Senone I-Vectors for Robust Speaker Verification

Zhili TAN[1]*, Yingke ZHU*[2]*, Man-Wai MAK*[1]*, Brian Kan-Wing MAK*[2]

[1]Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University
[2]Dept. of Computer Science and Engineering, The Hong Kong University of Science and Technology
eddy.zhili@connect.polyu.hk, yzhuav@cse.ust.hk, man.wai.mak@polyu.edu.hk, mak@cse.ust.hk

Abstract

Recent research has shown that using senone posteriors for i-vector extraction can achieve outstanding performance. In this paper, we extend this idea to robust speaker verification by constructing a deep neural network (DNN) comprising a deep belief network (DBN) stacked on top of a denoising autoencoder (DAE). The proposed method addresses noise robustness in two perspectives: (1) denoising the MFCC vectors through the DAE and (2) extracting noise robust bottleneck (BN) features and senone posteriors from the DBN for total-variability matrix training and i-vector extraction. The DAE comprises several layers of restricted Boltzmann machines (RBM), which are trained to minimize the mean squared error between the denoised and clean MFCCs. After training the DAE, three layers of RBMs are put on top of it to form the DNN. The whole network is fine-tuned by backpropagation to minimize the cross-entropy between the senone labels and network outputs. This architecture allows us to extract BN features and estimates senone posteriors given noisy MFCCs as input, resulting in robust BN-based senone i-vectors. Results on NIST 2012 SRE show that these senone i-vectors outperform the conventional i-vectors and the BN-based i-vectors in which the posteriors are obtained from a GMM.

Index Terms: speaker verification, i-vectors, senone posteriors, deep learning, denoising autoencoders.

1. Introduction

In recent years, the i-vector approach [1] that confines the speaker and channel variability into a low dimensional subspace has dominated the speaker verification community. Due to the great success of deep learning [2], a lot of effort has been made on combining i-vectors and deep neural networks (DNNs). There are several ways to achieve this combination. For example, in [3, 4], researchers explored the potential of using bottleneck (BN) features extracted from deep belief networks (DBNs) to replace the standard mel-frequency cepstral coefficients (MFCCs) [5]. As another example, in [6], DBNs pre-trained by contrastive divergence [7], were used to generate the posteriors of the mixtures of a universal background model (UBM).

Inspired by the great success of DNNs [8], convolutional neural networks (CNNs) [9] and recurrent neural networks (RNNs) [10, 11] in large vocabulary continuous speech recognition, an i-vector extraction method that uses the posteriors of senones rather than the posteriors of GMM-mixtures was proposed in [12, 13]. Aligning acoustic frames to senones allows direct comparisons of speakers based on the same set of sub-phonetic units produced by the speakers [14]. In [15], the method was extended to replace the MFCCs in [12] by bottleneck features extracted from a DNN. A similar idea has also been applied to i-vector based DNN adaptation for robust speech recognition [16].

DNNs are also applicable to restoring spectral vectors for speech enhancement [17, 18] and restoration of unreliable i-vectors in short-utterance speaker recognition [19]. The idea is to use denoising deep autoencoders (DAE) [20, 21, 22] to denoise or restore speech either in the spectral domain or in the i-vector space.

This paper explores the use of DNNs for extracting robust bottleneck features from noisy speech and for computing senone posteriors for BN-based i-vector extraction. We have recently proposed a denoising deep classifier (DDC) by stacking restricted Boltzmann machines (RBMs) on the top of a DAE [23]. The whole network was trained to produce the posteriors of speaker IDs given noisy speech as input. Bottleneck features were then extracted from the RBM layer just below the output (softmax) layer. Results in [23] suggest that the DAE is very effective in suppressing the effect of noise in the input speech, making the BN feature noise robust. A drawback of the method, however, is that the BN vectors of the same utterance are very close to each other in the BN space, causing numerical difficulty when training the BN-based UBM and the total variability matrix. In this paper, we proposed to solve this problem by training the DDC to produce senone posteriors instead of speaker posteriors. The advantage of this remedy is that as long as a training utterance is phonetically balance, the BN vectors extracted from the RBM layer will scatter over different regions of the BN space. Together with the denoising capability of DAE, the proposed denoising deep classifier can produce noise robust BN features and robust senone posteriors for i-vector extraction. Experimental results on NIST 2012 SRE demonstrate that the proposed BN-based i-vectors are less susceptible to babble noise, even at 0dB.

2. System Overview

2.1. Conventional i-vector extractor

I-vectors, based on factor analysis, is a dimension reduction method that compresses the speaker and channel information of GMM-supervectors into a subspace. Given the i-th utterance, we denote $\mathcal{O}_i = \{\boldsymbol{o}_{i1}, \ldots, \boldsymbol{o}_{iT_i}\}$ as a set of F-dimensional acoustic feature vectors, which are assumed to follow a mixture distribution:

$$p(\boldsymbol{o}_{it}) = \sum_c \pi_c p(\boldsymbol{o}_{it}|c), \quad c = 1, \ldots, C$$

where $p(\boldsymbol{o}_{it}|c)$ is the conditional likelihood of $\boldsymbol{o}_{it}$ and π_c's are the mixture weights.

The work described in this paper was partially supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Nos. PolyU 152068/15E and HKUST 616513).

In the conventional i-vector framework, the GMM-supervector representing the i-th utterance is assumed to follow a factor analysis model of the form:

$$\boldsymbol{\mu}_i = \boldsymbol{\mu}^{(b)} + \boldsymbol{T}\boldsymbol{w}_i + \boldsymbol{\epsilon}_i,$$

where $\boldsymbol{\mu}^{(b)}$ is the supervector formed by stacking the mean vectors of the UBM, $\boldsymbol{T}$ is a $CF \times D$ low-rank total variability matrix (T-matrix) modeling the speaker and channel subspace, $\boldsymbol{w}_i$ is a latent factor of dimension D, and $\boldsymbol{\epsilon}_i$ is the residual noise following a zero-mean Gaussian distribution. In practice, $\boldsymbol{\epsilon}_i$ is assumed to follow a Gaussian distribution: $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}^{(b)})$, where $\boldsymbol{\Sigma}^{(b)}$ is the covariance matrix of the UBM.

Given N training utterances, the T-matrix can be estimated by the following EM algorithm [24, 25]:

• E-step:

$$\langle \boldsymbol{w}_i|\mathcal{O}_i\rangle = \boldsymbol{L}_i^{-1}\sum\nolimits_c \boldsymbol{T}_c^{\intercal}\boldsymbol{\Sigma}_c^{-1}\tilde{\boldsymbol{f}}_{ic},$$
$$\langle \boldsymbol{w}_i\boldsymbol{w}_i^{\intercal}|\mathcal{O}_i\rangle = \boldsymbol{L}_i^{-1} + \langle \boldsymbol{w}_i|\mathcal{O}_i\rangle\langle \boldsymbol{w}_i|\mathcal{O}_i\rangle^{\intercal},$$
$$\boldsymbol{L}_i^{-1} = \boldsymbol{I} + \boldsymbol{T}^{\intercal}(\boldsymbol{\Sigma}_c)^{-1}\boldsymbol{N}_i\boldsymbol{T}, \quad i = 1, ..., N;$$

• M-step:

$$\boldsymbol{T}_c = \left[\sum\nolimits_i \tilde{\boldsymbol{f}}_{ic}\langle \boldsymbol{w}_i|\mathcal{O}_i\rangle^{\intercal}\right]\left[\sum\nolimits_i N_{ic}\langle \boldsymbol{w}_i\boldsymbol{w}_i^{\intercal}|\mathcal{O}_i\rangle\right]^{-1}.$$

Here, i indexes the set of training utterances, $\boldsymbol{T}_c$ is the c-th partition of $\boldsymbol{T}$, $\boldsymbol{\Sigma}_c$ is the c-th block of $\boldsymbol{\Sigma}^{(b)}$, and N_{ic} and $\tilde{\boldsymbol{f}}_{ic}$ are the 0th- and 1st-order Baum-Welch statistics respectively:

$$N_{ic} = \sum\nolimits_t \gamma_c(\boldsymbol{o}_{it}),$$
$$\tilde{\boldsymbol{f}}_{ic} = \sum\nolimits_t \gamma_c(\boldsymbol{o}_{it})(\boldsymbol{o}_{it} - \boldsymbol{\mu}_c).$$

Given the t-th frame of the i-th utterance, $\boldsymbol{o}_{it}$ is the MFCC vector of the t-th frame and $\gamma_c(\boldsymbol{o}_{it})$ is the posterior of the c-th mixture component in the UBM:

$$\gamma_c(\boldsymbol{o}_{it}) = \frac{\lambda_c^{(b)}\mathcal{N}(\boldsymbol{o}_{it}|\boldsymbol{\mu}_c^{(b)}, \boldsymbol{\Sigma}_c^{(b)})}{\sum_{j=1}^{C}\lambda_j^{(b)}\mathcal{N}(\boldsymbol{o}_{it}|\boldsymbol{\mu}_j^{(b)}, \boldsymbol{\Sigma}_j^{(b)})},$$

where $\boldsymbol{\theta} = \{\lambda_j^{(b)}, \boldsymbol{\mu}_j^{(b)}, \boldsymbol{\Sigma}_j^{(b)}\}_{j=1}^{C}$ are UBM parameters.

2.2. Generalized i-vector extractor

In most systems, $\{\boldsymbol{\mu}_c\}$ and $\{\boldsymbol{\Sigma}_c\}$ are obtained from the UBM. However, they can also be obtained using the sufficient statistics as follows:

$$\boldsymbol{\mu}_c = \frac{\sum_i\sum_t \gamma_c(\boldsymbol{o}_{it})\boldsymbol{o}_{it}}{\sum_i N_{ic}}$$
$$\boldsymbol{\Sigma}_c = \frac{\sum_i \boldsymbol{S}_{ic}}{\sum_i N_{ic}},$$

where $\boldsymbol{S}_{ic} = \sum_t \gamma_c(\boldsymbol{o}_{it})(\boldsymbol{o}_{it} - \boldsymbol{\mu}_c)(\boldsymbol{o}_{it} - \boldsymbol{\mu}_c)^{\intercal}$. Therefore, without the UBM, we can still estimate the T-matrix and i-vectors as long as the Baum-Welch statistics are available. In fact, only the observed vectors $\boldsymbol{o}_{it}$ and mixture posteriors $\gamma_c(\boldsymbol{o}_{it})$ are necessary for i-vector extraction.

For example, we may replace the MFCC by other types of acoustic features and estimate the mixture posteriors $\gamma_c(\mathbf{o}_{it})$ from other model rather than the UBM Specially, the acoustic feature vectors and mixture posteriors can respectively be written in more general forms:

$$\boldsymbol{o}_{it} = f(\boldsymbol{s}_{it}), \qquad \gamma_c(\boldsymbol{s}_{it}) = P(c|\boldsymbol{s}_{it}), \tag{1}$$

where $\boldsymbol{s}_{it}$ represents the speech signal in a contextual window comprising multiple frames centered at frame t and $f(\boldsymbol{s}_{it})$ is a function that extracts acoustic vectors from $\boldsymbol{s}_{it}$.

2.3. DNN with Denoising Autoencoder

In [13], $P(c|\boldsymbol{s}_{it})$ are given by a DNN which is trained to produce the posteriors of senones given multiple frames of MFCCs as input. Here, we train a DNN formed by stacking a DBN on top of a denoising deep autoencoder [23] to improve the noise robustness of $P(c|\boldsymbol{s}_{it})$. Furthermore, to enrich the contextual information in $\mathbf{o}_{it}$, they are extracted from the bottleneck layer just below the softmax layer of the DNN. More precisely, $f(\boldsymbol{s}_{it})$ in Eq. 1 represents the combined effect of the denoising operation in the DAE and the feature extraction operation in the DBN using contextual MFCCs ($\boldsymbol{s}_{it}$) as input.

Fig. 1 illustrates the procedures to train our denoising deep classifier.[1] To equip our autoencoder with denoising ability, we used both clean and noisy speech as input and their corresponding clean counterparts as target outputs, with the squared loss as the error function. In the RBM pre-training, only the first half of the RBMs are needed to be trained, and the second half of the RBMs are their mirrored ones due to the symmetry of the autoencoder. Since we used MFCCs as inputs to the DNN, the first RBM is a Gaussian-Bernoulli RBM and the last layer of the autoencoder is linear.

Once the denoising deep autoencoder has been trained, we built the denoising deep classifier using the senone labels as the targets. By adding three layers of RBMs on the top of the DAE followed by backpropagation fine-tuning, the nextwork can extract the phonetic information even if the input is noisy. Although the whole denoising deep classifier is fine-tuned without parameter fixing in the bottom layers, the part of previous DAE may still keep the denoising ability.

The first RBM on top of the DAE is Gaussian-Bernoulli and the last RBM is Bernoulli-Gaussian where the Gaussian hidden layer is of small size. The reason is that we aim to extract the low dimensional BN features with Gaussian distributions from the BN layer — the one below the softmax output layer. The BN features are used to replace MFCCs in the i-vector framework.

Except for the BN layer and the last layer of the DAE, all hidden layers comprise sigmoid units. The output comprises softmax nodes. More specifically, assume that there are K distinct senones, the DNN outputs are given by

$$y_k(\mathbf{x}) = \frac{e^{h_k}}{\sum_{k'=1}^{K} e^{h'_k}}, \quad k = 1, \ldots, K,$$

where $\mathbf{x}$ is the input to the DNN, h_k is the activation of the k-th output node, and $y_k(\mathbf{x})$ is the softmax output at node k. The network is trained by minimizing the cross-entropy:

$$E(\mathcal{X}, \mathcal{Z}, \mathcal{C}) = -\sum\nolimits_{i=1}^{K}\sum\nolimits_{j=1}^{M_i}\sum\nolimits_{k=1}^{K} z_{i,j,k}\log(y_k(\mathbf{x}_{i,j}))$$

where $\mathbf{z}_{i,j}$'s are one-of-K vectors indicating to which senone the input vector $\mathbf{x}_{i,j}$ belongs and M_i is the number of vectors from senone i.

2.4. Senone i-vectors

Sections 2.2 and 2.3 give us a new i-vector framework: senone i-vectors. If we can integrate the DDC into i-vector extractor, the resulting senone i-vectors should be noise robust. They should also outperform the conventional i-vectors due to the phonetic information from the BN layers.

Fig. 2 illustrates the procedure of senone i-vector extraction. As we have discussed in Section 2.2, only the 0th-, 1st-

[1] In the sequel, we denote a DNN equipped with a DAE in the bottom layers (Fig. 2) as denoising deep classifier (DDC).

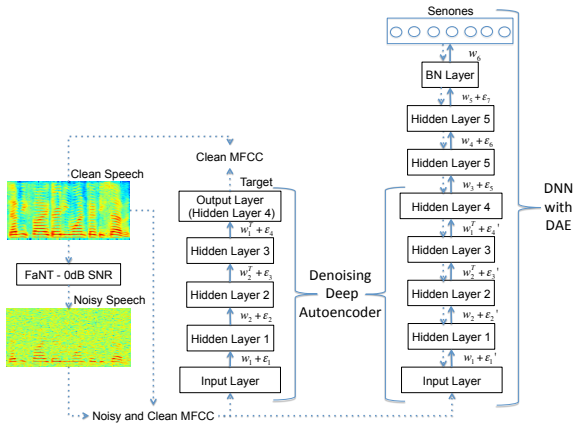

Figure 1: *Procedure of training the denoising deep classifier.*

and 2nd-order Baum-Welch statistics are needed for T-matrix training, and the 0th- and 1st-order statistics are necessary for i-vector extraction. Our idea is to replace MFCC acoustic features and the UBM posteriors by the BN features and senone posteriors from the DDC (DNN with DAE).

Since the BN features are highly correlated, we used principal component analysis (PCA) whitening to perform decorrelation. The decorrelation process allows us to use diagonal covariance matrices for the BN-based UBM.

Following the notation in Section 2.2, the procedure for extracting senone i-vectors is as follows:

- BN feature vectors: $\boldsymbol{o}_{it} = \text{BN}(\boldsymbol{s}_{it})$
- Senone posteriors: $\gamma_c(\boldsymbol{s}_{it}) = P_{DNN}(c|\boldsymbol{s}_{it})$, which is the output of the c-th node in the softmax output layer.
- Baum-Welch statistics:

$$N_{ic} = \sum_t P_{DNN}(c|\boldsymbol{s}_{it}),$$

$$\tilde{\boldsymbol{f}}_{ic} = \sum_t P_{DNN}(c|\boldsymbol{s}_{it})(\text{BN}(\boldsymbol{s}_{it}) - \boldsymbol{\mu}_c),$$

$$\boldsymbol{S}_{ic} = \sum_t P_{DNN}(c|\boldsymbol{s}_{it})(\text{BN}(\boldsymbol{s}_{it}) - \boldsymbol{\mu}_c)(\text{BN}(\boldsymbol{s}_{it}) - \boldsymbol{\mu}_c)^{\mathsf{T}},$$

where:

$$\boldsymbol{\mu}_c = \frac{\sum_i \sum_t \gamma_c(\boldsymbol{s}_{it})\text{BN}(\boldsymbol{s}_{it})}{\sum_i N_{ic}}, \qquad \boldsymbol{\Sigma}_c = \frac{\sum_i \boldsymbol{S}_{ic}}{\sum_i N_{ic}}.$$

$$\boldsymbol{S}_{ic} = \sum_t \gamma_c(\boldsymbol{s}_{it})(\text{BN}(\boldsymbol{s}_{it}) - \boldsymbol{\mu}_c)(\text{BN}(\boldsymbol{s}_{it}) - \boldsymbol{\mu}_c)^{\mathsf{T}}.$$

Therefore we can combine the BN features and DNN posteriors to generate the senone i-vectors, and this combination integrates the phonetic information in the DNN into the i-vectors.

3. Experiments

3.1. Speech data and feature extraction

Speaker verification experiments were performed on the NIST 2012 SRE under Common Condition 4 (CC4). This common condition involves 723 target speakers with 7116 target utterances and 3900 test utterances. Each utterance is about 10 to 300 seconds long, sampled at 8kHz, and spoken in English. The baseline is a conventional i-vector/PLDA system, where the acoustic features are MFCCs and the posteriors were obtained from a GMM-based UBM with 1024 mixtures. 19 MFCCs and log-energy were computed for each frame. Together with their

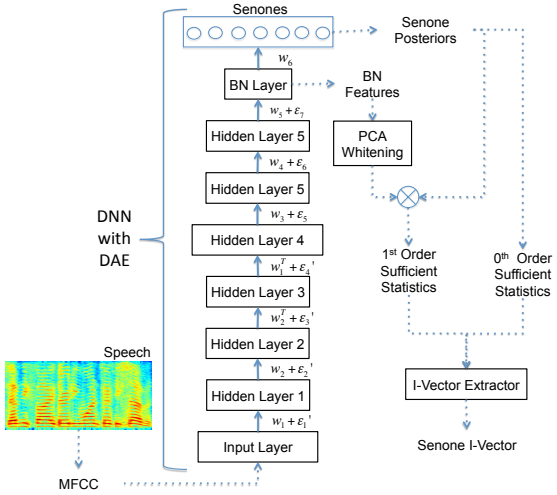

Figure 2: *Procedure of senone i-vector extraction.*

1st and 2nd derivatives, a 60-dimensional MFCC vector was obtained for every 20-ms frame. Feature warping were then applied.

All i-vector extractors have 500 total factors. The PLDA further reduces the speaker subspace to 150 dimensions.

3.2. Senone label extraction

We used a DNN-HMM acoustic model trained on SwitchBoard-1 release 2 to obtain the senone label for each frame. SWitchBoard-1 release 2 contains approximately 290 hours of US English telephone conversations spoken by 500 speakers. The 4870 conversation sides were spliced into 259,890 utterances for acoustic modeling. The original DNN has 6 hidden layers with 2048 nodes per layer, and an output softmax layer with 8704 nodes, corresponding to 8704 clustered states (senones). We further clustered the 8704 senones to 2000 senones, resulting in a DNN with 2000 outputs nodes. The features are 13-dimensional cepstral mean-variance normalized (CMVN) MFCCs, and they were extracted from speech data every 10ms over a window of 25ms. For each frame, its neighbouring 4 frames were included and transformed by linear discriminative analysis (LDA) to 40 dimensions, followed by maximum likelihood linear transformation. Speaker adaptation was also applied with feature-space maximum likelihood linear regression (fMLLR).

For each frame, the fMLLR-transformed vectors of the 5 preceding and 5 succeeding frames were fed to the DNN, which outputs the posterior probabilities of different senones, and the one with the highest posterior is the senone label for the frame.

3.3. Training of denoising deep classifier

The senone labels produced by the DNN-HMM were used as targets for training the denoising deep classifier (DDC) shown in Fig. 1. The FaNT tool [26] was used to add babble noise to the 7116 target-speaker utterances of CC4 at 15dB, 6dB and 0dB respectively. The input of the DDC comprises eleven 60-dimensional MFCC vectors extracted from 11 contextual frames, which amounts to $20 \times 11 = 220$ input nodes. Element-wise z-norm was applied to the 220 inputs so that Gaussian-Bernoulli RBM pre-training can be applied. As shown in Fig. 1, the DAE has a structure 220-256-256-256-220, where the first

Table 1: *Performance of various i-vector/PLDA systems in NIST 2012 SRE (CC4, male speaker, core task) with test utterances contaminated with different levels of babble noise. DNN_2 is the DNN with DAE (Fig. 2). DNN_1 has the same structure as DNN_2, but its bottom layers are not a DAE.*

		Original		15dB		6dB		0dB	
Acoustic Features	Posteriors from	EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF
MFCC	GMM	3.675	0.311	3.495	0.310	3.842	0.406	6.515	0.720
BN Features from DNN_1	DNN_1	3.346	**0.268**	2.650	0.223	2.990	0.286	**3.419**	**0.446**
BN Features from DNN_2	DNN_2	**3.243**	**0.268**	**2.403**	**0.211**	**2.882**	**0.277**	3.741	0.453

Table 2: *Performance of BN-based i-vector/PLDA systems based on various posteriors in NIST 2012 SRE (CC4, male speaker, core task) with test utterances contaminated with different levels of babble noise. DNN_2 is the DNN with DAE (Fig. 2).*

		Posteriors from	
		GMM	DNN_2
15dB	EER	3.269	**2.448**
	minDCF	0.263	**0.236**
6dB	EER	3.493	**2.774**
	minDCF	0.368	**0.311**
0dB	EER	4.608	**4.503**
	minDCF	0.551	**0.544**

and last values are the number of inputs and outputs, respectively.

After the DAE had been fine-tuned by backpropagation, three RBMs were put on the top of the DAE, where the bottom one is a Gaussian-Bernoulli RBM and the top one is a Bernoulli-Gaussian RBM. Backpropagation fine-tuning was then applied to the combined DAE and RBMs using the 2000 senone labels as target outputs with the one-of-K coding scheme. As shown in Fig. 1, the final DDC has a structure 220-256-256-256-220-256-256-60-2000, where the last softmax layer has 2000 nodes and the bottleneck layer has 60 nodes.

As the procedure in Section 2.4 and Fig. 2 shown, we can obtain the senone i-vectors by combining BN features and senone posteriors. With the same PLDA back-end as the baseline, we can compare the performance of senone i-vectors and conventional i-vectors.

3.4. Results on NIST 2012 SRE with babble noise

Table 1 shows the EER and minDCF of various i-vector/PLDA systems that use different acoustic features and different ways of computing the senone posteriors. To investigate the noise robustness of senone i-vectors, we used the FaNT tool to add babble noise to the test utterances at SNR of 15dB, 6dB and 0dB, respectively. Because some of the test utterances in CC4 have SNR lower than 15dB, no noise will be added to these files when the target SNR is 15dB. As a result, the performance at 15dB in Table 1 is based on test utterances with SNR at or below 15dB; similarly, the performance at 6dB is based on test utterances with SNR at or below 6dB.

The utterances used for training the PLDA models for the four test conditions in Table 1 come from the same set of conversations (target-speakers' utterances in SRE12). However, depending on the SNR of the test conditions in Table 1, two PLDA models were trained by using utterances with different SNR ranges. Specifically, babble noise was added to the original telephone conversations of target speakers at SNR of 0dB, 6dB and 15dB, which results in 3 groups of training utterances, namely 15dB group, 6dB group, and 0dB group. Then, for the test conditions labeled with "Original", "15dB", and "6dB" in Table 1, the PLDA models were trained by using the original telephone and microphone (interview speech and telephone speech recorded over microphone channels) utterances plus the noise contaminated telephone utterances with SNR of 6dB and 15dB. For the 0dB test-condition in Table 1, in addition to the above utterances, 0dB telephone utterances were also used for training the PLDA models.

Because the babble noise poses a great challenge to voice activity detection (VAD), we used the VAD decisions obtained from the original test utterances for all of the test conditions. Although this procedure causes under-estimations of the performance in Table 1, it avoids the complications arising from the wrong VAD decisions. It also allows us to purely compare the capability of different acoustic features and frame-posterior estimation methods, as the comparisons will become meaningless when too many non-speech frames were leaked into the feature and i-vector extraction processes.

Table 1 shows that BN features with frame posteriors obtained from DNNs achieve better performance under all test conditions. Specifically, the one with posteriors from DDC achieves the best performance in most of the cases, proving the denoising capacity of the DAE. The performance is also significantly better than the baseline where conventional i-vector extraction method was used. The good performance is attributed to (1) the denoising capability of the DAE, (2) the use of contextual information in the DDC input (11 frames), and (3) the phonetic-aware frame posteriors provided by the DDC.

Given the BN-features, we may compute the i-vectors based on the posteriors given by a BN-based UBM or the senone posteriors given by the DDC (Fig. 2). To compare these two approaches, we used the original and 0dB telephone utterances to train a DDC and a BN-based UBM and compared the performance of the resulting BN-based i-vectors. Table 2 clearly demonstrates that using the posteriors from the DDC produces significantly better performance, even for the SNR conditions (15dB and 6dB) never seen by the DDC.

4. Conclusions and Future Work

This paper has shown that robust BN features and frame posteriors can be obtained from a denoising deep classifier (DDC) formed by the combination of a denoising deep autoencoders (DAE) and a deep belief network (DBN). The DAE is able to suppress noise in MFCC vectors and the DBN enforces the frame alignments to respect the phonetic context of input speech. A possible extension is to train a large DDC using noisy speech with a wide range of SNR or to train multiple DDCs so that each one focuses on a narrow range of SNR.

5. References

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

[2] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[3] Y. Tian, M. Cai, L. He, and J. Liu, "Investigation of bottleneck features and multilingual deep neural networks for speaker verification," in *Proc. Interspeech*, 2015.

[4] W. Geng, J. Li, S. Zhang, X. Cai, and B. Xu, "Multilingual tandem bottleneck feature for language identification," in *Proc. Interspeech*, 2015.

[5] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition," *Speech Communication*, vol. 54, no. 4, pp. 543–565, 2012.

[6] W. Campbell, "Using deep belief networks for vector-based speaker recognition," in *Proc. Interspeech*, 2014.

[7] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[8] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[9] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for LVCSR," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[10] J. Li, A. Mohamed, G. Zweig, and Y. Gong, "Exploring multidimensional LSTMs for large vocabulary ASR," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4940–4944.

[11] L. Lu, X. Zhang, and S. Renais, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5060–5064.

[12] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.

[13] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, "Study of senone-based deep neural network approaches for spoken language recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 24, no. 1, pp. 105–116, 2016.

[14] D. Garcia-Romero and A. McCree, "Insights into deep neural networks for speaker recognition," in *Proc. Interspeech*, 2015, pp. 1141–1145.

[15] F. Richardson, F. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[16] C. Yu, A. Ogawa, M. Delcroix, T. Yoshioka, T. Nakatani, and J. H. Hansen, "Robust i-vector extraction for neural network adaptation in noisy environment," in *Proc. Interspeech*, 2015, pp. 2854–2857.

[17] X.-L. Zhang and D. Wang, "Multi-resolution stacking for speech separation based on boosted dnn," in *Proc. Interspeech*, 2015, pp. 1745–1749.

[18] K. Kinoshita, M. Delcroix, A. Ogawa, and T. Nakatani, "Text-informed speech enhancement with deep neural networks," in *Proc. Interspeech*, 2015.

[19] H. Yamamoto and T. Koshinaka, "Denoising autoencoder-based speaker feature restoration for utterances of short duration," in *Proc. Interspeech*, 2015, pp. 1052–1056.

[20] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[21] J. H. Liu, W. Q. Zheng, and Y. X. Zou, "A robust acoustic feature extraction approach based on stacked denoising autoencoder," in *Multimedia Big Data (BigMM), IEEE International Conference on*, 2015, pp. 124–127.

[22] T. Pekhovsky, S. Novoselov, A. Sholohov, and O. Kudashev, "On autoencoders in the i-vector space for speaker recognition," in *Odyssey*, 2016.

[23] Z. Tan and M. W. Mak, "Bottleneck features from SNR-adaptive denoising deep classifier for speaker identification-adaptive denoising deep classifier for speaker identification," in *Asia-Pacific Signal and Information Processing Association, Annual Summit and Conference (APSIPA ASC)*, 2015, pp. 1035–1040.

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

[25] M. W. Mak, "Lecture notes on factor analysis and i-vectors," Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University, Tech. Rep., 2016.

[26] H. Hirsch, "Fant-filtering and noise adding tool," 2005.