. The proposed H-RDNN. The blue arrows represent the data flow. Noisy i-vector $\mathbf{x}_n$ is fed to the 1-st regression DNN. The 2-nd regression DNN takes the output $\mathbf{x}'_n$ of the 1-st regression DNN and produces the final denoised i-vector $\mathbf{x}''_n$. $f_{\mathbf{\Theta}}^{reg}(\cdot)$ and $g_{\mathbf{\Phi}}^{reg}(\cdot)$ are the mapping functions of the 1-st and 2-nd regression networks, respectively. $\mathbf{t}_n$ and $\mathbf{t}'_n$ are the target i-vectors in Eq. 1 and Eq. 2, respectively. MSE: mean squared error.

first regression network and β_{reg_1} is a coefficient controlling the degree of regularization. After training, the first regression DNN is able to suppress both channel and SNR variations within each speaker cluster simultaneously.

After the first denoising stage, a second regression DNN is trained to regularize the outliers that cannot be denoised properly by the first regression DNN. Suppose all i-vectors have been processed by the first DNN followed by whitening and LN. Given a training set composed of N utterances: $\mathcal{S}' = \{\mathbf{x}'_n, \mathbf{t}'_n; n = 1, \ldots, N\}$, the MSE of the regression network $g_{\mathbf{\Phi}}^{reg}(\cdot)$ and a regularization term are jointly minimized:

$$\min_{\mathbf{\Phi}} \frac{1}{N} \sum_{n=1}^{N} \frac{1}{2} ||g_{\mathbf{\Phi}}^{reg}(\mathbf{x}'_n) - \mathbf{t}'_n||_2^2 + \frac{\beta_{reg_2}}{2} ||\mathbf{\Phi}||_2^2, \qquad (2)$$

where $\mathbf{x}'_n$ is the n-th i-vector denoised by the first DNN, i.e., $\mathbf{x}'_n = f_{\mathbf{\Theta}}^{reg}(\mathbf{x}_n)$; $\mathbf{t}'_n$ is the corresponding i-vector from the original i-vector set (no noise corruption) and then denoised by the first DNN, i.e., $\mathbf{t}'_n = f_{\mathbf{\Theta}}^{reg}(\mathbf{x}_n^{org})$; $\mathbf{x}''_n = g_{\mathbf{\Phi}}^{reg}(\mathbf{x}'_n)$ is the output of the regression layer of the second regression DNN; $\mathbf{\Phi}$ represents the weights in the second regression network and β_{reg_2} is a coefficient controlling the degree of regularization.

B. Multi-Task DNN

To reduce speaker information loss in the regression task, we need to ensure a large between-speaker scatter and a small within-speaker scatter in the DNN-transformed i-vectors. This can be achieved by training a multi-task DNN (MT-DNN) as shown in Fig. 2. Denote $\mathbf{x}_n$, $\mathbf{t}_n$, and $\boldsymbol{\ell}_n$ as the pre-processed i-vector, the target i-vector, and the speaker label vector (in one-hot format) of the n-th utterance (could be clean or noisy), respectively. Suppose we have a training set $\mathcal{S}'$ of N utterances: $\mathcal{S}' = \{\mathbf{x}_n, \mathbf{t}_n, \boldsymbol{\ell}_n; n = 1, \ldots, N\}$. For the regression network $f_{\mathbf{\Theta}_1}^{reg}(\cdot)$, the MSE is minimized in the same way as Eq. 1. The output of the top regression layer is the denoised i-vector $\mathbf{x}'_n$, i.e., $\mathbf{x}'_n = f_{\mathbf{\Theta}_1}^{reg}(\mathbf{x}_n)$. For the classification network $f_{\mathbf{\Theta}_2}^{cls}(\cdot)$, the cross-entropy (CE) cost together with the Frobenius norm of weights in the classification network are jointly minimized:

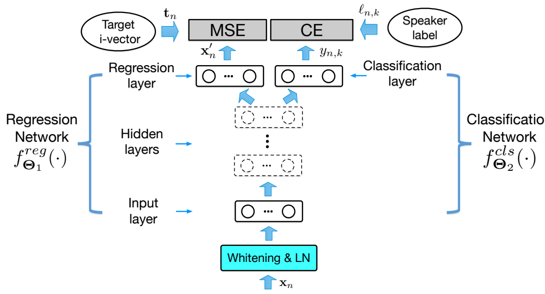

Fig. 2. The proposed MT-DNN. Noisy i-vector $\mathbf{x}_n$ is used as the input. The output of the regression task is $\mathbf{x}'_n$ and the output of the classification task is $y_{n,k}$. The target i-vector for the regression task is $\mathbf{t}_n$ in Eq. 1 and the target label for the classification task is $\ell_{n,k}$ in Eq. 3. The regression and classification tasks are trained in an alternating manner. MSE: mean squared error; CE: cross entropy.

$$\min_{\mathbf{\Theta}_2} -\frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \ell_{n,k} \log y_{n,k} + \frac{\beta_{cls}}{2} ||\mathbf{\Theta}_2||_2^2. \qquad (3)$$

In Eq. 3, $\ell_{n,k}$ is the k-th element of $\boldsymbol{\ell}_n$; if the utterance of $\mathbf{x}_n$ is spoken by the k-th speaker, then $\ell_{n,k} = 1$, otherwise it is equal to 0; $y_{n,k}$ is the posterior probability of the k-th speaker (K speakers in total) and is an element of $\mathbf{y}_n$, where $\mathbf{y}_n$ is the output of the classification network, i.e., $\mathbf{y}_n = f_{\mathbf{\Theta}_2}^{cls}(\mathbf{x}_n)$; $\mathbf{\Theta}_2$ represents the weights of the classification network and β_{cls} is the coefficient controlling the degree of regularization.

The regression and classification tasks are trained in an alternating manner, i.e., for each iteration, the gradients for updating network parameters are computed either from the regression cost function or from the classification cost function.

III. Experimental Setup

A. Speech Data and Acoustic Features

Evaluations were performed on common conditions 4 (CC4, male) and 5 (CC5, male) of NIST 2012 SRE [17]. Speech files from NIST 2005–2010 SREs were used as development data. Speech regions were extracted by performing a two-channel voice activity detection (VAD) [18]. A 25-ms Hamming window with a frame shift of 10 ms was used for extracting windowed speech signals. For each frame, 19 Mel frequency cepstral coefficients (MFCC) with the log energy and their first and second derivatives were computed to form a 60-dimensional acoustic feature vector. Then, cepstral mean normalization [19] followed by feature warping [20] with a 3-second window were applied to the acoustic features.

Three types of noise were considered. They are babble noise from the PRISM dataset [21], heating, ventilation, and air conditioning noise (HVAC) from freesound.org, and factory noise from NOISEX-92 [22]. These noises were added to the evaluation test set in CC4 at SNR of 15 dB, 6 dB and 0 dB using the FaNT tool [23]. Likewise, the original telephone speech files from 2006–2010 SREs were also corrupted by the three different noises at SNR of 15 dB and 6 dB to form 3 SNR groups. Therefore, for each noise type, multi-condition training was applied to train the DNNs and PLDA models using the 3 SNR groups.

TABLE I
PERFORMANCE OF PLDA BASELINE, MIXTURES OF PLDA AND 3 NEURAL NETWORK-BASED MODELS IN CC 4 AND CC 5 OF NIST 2012 SRE. WCCN+LN+LDA-WCCN IS THE *i*-VECTOR PRE-PROCESSING METHOD FOR PLDA-BASED MODELS. WCCN IS THE *i*-VECTOR POST-PROCESSING METHOD FOR DNN-BASED MODELS. SI-MPLDA: SNR-INDEPENDENT MIXTURE OF PLDA [9]; SD-MPLDA: SNR-DEPENDENT MIXTURE OF PLDA [9]; DAE: DENOISING AUTOENCODER; H-RDNN: HIERARCHICAL REGRESSION DNN; MT-DNN: MULTI-TASK DNN. WCCN: WITHIN-CLASS COVARIANCE NORMALIZATION; LDA: LINEAR DISCRIMINANT ANALYSIS; LN: LENGTH NORMALIZATION.

Model	Main Task	Auxiliary Task	Noise Type	CC 4		CC 5	
				Original		Original	
				EER(%)	minDCF	EER(%)	minDCF
Multi-condition PLDA	N/A		Babble	4.02	0.352	3.61	0.343
			HVAC	4.23	0.331	3.24	0.307
			Factory	4.23	0.348	3.44	0.301
SI-mPLDA (4 mixtures)			Babble	3.88	0.333	3.21	0.306
SD-mPLDA (4 mixtures)				3.80	0.353	3.48	0.338
DAE+PLDA	Regression	None	Babble	3.32	0.339	2.93	0.329
H-RDNN+PLDA				3.24	0.348	2.95	0.338
MT-DNN+PLDA		Classification		**3.12**	**0.325**	**2.76**	**0.307**
DAE+PLDA	Regression	None	HVAC	3.18	0.322	2.68	0.301
H-RDNN+PLDA				3.08	0.341	2.66	0.305
MT-DNN+PLDA		Classification		**2.92**	**0.304**	**2.52**	**0.271**
DAE+PLDA	Regression	None	Factory	3.23	0.315	**2.62**	0.308
H-RDNN+PLDA				**3.06**	0.325	2.67	0.320
MT-DNN+PLDA		Classification		3.09	**0.302**	2.72	**0.298**

TABLE II
PERFORMANCE IN CC4 OF NIST 2012 SRE UNDER 3 DIFFERENT TYPES OF NOISE AND 3 SNR CONDITIONS IN THE TEST SEGMENTS.

Model	Main Task	Auxiliary Task	Noise Type	15 dB		6 dB		0 dB	
				EER(%)	minDCF	EER(%)	minDCF	EER(%)	minDCF
Multi-condition PLDA	N/A		Babble	2.54	0.266	2.84	0.325	4.56	0.500
			HVAC	2.55	0.263	3.07	**0.290**	4.58	0.411
			Factory	2.63	0.244	2.74	**0.263**	4.15	0.385
SI-mPLDA (4 mixtures)			Babble	2.42	**0.237**	2.85	**0.314**	4.55	0.478
SD-mPLDA (4 mixtures)				2.68	0.271	2.91	0.335	4.36	0.497
DAE+PLDA	Regression	None	Babble	2.13	0.278	2.55	0.337	3.89	0.437
H-RDNN+PLDA				2.15	0.280	2.56	0.341	3.92	0.435
MT-DNN+PLDA		Classification		**2.05**	0.272	**2.48**	0.316	**3.82**	**0.428**
DAE+PLDA	Regression	None	HVAC	2.02	0.281	2.56	0.311	3.90	0.396
H-RDNN+PLDA				2.04	0.279	**2.53**	0.310	**3.89**	0.394
MT-DNN+PLDA		Classification		**1.94**	0.260	2.57	0.298	4.00	**0.380**
DAE+PLDA	Regression	None	Factory	2.21	0.265	2.41	0.293	3.83	0.359
H-RDNN+PLDA				2.16	0.266	2.39	0.296	**3.81**	**0.357**
MT-DNN+PLDA		Classification		**2.05**	0.249	**2.34**	0.281	3.86	0.366

B. I-Vector/PLDA System

The *i*-vector extractor is based on a gender-dependent UBM with 1024 Gaussian mixtures and a total variability matrix with 500 total factors. Microphone and telephone speech files from 2005–2008 SREs were used to train the UBM and total variability matrix. 500-dimensional *i*-vectors, derived from 3 SNR groups comprising 21,468 utterances from 763 speakers, were pooled together and then processed by WCCN and LN. Thereafter, a projection matrix computed by LDA and WCCN was applied to reduce the dimensionality of *i*-vectors from 500 to 200. A Gaussian PLDA model was then trained using 200-dimensional *i*-vectors with 150 speaker factors.

C. Neural Network Systems

We evaluated three neural network systems: conventional DAE, H-RDNN (Fig. 1) and MT-DNN (Fig. 1). The DAE consists of one input layer (500 nodes), two hidden layers (2000 nodes each) with hyperbolic tangent (tanh) activations and one output layer (500 nodes) with linear activations. The sub-networks of H-RDNN have the same setup as the DAE. For the MT-DNN, the regression network is the same as that of the H-RDNN in the first stage. The classification network has a softmax layer with 763 nodes (one per speaker). Before training, all weights were initialized using Xavier initialization [24] with parameters suitable for tanh activations. Dropout [25] together with L_2-norm weight decay were applied to provide regularization. Adadelta [26] with a batch size of 150 was used as the gradient descent algorithm. All DNN programs were run on the TensorFlow platform [27].

IV. RESULTS AND DISCUSSIONS

Fig. 3 shows the distributions of 20 speaker clusters formed by the raw *i*-vectors (original, 15 dB, and 6 dB) and the *i*-vectors transformed by different DNN models. Evidently,

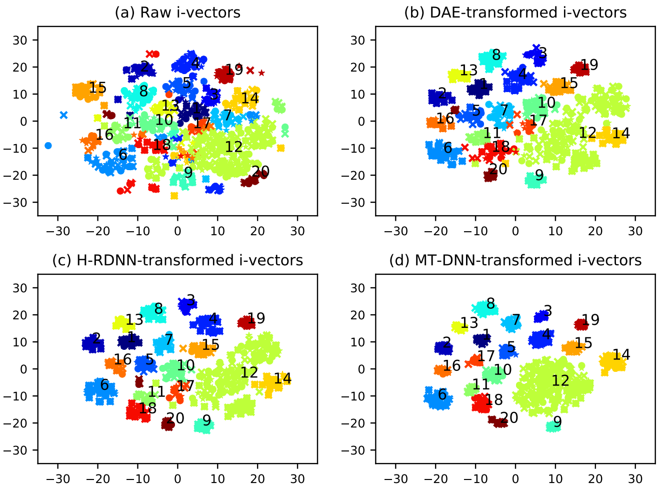

Fig. 3. T-SNE plots [28] of 20 speaker clusters from 3 SNR groups (org+15dB+6dB, telephone speech, babble noise). The raw *i*-vectors in (a) were transformed by DAE (b), H-RDNN (c), and MT-DNN (d). Speakers are marked with different colors and *i*-vectors from the three SNR groups are marked with ○, ×, and ∗, respectively.

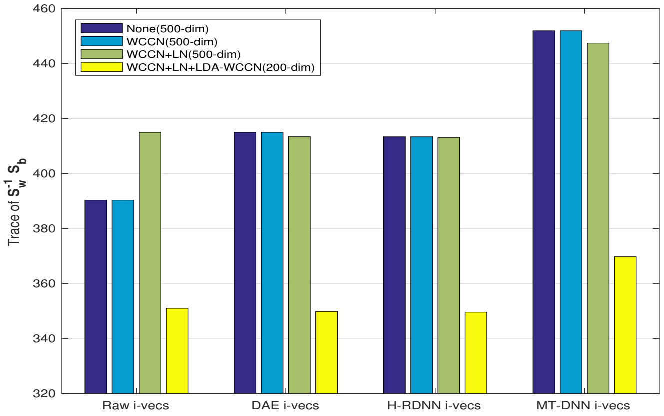

Fig. 4. Dispersion of 20 speaker clusters from 3 SNR groups (org+15dB+6dB, telephone speech, babble noise). The x-axis indicates the types of DNN transformation methods applied to the raw *i*-vectors. The y-axis indicates the values of $\mathrm{Tr}(\mathbf{S}_w^{-1}\mathbf{S}_b)$. The colors in the legend denotes different *i*-vector post-processing methods applied to the DNN-transformed *i*-vectors.

the speaker clusters of the original *i*-vectors (top-left) are less distinguishable than those of the DNN-transformed *i*-vectors. Taking the 6-th speaker cluster (wathet blue) in the top-left subfigure as an example, the left-most • of this speaker deviates significantly from its centroid. As this *i*-vector is derived from an uncontaminated utterance, the deviation is mainly caused by channel effects. The 15-dB and 6-dB *i*-vectors marked with crosses and asterisks exhibit severe variations within each speaker cluster. After denoising, *i*-vectors processed by the MT-DNN have the most compact clusters. This suggests that the MT-DNN-transformed *i*-vectors are less channel- and SNR-dependent and more speaker-dependent, which is a favorable property for PLDA modeling.

Fig. 4 shows the trace of $\mathbf{S}_w^{-1}\mathbf{S}_b$ obtained from the same set of *i*-vectors used for producing Fig. 3, where $\mathbf{S}_b$ and $\mathbf{S}_w$ are the between- and within-speaker scatter matrices. This value measures the dispersion of speaker clusters. The bars indicated by different colors represent different *i*-vector post-processing methods. Since $\mathrm{Tr}\{\mathbf{S}_w^{-1}\mathbf{S}_b\}$ is invariant to the linear transformation of WCCN, the first two bars (None, WCCN) have the same values. Moreover, it shows that the speaker clusters processed by the MT-DNN have the largest degree of separation.

Table I shows the results with respect to EER and SRE12 minDCF [17] achieved by the conventional methods and the 3 DNN-based methods. This table shows the best combination of WCCN, LN and LDA-WCCN for the PLDA baseline, SI-mPLDA model and SD-mPLDA model. We found that WCCN is the most suitable post-processing method for DNN-transformed *i*-vectors. The 4-th column in Table I reveals that DNN-based models significantly outperform the multi-condition PLDA and mixtures of PLDA under 3 different noise types. Among 3 DNN-based models, conventional DAE shows some denoising effects, with 21.0% reduction in EER and 3.1% reduction in minDCF on average when compared with the PLDA baseline. The performance of H-RDNNs is very close to that of the DAEs. A possible reason is that the *i*-vectors processed by H-RDNN in the first stage is already close to clean. Therefore, the denoising effect may not gain a lot from the second stage of H-RDNN. MT-DNN, with the speaker classification task, achieves the best performance with 24.5% reduction in EER and 8.7% reduction in minDCF on average when compared with the PLDA baseline.

Table II shows the performance achieved by different models under 9 different acoustic conditions (3 types of noise and 3 SNR groups). Three phenomena can be observed. First, the EERs and minDCFs (15 dB and 6 dB) in Table II are lower than those in Table I. It is possible because the SNRs of the noise-contaminated test segments in Table II do not vary a lot, which is easier for the DNNs and PLDA models to handle. Second, the DNN-based models show noise robustness under 9 different noise conditions. Third, MT-DNN together with the PLDA backend perform better than other DNN-based models and PLDA-based models in most cases.

Table I shows that the MT-DNN performs significantly better than the DAE; but in Table II, the MT-DNN performs slightly better only. A possible reason is that the wide SNR range in the original test segments of CC4 and CC5 makes the denoising task very difficult. The DAE is not good enough to handle the task. On the other hand, the narrow SNR ranges in Table II make the job of denoising easier, which reduces the performance gap between the two models.

V. Conclusions

This paper presents a DNN approach to compensating for channel and SNR variabilities. We demonstrated the noise effects on the compactness of speaker-dependent *i*-vector clusters. From this starting point, we illustrated why the noise sources have negative effects on PLDA modeling and proposed H-RDNN and MT-DNN to address the problem. Results showed that the MT-DNN with regression and classification tasks trained alternatively can improve the robustness of speaker verification systems.

References

[1] 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.

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

[3] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." in *Proc. of Odyssey: Speaker and Language Recognition Workshop*, 2010.

[4] T. Hasan and J. H. Hansen, "Acoustic factor analysis for robust speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 842–853, 2013.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[6] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. of Ninth International Conference on Spoken Language Processing*, 2006.

[7] T. Hasan, S. O. Sadjadi, G. Liu, N. Shokouhi, H. Bořil, and J. H. Hansen, "CRSS systems for 2012 NIST speaker recognition evaluation," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6783–6787.

[8] D. Garcia-Romero, X. Zhou, and C. Y. Espy-Wilson, "Multicondition training of Gaussian PLDA models in i-vector space for noise and reverberation robust speaker recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4257–4260.

[9] M. W. Mak, X. Pang, and J. T. Chien, "Mixture of PLDA for noise robust i-vector speaker verification," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 1, pp. 130–142, 2016.

[10] N. Li, M. W. Mak, and J. T. Chien, "DNN-driven mixture of PLDA for robust speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1371–1383, 2017.

[11] Y. Z. Işik, H. Erdogan, and R. Sarikaya, "S-vector: A discriminative representation derived from i-vector for speaker verification," in *Proc. of Signal Processing Conference (EUSIPCO), 23rd European*. IEEE, 2015, pp. 2097–2101.

[12] S. Novoselov, T. Pekhovsky, O. Kudashev, V. S. Mendelev, and A. Prudnikov, "Non-linear PLDA for i-vector speaker verification," in *Proc. of Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[13] T. Pekhovsky, S. Novoselov, A. Sholohov, and O. Kudashev, "On autoencoders in the i-vector space for speaker recognition," in *Proc. of Odyssey: The Speaker and Language Recognition Workshop*, 2016, pp. 217–224.

[14] S. Mahto, H. Yamamoto, and T. Koshinaka, "I-vector transformation using a novel discriminative denoising autoencoder for noise-robust speaker recognition," in *Proc. of Interspeech*, 2017, pp. 3722–3726.

[15] N. Li and M.-W. Mak, "SNR-invariant PLDA with multiple speaker subspaces," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5565–5569.

[16] Z. Tan, M.-W. Mak, B. K.-W. Mak, and Y. Zhu, "Denoised senone i-vectors for robust speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 4, pp. 820–830, 2018.

[17] NIST, "The NIST year 2012 speaker recognition evaluation plan," http://www.nist.gov/itl/iad/mig/sre12.cfm, 2012.

[18] M. W. Mak and H. B. Yu, "A study of voice activity detection techniques for NIST speaker recognition evaluations," *Computer Speech & Language*, vol. 28, no. 1, pp. 295–313, 2014.

[19] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *the Journal of the Acoustical Society of America*, vol. 55, no. 6, pp. 1304–1312, 1974.

[20] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, Crete, Greece, Jun. 2001, pp. 213–218.

[21] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot *et al.*, "Promoting robustness for speaker modeling in the community: The PRISM evaluation set."

[22] "http://www.speech.cs.cmu.edu/comp.speech/section1/data/noisex.html."

[23] H. G. Hirsch, "FaNT: Filtering and Noise Adding Tool," Niederrhein University of Applied Sciences, 2005. [Online]. Available: http://dnt.kr.hsnr.de/download.html

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[27] M. Abadi, et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[28] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.