The following publication Z. Tan, M. Mak, B. K. Mak and Y. Zhu, "Denoised Senone I-Vectors for Robust Speaker Verification," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 4, pp. 820-830, April 2018 is available at https://doi.org/10.1109/TASLP.2018.2796843.

Denoised Senone I-Vectors for Robust Speaker Verification

Zhili TAN, Man-Wai MAK, *Senior Member, IEEE,* Brian Kan-Wing MAK, *Senior Member, IEEE,* Yingke ZHU

***Abstract*—Recently, it has been shown that senone i-vectors, whose posteriors are produced by senone deep neural networks (DNNs), outperform the conventional Gaussian mixture model (GMM) i-vectors in both speaker and language recognition tasks. The success of senone i-vectors relies on the capability of the DNN to incorporate phonetic information into the i-vector extraction process. In this paper, we argue that to apply senone i-vectors in noisy environments, it is important to robustify the phonetically discriminative acoustic features and senone posteriors estimated by the DNN. To this end, we propose a deep architecture formed by stacking a deep belief network (DBN) on top of a denoising autoencoder (DAE). After backpropagation fine-tuning, the network, referred to as denoising autoencoder–deep neural network (DAE–DNN), facilitates the extraction of robust phonetically-discriminitive bottleneck (BN) features and senone posteriors for i-vector extraction. We refer to the resulting i-vectors as denoised BN-based senone i-vectors. Results on NIST 2012 SRE show that senone i-vectors outperform the conventional GMM i-vectors. More interestingly, the BN features are not only phonetically discriminative, results suggest that they also contain sufficient speaker information to produce BN-based senone i-vectors that outperform the conventional senone i-vectors. This work also shows that DAE training is more beneficial to BN feature extraction than senone posterior estimation.**

***Index Terms*—speaker verification, i-vectors, phonetically discriminative features, senone posteriors, deep learning, denoising autoencoders, noise robustness.**

I. Introduction

Speaker verification, an important pathway to biometric authentication, has been dominated by the combination of the i-vector approach [1] and probabilistic linear discriminant analysis (PLDA) [2] since 2011. The former is considered as a feature extraction method that converts variable-length utterances into fixed-length feature vectors. The latter is a probabilistic backend where unwanted variabilities are marginalized out when computing the likelihood ratio scores. To raise the efficiency of this framework, efforts have been made to improve the combination of i-vector and PLDA. For example, Cumani and Laface [3] proposed nonlinearly transforming the i-vectors to make them more suitable for PLDA modeling.

A major limitation of the i-vector/PLDA framework is that the speaker characteristics in i-vectors can be easily distorted by background noise and reverberation effects. One approach to improving the robustness of i-vector systems is to directly reduce the distortion at the spectral level. For example, Xing and Hansen [4] reduced the frequency-shift distortion due to modulation/demodulation carrier mismatch for speaker recognition. Human tend to change their vocal effort under noisy environments (a phenomenon known as the Lombard effect), causing acoustic mismatch between normal speech and shouted speech. Saedi *et al.* [5] addressed this problem by compressing/expanding power spectra in autocorrelation-based linear prediction features. Both [4] and [5] demonstrate that reducing spectral distortion can make the i-vectors more resilient to background noises.

The use of speech enhancement techniques to improve speaker recognition performance has drawn the attention of the speaker recognition community. Unlike conventional speech enhancement, the goal is to robustify the feature vectors instead of reconstructing the clean waveforms. For example, Hasan and Hansen [6] performed feature-domain factor analysis to enhance and transform acoustic vectors. The transformed feature vectors were then used for computing the sufficient statistics in the i-vector extraction procedure. In [7], [8], [9], i-vectors extracted from short utterances or from noisy utterances are restored by stacked denoising autoencoders [10].

Attempts have also been made to improve noise robustness in PLDA models. For example, Hasan *et al.* [11] and Garcia-Romero *et al.* [12] trained a PLDA model by pooling speeches from multiple conditions, and Li and Mak [13], [14] modeled the noise-level variability in utterances by introducing an SNR factor and an SNR subspace into the PLDA model. In [15], [16], Mak *et al.* advocated that utterances of different SNR levels will not only cause i-vectors to fall on different regions of the i-vector spaces but also change the orientation of the speaker subspace. A mixture PLDA model with mixture alignments determined by the SNR level of utterances was then derived to model SNR-dependent i-vectors.

Because of the great success of deep neural networks (DNNs) [17], convolutional neural networks (CNNs) [18] and recurrent neural networks (RNNs) [19], [20], [21] in automatic speech recognition (ASR), the application of deep learning [22] to speaker verification has been under the spotlight recently. One promising approach is to replace the PLDA backend by DNNs. For instance, Ghahabi and Hernando [23] trained one DNN for each target speaker to discriminate his/her i-vectors from those of the other speakers. Each DNN receives i-vectors as input and produces the posterior probabilities of the target and non-target classes as output. Given a test i-vector, the log-posterior ratio can then be obtained from the network outputs. In [24], the whole i-vector extraction

Z. L. Tan and M. W. Mak are with Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR (Email: eddy.zhili@connect.polyu.hk; enmwmak@polyu.edu.hk). Yingke ZHU and Brian K. W. Mak are with Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong SAR (Email: yzhuav@cse.ust.hk; mak@cse.ust.hk). This work was supported in part by The RGC of Hong Kong SAR (Grant Nos. PolyU 152518/16E and PolyU 152068/15E).

cum PLDA scoring pipeline is replaced by RNNs. Specifically, long short-term memory RNNs were collaboratively trained for speech and speaker recognition tasks, and the contextual information obtained from the speech recognition RNN was found to be assistive to the speaker recognition RNN.

Another promising approach is to integrate DNNs into the i-vector framework. Campell [25] used DBNs pre-trained by contrastive divergence [26] to generate the posteriors of the mixtures of a universal background model (UBM). The posteriors are then used for computing the sufficient statistics of vector-based speaker recognition systems. Lei *et al.* [27], [28] replaced the posteriors of UBM's mixture components in the i-vector extractor by the posteriors of senones. In this approach, acoustic frames are aligned to senones by a DNN so that speakers can be compared based on the same set of sub-phonetic units [29].

It is believed that better and possibly more robust features can be extracted from DNNs. For example, bottleneck features were extracted from DNNs in [30], [31]. The bottleneck features can replace the standard mel-frequency cepstral coefficients [32]. A similar idea has also been applied to i-vector based DNN adaptation for robust speaker recognition [33]. Richardson *et al.* [34] demonstrated that GMM i-vectors based on the phonetically discriminative BN features outperforms the ones based on MFCC significantly on the 2013 Domain Adaptation Challenge (DAC13). Sarker *et al.* [35] showed that the phonetically discriminative BN features are complementary to the short-term cepstral features, and therefore improve the performance significantly on NIST 2008 and 2010 SRE by both score domain and feature domain fusion. These works show that the phonetically discriminative BN features still retain speaker-specific information, possibly taking the benefits of the contextual input window of DNNs.

This paper explores and extends our early work [36] on using DNNs for extracting phonetically discriminative and noise robust bottleneck features from noisy speech and for computing senone posteriors for BN-based i-vector extraction. We have recently proposed a denoising autoencoder–deep neural network by stacking restricted Boltzmann machines (RBMs) on top of a denoising autoencoder [37]. 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 [37] suggest that the DAE is very effective in suppressing the effect of noise in the input speech, making the BN features noise robust. Similar to the DNNs in d-vectors [38] and speaker embedding [39], the DNN in [37] produces speaker posteriors. Because the DNNs of these methods are trained to produce speaker posteriors, their frame-based activations at the bottleneck layer tend to be very similar across the whole utterance. As will be explained in Section II-A, this property will cause numerical difficulty when training the BN-based UBM and the total variability (TV) matrix when the utterances are long. The d-vectors and speaker embedding avoid this problem by averaging the activations across the frames of the entire utterance, which essentially bypasses the UBM training and TV matrix estimation. However, the averaging process throws away lots of speaker information in the frame-based BN vectors, which explains why the performance of d-vectors and speaker embedding is poorer than i-vectors for long utterances [38], [39].

To exploit the denoising capability of denoising autoencoders (DAE) without throwing away speaker information, we propose training the denoising DNN in [37] to produce senone posteriors instead of speaker posteriors. The advantage of this strategy is that as long as a training utterance is phonetically balanced, its BN vectors will be scattered over different regions of the BN feature space, which solves the numerical problem. With the denoising capability of DAE, the network can produce noise robust BN features and robust senone posteriors for i-vector extraction. We refer to the resulting network as DAE–DNN.

Experimental results on NIST 2012 SRE demonstrate that the proposed BN-based i-vectors are less susceptible to babble noise, even at 0dB. We found that no matter under the GMM i-vector framework or the senone i-vector framework, the phonetically discriminative BN features outperform MFCC in speaker verification tasks. This suggests that the phonetically discriminative BN features still retain speaker-specific information. Furthermore, we demonstrate that the denoising capability works for our denoised BN-based senone i-vectors rather than the denoised MFCC-based senone i-vectors. Specifically, by comparing the combinations of phonetically discriminative BN features and senone posteriors with and without DAE training, we validate that the DAE training is more useful for extracting phonetically discriminative BN features than estimating senone posteriors, especially under common condition 5 of NIST 2012 SRE.

II. System Overview

A. Conventional I-vector Extractor

I-vector extraction is a factor analysis method that compresses all sort of variabilities in speech (including speaker variability) into a low-dimensional subspace of the GMM-supervector space [40]. One important property of i-vectors is that they are low-dimensional representations of utterances regardless of their duration.

We denote $\mathcal{O}_i = \{\boldsymbol{o}_{i1}, \ldots, \boldsymbol{o}_{iT_i}\}$ as a set of F-dimensional acoustic vectors of the i-th utterance, such as MFCC, which are assumed to follow a mixture distribution:

$$p(\boldsymbol{o}_{it}) = \sum_{c=1}^{C} \lambda_c p(\boldsymbol{o}_{it}|c),$$

where $p(\boldsymbol{o}_{it}|c)$ is the conditional likelihood of $\boldsymbol{o}_{it}$ and λ_c's are the mixture weights. The GMM-supervector representing the i-th utterance is assumed to be generated by a factor analysis model:

$$\boldsymbol{\mu}_i = \boldsymbol{\mu}^{(b)} + \boldsymbol{T}\boldsymbol{w}_i, \tag{1}$$

where $\boldsymbol{\mu}^{(b)}$ is the supervector formed by stacking the mean vectors of a universal background model (UBM), $\boldsymbol{T}$ is a $CF \times D$ low-rank total variability matrix (T-matrix) modeling the speaker and channel subspaces, and $\boldsymbol{w}_i$ is a D-dimensional latent factor whose prior follows a standard Gaussian distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. While $\boldsymbol{\mu}_i$ and $\boldsymbol{w}_i$ are utterance-dependent, $\boldsymbol{\mu}^{(b)}$ and $\boldsymbol{T}$ are shared across all speakers and utterances.

Eq. 1 is a generative model in that given the $\boldsymbol{w}_i$ of a speaker, his/her supervector $\boldsymbol{\mu}_i$ can be generated. Of course, the model is not perfect and there will be discrepancy (error) between the truth value of $\boldsymbol{\mu}_i$ and the generated one. In factor analysis, we typically assume that the discrepancy follows a Gaussian distribution with zero mean and covariance $\boldsymbol{\Sigma}$. As the dimension of $\boldsymbol{\mu}_i$ is very high, $\boldsymbol{\Sigma}$ is assumed to be diagonal. In most practical implementation of i-vector extraction, $\boldsymbol{\Sigma}$ is approximated by the covariance matrices in the UBM, i.e.,

$$\boldsymbol{\Sigma} \approx \text{diag}\{\boldsymbol{\Sigma}^{(b)}\} = \text{diag}\{\boldsymbol{\Sigma}_1^{(b)}, \ldots, \boldsymbol{\Sigma}_C^{(b)}\},$$

where $\boldsymbol{\Sigma}_c^{(b)}$ is the c-th covariance matrix of the UBM, which is typically diagonal.

Given N training utterances, the T-matrix can be estimated by the following EM algorithm [41], [42]:

• E-step:

$$\langle \boldsymbol{w}_i | \mathcal{O}_i \rangle = \boldsymbol{L}_i^{-1} \sum_{c=1}^{C} \boldsymbol{T}_c^{\mathsf{T}} (\boldsymbol{\Sigma}_c^{(b)})^{-1} \tilde{\boldsymbol{f}}_{ic}, \tag{2a}$$

$$\langle \boldsymbol{w}_i \boldsymbol{w}_i^{\mathsf{T}} | \mathcal{O}_i \rangle = \boldsymbol{L}_i^{-1} + \langle \boldsymbol{w}_i | \mathcal{O}_i \rangle \langle \boldsymbol{w}_i | \mathcal{O}_i \rangle^{\mathsf{T}}, \tag{2b}$$

$$\boldsymbol{L}_i = \boldsymbol{I} + \sum_{c=1}^{C} \boldsymbol{N}_{ic} \boldsymbol{T}_c^{\mathsf{T}} (\boldsymbol{\Sigma}_c^{(b)})^{-1} \boldsymbol{T}_c, \tag{2c}$$

where $i = 1, \ldots, N$.

• M-step:

$$\boldsymbol{T}_c = \left[\sum\nolimits_i \tilde{\boldsymbol{f}}_{ic} \langle \boldsymbol{w}_i | \mathcal{O}_i \rangle^{\mathsf{T}} \right] \left[\sum\nolimits_i N_{ic} \langle \boldsymbol{w}_i \boldsymbol{w}_i^{\mathsf{T}} | \mathcal{O}_i \rangle \right]^{-1}. \tag{3}$$

In Eq. 2 and Eq. 3, $\langle \cdot | \cdot \rangle$ denotes conditional expectation; i indexes the set of training utterances; N is the number of training utterances; $\boldsymbol{T}_c$ is the c-th partition of $\boldsymbol{T}$; $\boldsymbol{\Sigma}_c^{(b)}$ is the c-th covariance matrix of the UBM; N_{ic} and $\tilde{\boldsymbol{f}}_{ic}$ are the 0th- and 1st-order Baum-Welch statistics respectively:

$$\begin{aligned} N_{ic} &= \sum\nolimits_t \gamma_c(\boldsymbol{o}_{it}), \\ \tilde{\boldsymbol{f}}_{ic} &= \sum\nolimits_t \gamma_c(\boldsymbol{o}_{it})(\boldsymbol{o}_{it} - \boldsymbol{\mu}_c^{(b)}). \end{aligned} \tag{4}$$

Given the t-th frame of the i-th utterance, $\boldsymbol{o}_{it}$ is the MFCC vector of the t-th frame and $\gamma_c(\boldsymbol{o}_{it})$ in Eq. 4 is the posterior of the c-th mixture component in the UBM:

$$\gamma_c(\boldsymbol{o}_{it}) = \frac{\lambda_c^{(b)} \mathcal{N}(\boldsymbol{o}_{it} | \boldsymbol{\mu}_c^{(b)}, \boldsymbol{\Sigma}_c^{(b)})}{\sum_{j=1}^{C} \lambda_j^{(b)} \mathcal{N}(\boldsymbol{o}_{it} | \boldsymbol{\mu}_j^{(b)}, \boldsymbol{\Sigma}_j^{(b)})}, \tag{5}$$

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

Once the T-matrix has been estimated, the i-vector $\langle \boldsymbol{w}_i | \mathcal{O}_i \rangle$ representing the i-th utterance can be computed according to Eq. 2a.

Note that the acoustic vectors $\boldsymbol{o}_{it}$'s are not limited to MFCCs. Instead, they can be BN vectors extracted from a DNN. However, caution should be taken when BN vectors are used. If the DNN is trained to produce speaker posteriors, the BN vectors from the same utterance will be very similar because they come from the same speaker. In other words, for the entire utterance, the frame-based activations at the bottleneck layer are very similar so that the DNN can give a large posterior probabilities in the output node corresponding to the speaker and small probabilities in the rest. The similarity in the BN vectors causes them to align to the same (potentially small) group of Gaussians in the BN-based UBM. This property leads to sparsity in the zeroth- and first-order statistics in Eq. 4, which in turns causes numerical difficulty when computing the matrix inverses in Eq. 2 and Eq. 3. Another issue is that the BN vectors tend to form isolated islands in the BN space (otherwise they cannot differentiate speakers). The small within-speaker variances essentially reduce the effective number of vectors for training the BN-based UBM, which again causes numerical problems. Both of these drawbacks motivate us to use senone posteriors instead of speaker posteriors, as detailed in Section II-D.

B. Generalized I-vector Extractor

In most systems, $\{\boldsymbol{\mu}_c^{(b)}\}$ and $\{\boldsymbol{\Sigma}_c^{(b)}\}$ in Eqs. 2–5 are obtained from the UBM. However, they can also be computed from the sufficient statistics as follows:

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

$$\boldsymbol{\Sigma}_c = \frac{\sum_i \sum_t \gamma_c(\boldsymbol{o}_{it}) (\boldsymbol{o}_{it} - \boldsymbol{\mu}_c)(\boldsymbol{o}_{it} - \boldsymbol{\mu}_c)^{\mathsf{T}}}{\sum_i N_{ic}}.$$

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 the mixture posteriors $\gamma_c(\boldsymbol{o}_{it})$ are necessary for i-vector extraction.[1] This means that we may replace the MFCC by other types of acoustic features and estimate the mixture posteriors $\gamma_c(\mathbf{o}_{it})$ from other model, such as a DNN, rather than the UBM. Specifically, the acoustic feature vectors and mixture posteriors can respectively be written in more general forms:

$$\boldsymbol{o}_{it} = f(\boldsymbol{s}_{it}) \text{ and } \gamma_c(\boldsymbol{s}_{it}) = P(c | \boldsymbol{s}_{it}), \tag{6}$$

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 maps acoustic vectors in $\boldsymbol{s}_{it}$ to $\boldsymbol{o}_{it}$.

C. DNN with Denoising Autoencoder

In [28], $P(c|\boldsymbol{s}_{it})$'s in Eq. 6 are given by a DNN that is trained to produce the posteriors of senones given multiple frames of MFCCs as input. In this work, we trained a DNN formed by stacking a deep belief network (DBN) on top of a denoising autoencoder [37] to improve the noise robustness of $P(c|\boldsymbol{s}_{it})$. The network architecture of the stacked DNN is shown in the right part of Fig. 1. Because of the denoising capability of the DAE and the classification capability of the DNN, we refer to the stacked DNN as denoising autoencoder–deep neural network (DAE–DNN).

Fig. 1 illustrates the procedure to train the DAE–DNN. To equip the autoencoder with denoising capability, we used both clean and noisy speech as input and their corresponding clean counterpart as the target output. The denoising autoencoder

[1] In some literatures, $\gamma_c(\boldsymbol{o}_{it})$'s are referred to as frame posteriors. But they are in fact the posterior probabilities of mixture components.

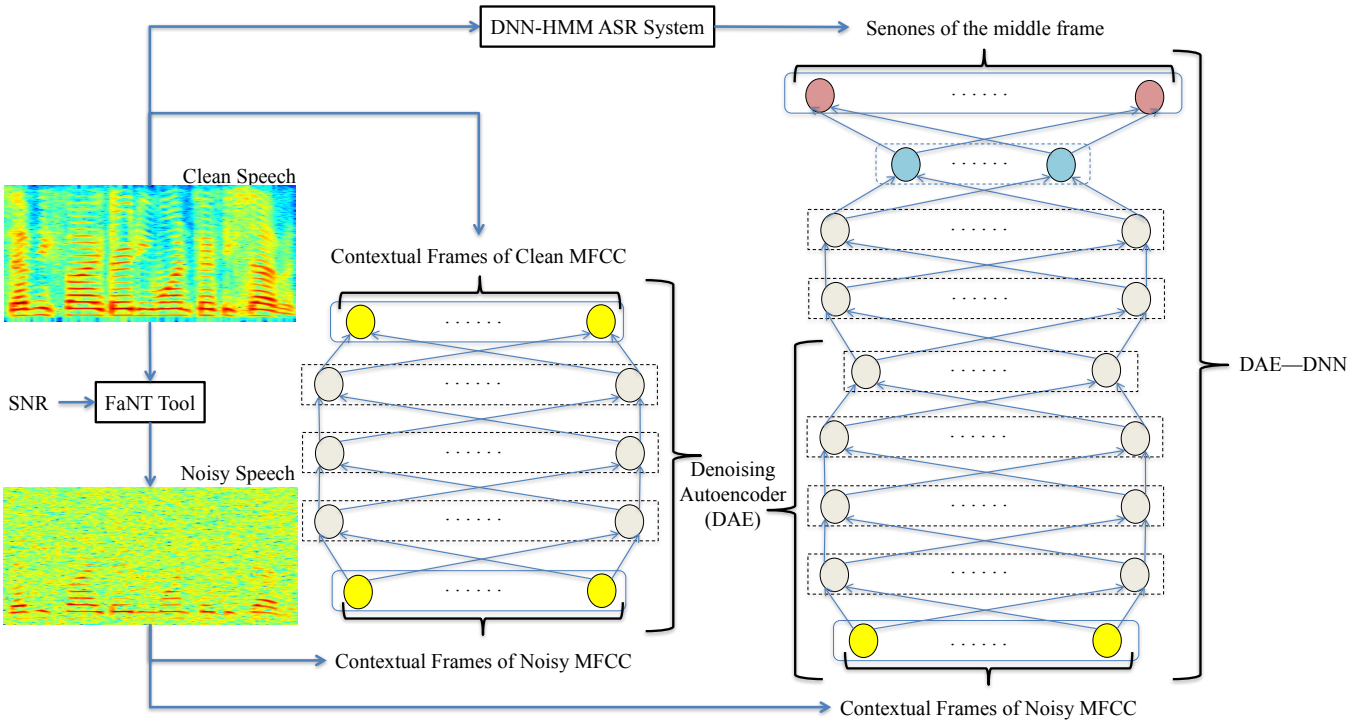


Fig. 1. *Procedure of training the Denoising Autoencoder–Deep Neural Network (DAE–DNN).*

comprises multiple layers of restricted Boltzmann machines, which are trained layer-by-layer using the contrastive divergence algorithm [26] [43]. Only the bottom half of the RBMs need to be trained, and the upper half are the mirrored copies of the lower half 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. The denoising autoencoder is then fine-tuned by the backpropagation algorithm to minimize the squared errors between the outputs and the clean MFCCs. In practice, we obtained the clean–noisy sample pairs by adding babble noise to clean speech using the FaNT tool [44], which will be explained in Section III-A.

Once the denoising autoencoder has been trained, we built the DAE–DNN using the senone labels as the targets. By adding three layers of RBMs on top of the DAE, the network can extract the phonetic information even if the input is noisy.

To enrich the contextual information in $\mathcal{O}_i$, the vectors $\mathbf{o}_{it}$'s are extracted from the bottleneck layer just below the softmax layer of the DNN (the blue nodes in Fig. 1). More precisely, $f(\boldsymbol{s}_{it})$ in Eq. 6 represents the combined effect of the denoising operation in the DAE and the feature extraction operation in the DNN using contextual MFCCs ($\boldsymbol{s}_{it}$) as input. 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. This creates a bottleneck layer (BN) from which the low dimensional BN features can be extracted. The BN features replace the MFCCs during i-vector extraction.

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

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

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 of node k. The network is trained by minimizing the cross-entropy:

$$E(\mathcal{X}, \mathcal{Z}, \mathcal{C}) = -\sum_{r=1}^{K} \sum_{j=1}^{M_r} \sum_{k=1}^{K} z_{r,j,k} \log(y_k(\mathbf{x}_{r,j}))$$

where $\mathbf{z}_{r,j}$'s are one-of-K vectors indicating to which senone the input vector $\mathbf{x}_{r,j}$ belongs and M_r is the number of vectors from senone r. To be more precise, $\mathbf{x}_{r,j}$ comprises contextual frames of MFCCs, which has the same meaning as $\boldsymbol{s}_{it}$ in Eq. 6.

To train the DNN, we need to collect all contextual frames of MFCCs belonging to the same senone (indexed by r). To avoid confusion, we use another symbol $\mathbf{x}$ and another set of subscripts (r and j) to highlight the grouping procedure.

D. Senone I-vectors

The procedures in Sections II-B and II-C produce a new variant of i-vectors: senone i-vectors. If the DAE–DNN can be integrated into the 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 II-B, only the 0th-, 1st- 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. The key idea in this work is to replace

MFCCs by BN features and mixture posteriors from the UBM by senone posteriors from the DAE–DNN.

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 II-B, the procedure for training the T-matrix is as follows:

Step 1: Extract BN feature vectors: $\boldsymbol{o}_{it}^{(d)} = \text{BN}(\boldsymbol{s}_{it})$

Step 2: Compute senone posteriors: $\gamma_c^{(d)}(\boldsymbol{s}_{it}) = P_{\text{DAE–DNN}}(c|\boldsymbol{s}_{it})$, which is the output of the c-th node in the softmax output layer.

Step 3: Compute Baum-Welch statistics:

$$\begin{aligned} N_{ic}^{(d)} &= \sum\nolimits_t P_{\text{DAE–DNN}}(c|\boldsymbol{s}_{it}) \\ \tilde{\boldsymbol{f}}_{ic}^{(d)} &= \sum\nolimits_t \left[P_{\text{DAE–DNN}}(c|\boldsymbol{s}_{it})(\text{BN}(\boldsymbol{s}_{it}) - \boldsymbol{\mu}_c^{(d)}) \right], \\ \boldsymbol{S}_{ic}^{(d)} &= \sum\nolimits_t \Big[P_{\text{DAE–DNN}}(c|\boldsymbol{s}_{it})(\text{BN}(\boldsymbol{s}_{it}) - \boldsymbol{\mu}_c^{(d)}) \times \\ &\qquad (\text{BN}(\boldsymbol{s}_{it}) - \boldsymbol{\mu}_c^{(d)})^{\intercal} \Big], \end{aligned} \tag{7}$$

where

$$\boldsymbol{\mu}_c^{(d)} = \frac{\sum_i \sum_t P_{\text{DAE–DNN}}(c|\boldsymbol{s}_{it})\text{BN}(\boldsymbol{s}_{it})}{\sum_i N_{ic}^{(d)}}.$$

Step 4: Compute the covariance matrices

$$\boldsymbol{\Sigma}_c^{(d)} = \frac{\sum_i \boldsymbol{S}_{ic}^{(d)}}{\sum_i N_{ic}^{(d)}}. \tag{8}$$

Step 5: Replace $\tilde{\boldsymbol{f}}_{ic}$, $\boldsymbol{N}_{ic}$ and $\boldsymbol{\Sigma}_c^{(b)}$ of Eq. 2 by $\tilde{\boldsymbol{f}}_{ic}^{(d)}$, $\boldsymbol{N}_{ic}^{(d)}$ and $\boldsymbol{\Sigma}_c^{(d)}$ in Eq. 7 and Eq. 8:

$$\langle \boldsymbol{w}_i | \mathcal{O}_i^{(d)} \rangle = \boldsymbol{L}_i^{-1} \sum_{c=1}^{C} \boldsymbol{T}_c^{\intercal} (\boldsymbol{\Sigma}_c^{(d)})^{-1} \tilde{\boldsymbol{f}}_{ic}^{(d)}, \tag{9a}$$

$$\langle \boldsymbol{w}_i \boldsymbol{w}_i^{\intercal} | \mathcal{O}_i^{(d)} \rangle = \boldsymbol{L}_i^{-1} + \langle \boldsymbol{w}_i | \mathcal{O}_i^{(d)} \rangle \langle \boldsymbol{w}_i | \mathcal{O}_i^{(d)} \rangle^{\intercal}, \tag{9b}$$

$$\boldsymbol{L}_i = \boldsymbol{I} + \sum_{c=1}^{C} \boldsymbol{N}_{ic}^{(d)} \boldsymbol{T}_c^{\intercal} (\boldsymbol{\Sigma}_c^{(d)})^{-1} \boldsymbol{T}_c, \tag{9c}$$

where $i = 1, \ldots, N$. This constitutes the E-step of the EM algorithm.

Step 6: Replace $\tilde{\boldsymbol{f}}_{ic}$, $\boldsymbol{N}_{ic}$ and $\langle \boldsymbol{w}_i | \mathcal{O}_i \rangle$ of Eq. 3 by $\tilde{\boldsymbol{f}}_{ic}^{(d)}$, $\boldsymbol{N}_{ic}^{(d)}$ and $\langle \boldsymbol{w}_i | \mathcal{O}_i^{(d)} \rangle$ in Eq. 7 and Eq. 9 to compute the T-matrix:

$$\begin{aligned} \boldsymbol{T}_c = &\left[\sum\nolimits_i \tilde{\boldsymbol{f}}_{ic}^{(d)} \langle \boldsymbol{w}_i | \mathcal{O}_i^{(d)} \rangle^{\intercal} \right] \times \\ &\left[\sum\nolimits_i N_{ic}^{(d)} \langle \boldsymbol{w}_i \boldsymbol{w}_i^{\intercal} | \mathcal{O}_i^{(d)} \rangle \right]^{-1}. \end{aligned} \tag{10}$$

This constitutes the M-step of the EM algorithm. Go back to Step 5 with the updated T-matrix until convergency.

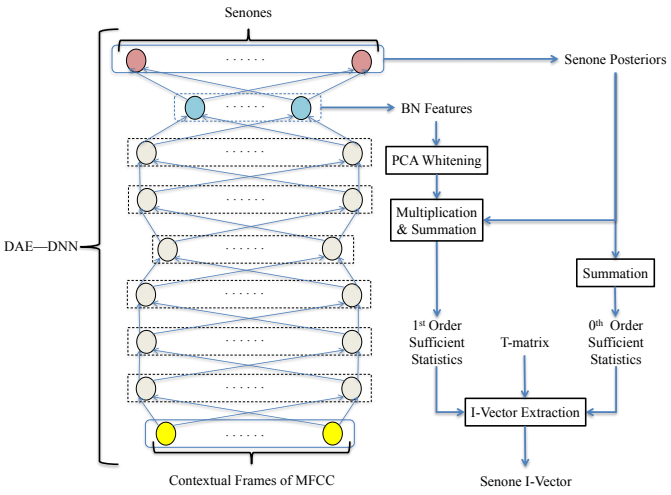


Fig. 2. *Procedure of senone i-vector extraction.*

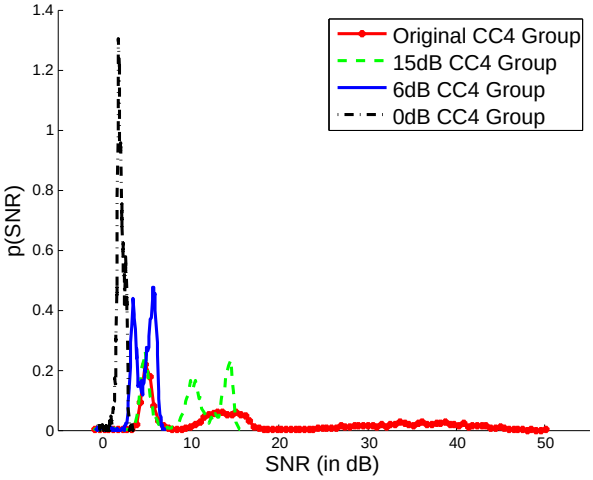


Fig. 3. The SNR distributions of the original and noise contaminated test utterances in NIST 2012 SRE (CC4, male). For the noise contaminated utterances, babble noise was added to the original utterances at an SNR of 0dB, 6dB, and 15dB, respectively.

Once the T-matrix has been estimated, the i-vector $\langle \boldsymbol{w}_i | \mathcal{O}_i^{(d)} \rangle$ representing the i-th utterance can be computed according to Eq. 9a:

$$\langle \boldsymbol{w}_i | \mathcal{O}_i^{(d)} \rangle = \boldsymbol{L}_i^{-1} \sum_{c=1}^{C} \boldsymbol{T}_c^{\intercal} (\boldsymbol{\Sigma}_c^{(d)})^{-1} \tilde{\boldsymbol{f}}_{ic}^{(d)}.$$

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

III. Experiments

A. 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 7,116 target utterances from NIST 2006–2010 SREs and 3,900 test utterances from NIST 2012 SRE, including 125,400 trials in

core test. Each utterance is about 10 to 300 seconds long, sampled at 8kHz, recorded by telephones, and spoken in English. The baseline is a conventional i-vector/PLDA system, where the acoustic features are MFCCs and the mixture posteriors were obtained from GMM-based UBMs with 1024 and 2048 mixtures. The test utterances in CC4 has a wide range of SNR, from 0dB to 50dB as shown in Fig. 3; therefore, CC4 is appropriate for verifying the noise robustness and denoising capability of the proposed algorithm.

To investigate the capability of various i-vector frameworks under noisy environments, we used the FaNT tool [44] to add babble noise to the target-speaker utterances and test utterances at the SNR of 15dB, 6dB, and 0dB, respectively. Therefore, we have four groups of training utterances and four groups of test utterances, with the first group being the original utterances and the last three groups having SNRs close to 15dB, 6dB, and 0dB, respectively. Hereafter, we refer to these 4 groups as SNR groups. The SNR distributions of the 4 groups of test utterances in CC4 are shown in Fig. 3. Note that although the target SNRs that we applied to FaNT are 0dB, 6dB, and 15dB, Fig. 3 shows that the peaks of the SNR distributions do not align to these targets. The misalignment is due to the discrepancy in the VAD decisions for adding noise and for measuring SNRs. Specifically, FaNT has its own VAD for estimating the amount of noise to be added to the clean signals, whereas the *measured* SNRs in Fig. 3 were based on the voltmeter function in FaNT and the decisions of our own noise-robust VAD [45].

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 may give over-optimistic performance, it avoids the complications arising from 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 are included in the i-vector extraction processes.

Nineteen MFCCs and log-energy were computed for each 25-ms frame. Together with their 1st and 2nd derivatives, a 60-dimensional acoustic vector was obtained every 10ms.

B. I-vector Extraction

All i-vector extractors have 500 total factors. The PLDA further reduces the speaker subspace to 150 dimensions. The GMM–UBMs and the total-variability matrixes were trained by using the utterances from the original 7,116 target telephone utterances mentioned earlier and the microphone utterances (interview speech) of the same set of target speakers in NIST 2006–2010 SREs. The PLDA model was trained by using the i-vectors derived from all of the original and noise contaminated telephone utterances and from the interview speech segments of NIST 2006–2010 SREs.

C. Senone Label Extraction

We used a DNN–HMM acoustic model trained on SwitchBoard-1 release 2 to obtain the senone label for each frame. This release contains approximately 290 hours of US English telephone conversations spoken by 500 speakers. The 4,870 conversation sides were spliced into 259,890 utterances for acoustic modeling. The original DNN has 6 hidden layers with 2,048 nodes per layer, and a softmax output layer with 8,704 nodes, corresponding to 8,704 clustered states (senones). We further clustered the 8,704 senones into 2,000 senones, resulting in a DNN with 2,000 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 based on feature-space maximum likelihood linear regression (fMLLR) was also applied.

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.

D. Training of the DAE–DNN

The input of the DAE–DNN comprises eleven 20-dimensional MFCC vectors extracted from 11 contextual frames, which amount 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 and the last values are the numbers of inputs and outputs, respectively. Only the first two layers of the DAE needed to be pre-trained by contrastive divergence, and the last two layers were stacked by flipping the first two RBMs. The DAE's output layer uses the linear activation function. After RBM pre-training, the DAE was fine-tuned by back-propagation (BP) using the the mean squared error criterion.

After BP fine-tuning, three RBMs were put on top of the DAE, where the bottom one is a Gaussian-Bernoulli RBM and the top one is a Bernoulli-Gaussian RBM. BP fine-tuning was then applied to the combined DAE and RBMs using the 2000 senone labels (in one-hot format) as the target outputs and cross-entropy as the minimization criterion. As shown in Fig. 1, the final DAE–DNN 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. Therefore the BN features have a dimension of 60. The bottleneck layer uses the linear activation function, and all the other hidden layers use sigmoid nonlinearity.

The training set for training the DAE–DNN comprises 7,116 clean (original) utterances from NIST 2006–2010 SREs and their 15dB, 6dB, and 0dB noise contaminated versions, which amount to a total of $7,116 \times 4 = 28,464$ training utterances. These utterances were spoken by 723 target speakers in CC4 of NIST 2012 SRE. The DAE on the left of Fig. 1 was trained to produce the clean MFCCs of the utterances, given the clean or noisy MFCCs as input. The DAE–DNN on the right of Fig. 1 was trained to produce the senone labels of the clean utterances based on the ASR–DNN mentioned in Section III-C.

TABLE I

Performance of various i-vector/PLDA systems on NIST 2012 SRE (CC4, male speaker, core task) with test utterances contaminated with babble noise at different SNRs. DAE–DNN is DNN with DAE training (Fig. 1). DNN is a DNN pre-trained by RBMs. The UBM here refers to a speaker-independent GMM. Denoised MFCC is the MFCC denoised by the DAE in Fig. 1 (left panel). BN features: Bottleneck features obtained from the DAE–DNN (Fig. 2)

		Original			15dB			6dB			0dB		
Acoustic Features	Posteriors from	EER	minDCF	actDCF	EER	minDCF	actDCF	EER	minDCF	actDCF	EER	minDCF	actDCF
MFCC	UBM(1024-mix)	2.62	0.285	0.835	3.58	0.336	0.847	3.27	0.372	0.871	4.84	0.501	0.915
MFCC	UBM(2048-mix)	3.60	0.442	0.969	3.22	0.458	0.966	3.65	0.505	0.976	5.39	0.651	0.989
MFCC	DNN	1.69	0.230	**0.786**	**1.92**	0.281	0.816	2.64	0.324	**0.799**	3.16	0.474	0.878
MFCC	DAE–DNN	1.82	0.253	0.808	2.75	0.273	**0.790**	2.57	0.296	0.816	3.43	0.474	0.861
Denoised MFCC	DAE–DNN	2.46	0.339	0.933	3.45	0.331	0.907	3.74	0.387	0.924	4.73	0.643	0.942
BN Features	DAE–DNN	**1.56**	**0.218**	0.859	2.17	**0.212**	0.799	**2.01**	**0.229**	0.817	**3.07**	**0.432**	**0.852**

TABLE II

Performance of BN-based i-vector/PLDA systems on NIST 2012 SRE (CC4, male speaker, core task) with test utterances contaminated with different levels of babble noise. DAE–DNN is DNN with DAE training (Fig. 1). The UBM here is a speaker-independent GMM trained by using BN features.

	Original		15dB		6dB		0dB	
Posteriors from	EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF
UBM(1024-mix)	3.19	0.357	4.11	0.350	3.73	0.358	4.70	0.484
UBM(2048-mix)	1.97	**0.203**	2.63	0.245	2.58	0.239	3.70	**0.389**
DAE–DNN	**1.56**	0.218	**2.17**	**0.212**	**2.01**	**0.229**	**3.07**	0.432

As the procedure in Section II-D and Fig. 2 show, 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 with the conventional i-vectors.

E. Enrollment and Test Utterances

Because CC4 in 2012 SRE involves noise-contaminated test utterances, this test condition covers a wide range of SNR distribution, and we refer to this test condition as "original". In addition to this "original" test condition, we created three test conditions based on the noise contaminated test utterances by the FaNT tool as mentioned in Section III-A. Specifically, for the 15dB test condition, test utterances added noise at the SNR of 15dB by the FaNT tool were used for scoring, and similarly for the 6dB and 0dB test conditions. For all of the original, 15dB, 6dB and 0dB CC4 test conditions of NIST 2012 SRE, we used the original target-speaker utterances and their noise contaminated counterparts from the 6dB and 15dB SNR groups as enrollment utterances in order to keep consistency. Therefore the enrollment utterances were the same for different test conditions.

We have also performed experiments under CC5 of NIST 2012 SRE, including 62,845 trials in the core test and 1,558,788 trials in the extended test. Unlike the test segments in CC4, the test segments in CC5 were intentionally collected in a noisy environment. Therefore, the noisy speech in CC5 is more realistic. The test segments in CC5 have a wide range of SNRs, from 10dB to over 40 dB. Because most of the test segments in CC5 have SNR over 20dB, we only used the i-vectors of the original enrollment segments (which also have high SNR) to represent the target speakers during the scoring stage.

F. Producing Likelihood-Ratio Scores

The PLDA model for scoring was trained by the the original enrollment utterances and their noise contaminated counterparts from the 0dB, 6dB and 15dB SNR groups. For real-world deployment, it is desirable to have application-independent decision thresholds [46] such that not only the equal error rate (EER) and minimum detection cost (minDCF) are minimized, but also the actual DCF (actDCF) at specific thresholds are also small. To this end, all of the original PLDA scores were subject to score calibration to produce true likelihood-ratio scores using the Bosaris toolkit [47]:

$$S' = w_0 + w_1 S, \tag{11}$$

where S is the original PLDA scores. This calibration step only shifts and scales the original PLDA scores, which reduce the actDCF (primary cost) without affecting the EER and minDCF.

IV. Results and Discussion

Table I shows the EER, minDCF and actDCF of various i-vector/PLDA systems that use different acoustic features and different ways of computing the senone posteriors or mixture posteriors. To study the benefit of DAE training in more details, in addition to the DAE–DNN, we also trained a DNN without DAE pre-training but with RBM pre-training, i.e., all hidden layers in Fig. 2 were initialized by RBMs. We refer to this network as DNN. It has a structure of 220-256-256-256-256-256-256-60-2000.

Surprisingly, a comparison between the first and the second rows of Table I suggests that MFCC-UBM with 2048 mixtures performs worse than the one with 1024 mixtures. Since the same amount of training data was used in these two models, increasing the number of mixtures, i.e., increasing the learning capacity of the model, does not necessarily improve performance. Furthermore, for UBM with 2048 mixtures, the misalignment of speech frames to mixture component would be more severe under noisy conditions, causing further performance degradation. Results from the first three rows of Table I suggest that the i-vectors derived from senone

TABLE III

Performance of various i-vector/PLDA systems on NIST 2012 SRE (CC4, male speaker, core task) with test utterances contaminated with different levels of babble noise. DAE–DNN is DNN with DAE training (Fig. 1). DNN has a similar structure as DAE–DNN, but without DAE training.

		Original		15dB		6dB		0dB	
BN Features from	Posteriors from	EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF
DAE–DNN	DAE–DNN	1.56	0.218	2.17	0.212	**2.01**	0.229	3.07	0.432
DAE–DNN	DNN	1.46	**0.212**	**2.08**	0.205	**2.01**	0.236	**2.90**	0.438
DNN	DAE–DNN	**1.30**	0.220	2.12	0.200	2.05	**0.227**	3.08	**0.425**
DNN	DNN	1.54	**0.212**	2.24	**0.199**	2.04	0.246	3.20	0.435

TABLE IV

Performance of various i-vector/PLDA systems on NIST 2012 SRE (CC5, male speaker, core task). DAE–DNN is DNN with DAE training (Fig. 1). DNN has a similar structure as DAE–DNN, but without DAE training.

		CC5 Core		CC5 Extended	
BN Features from	Posteriors from	EER	minDCF	EER	minDCF
DAE–DNN	DAE–DNN	2.18	**0.251**	**2.15**	**0.248**
DAE–DNN	DNN	**2.07**	0.261	2.16	**0.248**
DNN	DAE–DNN	2.24	0.278	2.19	0.253
DNN	DNN	2.35	0.263	2.68	0.255

posteriors obtained from the DNN outperform the i-vectors whose mixture posteriors $\gamma_c(\boldsymbol{o}_{it})$'s are obtained from the UBM. This confirms the findings in [27], [28], [34] that the phonetic information in the senone i-vectors is beneficial for speaker comparison. The comparison between the 3rd and the 4th rows suggest that the DAE training hurts the prediction of senones except for noisy conditions, such as 6dB. The poor performance of the denoised MFCCs in the 5th rows of Table I may be due to the mismatch between the denoised MFCCs extracted from the DAE (Fig. 1, left network) and the senone posteriors obtained from the DAE–DNN (Fig. 1, right network).

In Table I, the denoised MFCC and the senone posteriors of each frame were extracted from the left- and right-networks of Fig. 1, respectively. After DAE training of the left-network, it was used for initializing the lower part of the right-network. After backpropagation fine-tuning, the right-network is able to produce the senone posteriors given a contextual window of noisy MFCCs as input. For Row 4 of Table I, the right-network was asked to compute the senone posteriors given the noisy MFCCs, which are exactly the network input. As a result, there is a perfect match between the acoustic features (noisy MFCCs) and the senone posteriors. On the other hand, for Row 5 of Table I, the right-network was asked to compute the senone posteriors given the denoised MFCCs, which do not agree with the network input. This causes mismatch between the senone posteriors produced by the network and the acoustic features (denoised MFCCs), which explains why the performance in Row 5 is much poorer than that in Row 4 of Table I. Note that because of the backpropagation fine-tuning, the output of the lower part of the right-network in Fig. 1 cannot be considered as denoised MFCCs. As a result, denoised MFCCs can only be extracted from the left-network.

On the other hand, the last row suggests that the denoised senone i-vectors, in which both the BN features and posteriors are obtained from the DAE–DNN, achieve the best performance under all of the 4 SNR conditions. The good performance of these BN-based senone i-vectors is attributed to the fact that both the BN features and the senone posteriors are obtained from the same network (the DAE–DNN). Therefore, they work very well with each other. We conjecture that there is a compromise between the robustification of BN features by the DAE and the amount of speaker information loss. For the BN-based senone i-vectors, the benefit of the former prevails.

Although we have performed score calibration by using the logistic regression function in the Bosaris toolkit [47], all the systems still have high actDCF. This is mainly caused by the wide range of SNR in the training data for estimating the calibration weights. The training data comprise the original utterances and noise contaminated utterances at 0dB, 6dB and 15dB. More advanced calibration techniques [48], [49] are needed to improve the actDCF performance.

Table II compares the performance of two BN-based i-vector/PLDA systems. In the first two rows, the mixture posteriors $\gamma_c(\boldsymbol{o}_{it})$'s were obtained from the UBM; whereas in the last row, the senone posteriors $\gamma_c(\boldsymbol{s}_{it})$'s were obtained from the DAE–DNN. The BN features were extracted from the DAE–DNN shown in Fig. 2. The performance improves significantly when the number of UBM mixtures increases from 1024 to 2048. Because the DNN has 2000 outputs (each representing a senone), it is reasonable that the BN features have around 2000 clusters in the feature space. Therefore, the UBM with 2048 mixtures is more appropriate for modeling the BN features. On the other hand, each Gaussian in the 1024-mixture UBM requires to model roughly two senone clusters, which limits the performance of the BN i-vectors derived from this UBM. Under all of the 4 SNR conditions, using the posteriors from the DAE–DNN improves performance significantly. Although the performance of the system with senone posteriors drops when the test utterances become noisier, it is still superiors to the one with GMM mixture posteriors. It shows that the DAE–DNN can estimate the senone posteriors

TABLE V
Cross-entropy of DAE–DNN and DNN on the training set mentioned in Section III-D and the noise contaminated test utterances from CC4 of NIST 2012 SRE. DAE–DNN is DNN with DAE training (Fig. 1). DNN has a similar structure as DAE–DNN, but without DAE training.

	Training Set	15dB CC4	6dB CC4
DAE–DNN	**6.32**	6.91	6.97
DNN	6.34	**6.82**	**6.89**

accurately under noisy conditions to some extent.

With DAE–DNN and DNN, we have four combinations of BN features and senone posteriors as shown in Table III. Comparing Row 2 and Row 4 of Table III suggests that DAE training improves the performance of BN features if the posteriors are from the DNN without DAE training. Similarly, comparing Row 3 and Row 4 suggests that DAE training becomes important for estimating the senone posteriors when the BN features are obtained from the DNN without DAE training. However, comparing Row 1 and Row 3 suggests that DAE training is not necessary for extracting the BN features if the posteriors are obtained from the DAE–DNN. On the other hand, Row 1 and Row 2 suggest that DAE training is not necessary for estimating the senone posteriors when the BN features are obtained from the DAE–DNN. In conclusion, the DAE training can benefit the senone i-vector systems if one of the two components (BN feature extraction and posterior computation) receives DAE training. Overall speaking, the senone i-vectors whose phonetically discriminative BN features are obtained from the DAE–DNN and senone posteriors are obtained from the DNN without DAE training (Row 2) achieve the best performance.

The performance of BN-based senone i-vectors under CC5 of NIST 2012 SRE (male speaker) is shown in Table IV. The performance in Row 1 and Row 2 is significantly better than the one in Row 3 and Row 4, which suggests that DAE training benefits the BN features. However, comparisons between Row 1 and Row 2 and between Row 3 and Row 4 of Table IV suggest that DAE training is more effective for cleaning up BN features than for robustifying senone posteriors. This conclusion is consistent with the one under CC4, where the senone i-vectors with BN features from DAE–DNN and senone posteriors from DNN (Row 2 in Table III) achieve the best performance in Table III.

To verify the conclusion that DAE training is less beneficial for senone posteriors estimation, we calculated the cross-entropy of DAE–DNN and DNN on the training set mentioned in Section III-D and on the noise contaminated test utterances from CC4 of NIST 2012 SRE. The results are shown in Table V. The results show that the cross-entropy on the training set is almost the same regardless of whether DAE training is applied. However, with DAE training, the cross-entropies on the noise contaminated test utterances become higher. This suggests that while DAE training can benefit BN feature extraction, it reduces the generalization capability of the network, causing less accurate senone posteriors on the test utterances. This agrees with our conlcusions in Table I, Table III and Table IV.

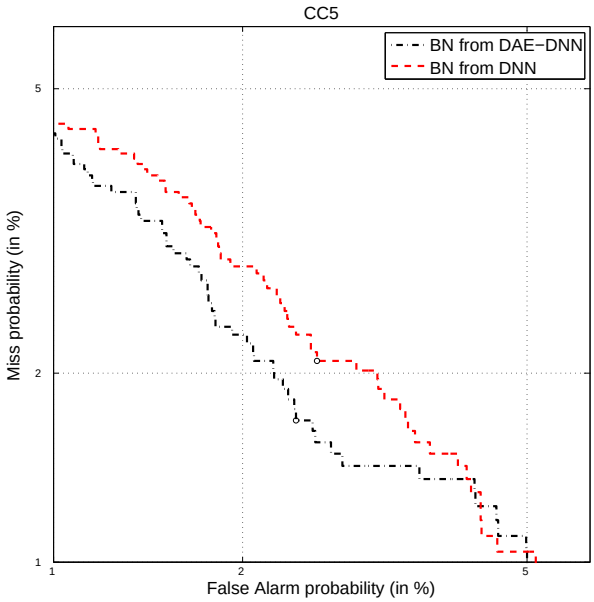


Fig. 4. *The DET performance (CC5 of NIST 2012 SRE) of two senone i-vector systems based on BN features. In the legend, "BN from DAE–DNN" and "BN from DNN" mean that the bottleneck features were obtained from a DNN with and without DAE training, respectively. They correspond to Row 2 and Row 4 in Table IV, respectively. In both cases, the senone posteriors were obtained from the DNN without DAE training.*

The DET curves of the systems corresponding to Row 2 and Row 4 in Table IV are shown in Fig. 4. The senone posteriors of both systems were obtained from a DNN without DAE training. The results clearly show that DAE training is beneficial to BN feature extraction, as the DET curve of "BN from DAE–DNN" is below that of "BN from DNN" for a wide range of decision thresholds.

V. Conclusions and Future Works

This paper has shown that robust BN features and frame posteriors can be obtained from a denoising autoencoder–deep neural network (DAE–DNN) formed by the combination of a denoising autoencoders (DAE) and a deep neural network (DNN). The DAE provides a good initial condition for the backpropagation to find a DNN that can suppress noise in MFCC vectors and enforces the frame alignments to respect the phonetic context of input speech. No matter under the GMM i-vector or the senone i-vector frameworks, the phonetically discriminative BN features outperform MFCCs in the speaker verification tasks, which suggests that the phonetically discriminative BN features still contain speaker information. We also demonstrated that the denoising capability of the DAE is beneficial to the BN features in senone i-vectors but not to the denoised-MFCC-based senone i-vectors.

By comparing the combinations of phonetically discriminative BN features and senone posteriors with and without DAE training, we validated that the DAE training is more beneficial for BN features extraction than senone posteriors estimation. The BN features extracted from DAE–DNN is still rich in speaker information, while the DNN without DAE training is good at estimating the speaker-independent senone

posteriors, leading to better generalization power in senone estimation on the test sets. This might be caused by the different initial points before backpropagation fine-tuning for DAE–DNN and DNN, respectively. The DAE training helps DAE–DNN to keep speaker information until the fifth layer in DAE–DNN, while DNN without DAE training can estimate speaker-independent senone posteriors more accurately, but loses more speaker information because no part of the network is trained to keep speaker information.

Overall speaking, our experiment results show that the demonised senone I-vectors whose BN features and senone posteriors are both extracted from a DAE–DNN are comparable with the one whose senone posteriors are estimated by a DNN, but the system involves only one neural network and thus has the advantage in accelerating the extraction of i-vectors and system implementation.

In our experiments the enrollment data were also used as training data. Although the NIST SRE 2012 protocol allows us to do this. In future work, it is better to use another pool of speakers to train the UBM and TV matrix to verify the generalization capability of the proposed methods.

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

[3] S. Cumani and P. Laface, "Nonlinear i-vector transformations for PLDA-based speaker recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 908–919, April 2017.

[4] H. Xing and J. H. L. Hansen, "Single sideband frequency offset estimation and correction for quality enhancement and speaker recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 124–136, Jan 2017.

[5] R. Saeidi, P. Alku, and T. Backstrom, "Feature extraction using power-law adjusted linear prediction with application to speaker recognition under severe vocal effort mismatch," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 42–53, Jan 2016.

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

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

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

[9] S. Novoselov, T. Pekhovsky, O. Kudashev, V. Mendelev, and A. Prudnikov, "Non-linear PLDA for i-vector speaker verification," *Interspeech*, pp. 214–218, September 2015.

[10] 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.

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

[12] 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 *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2012, pp. 4257–4260.

[13] N. Li and M. W. Mak, "SNR-invariant PLDA modeling in nonparametric subspace for robust speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 10, pp. 1648–1659, 2015.

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

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

[16] 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. PP, no. 99, pp. 1371–1383, 2017.

[17] 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)*, 2016, pp. 4960–4964.

[18] 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, pp. 4955–4959.

[19] 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)*, 2016, pp. 4940–4944.

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

[21] Z. Meng, S. Watanabe, J. R. Hershey, and H. Erdogan, "Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 271–275.

[22] 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.

[23] O. Ghahabi and J. Hernando, "Deep learning backend for single and multisession i-vector speaker recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 807–817, April 2017.

[24] Z. Tang, L. Li, D. Wang, and R. Vipperla, "Collaborative joint training with multitask recurrent model for speech and speaker recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 3, pp. 493–504, March 2017.

[25] W. M. Campbell, "Using deep belief networks for vector-based speaker recognition," in *Proc. Interspeech*, 2014, pp. 676–680.

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

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

[28] 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.

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

[30] W. Geng, J. Li, S. Zhang, X. Cai, and B. Xu, "Multilingual tandem bottleneck feature for language identification," in *Proc. Interspeech*, 2015, pp. 413–417.

[31] 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, pp. 1151–1155.

[32] 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.

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

[34] F. Richardson, D. 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.

[35] A. K. Sarkar, C.-T. Do, V.-B. Le, and C. Barras, "Combination of cepstral and phonetically discriminative features for speaker verification," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1040–1044, 2014.

[36] Z. L. Tan, Y. K. Zhu, M. W. Mak, and B. Mak, "Senone i-vectors for robust speaker verification," in *Int. Sym. on Chinese Spoken Language Processing (ISCSLP'16)*, Oct. 2016.

[37] Z. L. Tan and M. W. Mak, "Bottleneck features from SNR-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.

[38] E. Variani, X. Lei, E. McDermott, I. Lopez-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, pp. 4080–4084.

[39] G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," in *Proc. Interspeech*, 2017, pp. 1517–1521.

[40] 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.

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

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

[43] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al., "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, pp. 153, 2007.

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

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

[46] D. Leeuwen and Niko Brümmer, "The distribution of calibrated likelihood-ratios in speaker recognition," in *Interspeech 2013*, 2013, pp. 1619–1623.

[47] N. Brümmer and E de Villiers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," *arXiv preprint arXiv:1304.2865*, 2013.

[48] T. Hasan, R. Saeidi, J. H. L. Hansen, and D. A. van Leeuwen, "Duration mismatch compensation for i-vector based speaker recognition systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 7663–7667.

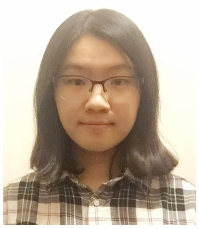
[49] Z. L. Tan and M. W. Mak, "I-vector DNN scoring and calibration for noise robust speaker verification," in *Proc. Interspeech*, 2017, pp. 1562–1566.

Man-Wai MAK (M'93–SM'15) received a Ph.D. in electronic engineering from the University of Northumbria in 1993. He joined the Department of Electronic and Information Engineering at The Hong Kong Polytechnic University in 1993 and is currently an Associate Professor in the same department. He has authored more than 180 technical articles in speaker recognition, machine learning, and bioinformatics. Dr. Mak also coauthored a postgraduate textbook *Biometric Authentication: A Machine Learning Approach*, Prentice-Hall, 2005 and a research monograph *Machine Learning for Protein Subcellular Localization Prediction*, De Gruyter, 2015. He served as a member of the IEEE Machine Learning for Signal Processing Technical Committee in 2005–2007. He has served as an associate editor of IEEE/ACM Transactions on Audio, Speech and Language Processing. He is currently an associate editor of *Journal of Signal Processing Systems* and *IEEE Biometrics Compendium*. He also served as Technical Committee Members of a number of international conferences, including ICASSP and Interspeech, and gave a tutorial on machine learning for speaker recognition in Interspeech'2016. Dr. Mak's research interests include speaker recognition, machine learning, and bioinformatics.

Brian Kan-Wing MAK (SM'10) received the B. Sc. degree in Electrical Engineering from the University of Hong Kong, the M. S. degree in Computer Science from the University of California, Santa Barbara, USA, and the Ph.D. degree in Computer Science from the Oregon Graduate Institute of Science and Technology, Portland, Oregon, USA. He had worked as a research programmer at the Speech Technology Laboratory of Panasonic Technologies Inc. in Santa Barbara, and as a research consultant at the AT&T Labs – Research, Florham Park, New Jersey, USA. He had been a visiting researcher of Bell Laboratories and Advanced Telecommunication Research Institute — International as well. Since April 1998, he has been with the Department of Computer Science in the Hong Kong University of Science and Technology, and is now an Associate Professor.

He had served or is serving on the editorial board of the IEEE Transactions on Audio, Speech and Language Processing, the Signal Processing Letters, and Speech Communication. He also had served on the Speech and Language Technical Committee of the IEEE Signal Processing Society. His interests include acoustic modeling, speech recognition, spoken language understanding, computer-assisted language learning, and machine learning. He received the Best Paper Award in the area of Speech Processing from the IEEE Signal Processing Society in 2004.

Zhili TAN received the B.Eng. and M.Sc. degrees in electronic engineering from the Chinese University of Hong Kong in 2013 and 2014 respectively. Since August 2014, he has been pursuing a Ph.D. degree in electronic and information engineering in the Hong Kong Polytechnic University. His research interest include speaker recognition and machine learning.



Yingke Zhu received the B.Eng degree in Communication Engineering from the Beijing University of Posts and Telecommunications in 2014. Since September 2014, she has been pursuing a Ph.D. degree in Computer Science in the Hong Kong University of Science and Technology. Her research interests include speech recognition, speaker recognition and deep learning.