Senone I-Vectors for Robust Speaker Verification

Zhili TAN\textsuperscript{1}, Yingke ZHU\textsuperscript{2}, Man-Wai MAK\textsuperscript{1}, Brian Kan-Wing MAK\textsuperscript{2}

\textsuperscript{1}Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University
\textsuperscript{2}Dept. of Computer Science and Engineering, The Hong Kong University of Science and Technology

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 \cite{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 \cite{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 \cite{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) \cite{5}. As another example, in \cite{6}, DBNs pre-trained by contrastive divergence \cite{7}, were used to generate the posteriors of the mixtures of a universal background model (UBM).

Inspired by the great success of DNNs \cite{8}, convolutional neural networks (CNNs) \cite{9} and recurrent neural networks (RNNs) \cite{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 \cite{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 \cite{14}. In \cite{15}, the method was extended to replace the MFCCs in \cite{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 \cite{16}.

DNNs are also applicable to restoring spectral vectors for speech enhancement \cite{17, 18} and restoration of unreliable i-vectors in short-utterance speaker recognition \cite{19}. The idea is to use denoising deep autoencoders (DAE) \cite{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 \cite{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 \cite{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 propose 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 balanced, 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 $O_i = \{o_{i1}, \ldots, o_{iT_i}\}$ as a set of $F$-dimensional acoustic feature vectors, which are assumed to follow a mixture distribution:

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

where $p(o_{it}|c)$ is the conditional likelihood of $o_{it}$ and $\pi_c$’s are the mixture weights.
In the conventional i-vector framework, the GMM-supervector representing the t-th utterance is assumed to follow a factor analysis model of the form:

$$\mu_t = \mu^{(b)} + T w_t + \epsilon_t,$$

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

$$\langle w_i | o_i \rangle = L_{ic}^{-1} \sum_c T_{ic} \Sigma_c^{-1} \tilde{f}_{ic},$$

$$\langle w_i w_i^\top | o_i \rangle = L_{ic}^{-1} + \langle w_i | o_i \rangle \langle w_i | o_i \rangle^\top,$$

$$L_{ic}^{-1} = I + T^\top (\Sigma_c)^{-1} N \Sigma_c T, \quad i = 1, \ldots, N;$$

**M-step:**

$$T_{ic} = \left[ \sum_j \tilde{f}_{ic} (\langle w_i | o_i \rangle)^j \right] \left[ \sum_i N_{ic} \langle w_i w_i^\top | o_i \rangle \right]^{-1}.$$

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

$$N_{ic} = \sum_i \gamma_c(o_{ic});$$

$$\tilde{f}_{ic} = \sum_i \gamma_c(o_{ic})(o_{ic} - \mu_c).$$

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

$$\gamma_c(o_{it}) = \frac{\lambda_c^{(b)} N(o_{it} | \mu^{(b)}_c, \Sigma^{(b)}_c)}{\sum_{j=1}^K \lambda_j^{(b)} N(o_{it} | \mu^{(b)}_j, \Sigma^{(b)}_j)};$$

where $\theta = \{\lambda_j^{(b)}, \mu_j^{(b)}, \Sigma_j^{(b)}\}_{j=1}^K$ are UBM parameters.

### 2.2. Generalized i-vector extractor

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

$$\mu_c = \frac{\sum_i \sum_j \gamma_c(o_{it}) o_{it}}{\sum_i N_{ic}},$$

$$\Sigma_c = \frac{\sum_i N_{ic} S_{ic}}{\sum_i N_{ic}},$$

where $S_{ic} = \sum_j \gamma_c(o_{it}) (o_{it} - \mu_c)(o_{it} - \mu_c)^\top$. 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 $o_{it}$ and mixture posteriors $\gamma_c(o_{it})$ are necessary for i-vector extraction.

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

$$\alpha_{it} = f(s_{it}), \quad \gamma_c(s_{it}) = P(c | s_{it}),$$

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

### 2.3. DNN with Denoising Autoencoder

In [13], $P(c | s_{it})$ 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 $P(c | s_{it})$. Furthermore, to enrich the contextual information in $o_{it}$, they are extracted from the bottleneck layer just below the softmax layer of the DNN. More precisely, $f(s_{it})$ 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 $(s_{it})$ as input.

Fig. 1 illustrates the procedures to train our denoising deep classifier. 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

$$y_k(x) = \frac{\epsilon_{hk}}{\sum_{k=1}^K e_i}, \quad k = 1, \ldots, K,$$

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

$$E(Y, Z, C) = -\sum_{k=1}^K \sum_{i=1}^{M_k} \sum_{t=1}^{K_k} z_{i,j,k} \log(y_k(x_{i,j}))$$

where $z_{i,j,k}$’s are one-of-$K$ vectors indicating to which senone the input vector $x_{i,j}$ belongs and $M_k$ 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- and 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. 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: \( \alpha_{it} = \text{BN}(s_{it}) \)
- Senone posteriors: \( \gamma_{t} (s_{it}) = P_{\text{DNN}}(c | s_{it}) \), which is the output of the \( c \)-th node in the softmax output layer.
- Baum-Welch statistics:
  \[
  N_{ic} = \sum_{t} P_{\text{DNN}}(c | s_{it}),
  \]
  \[
  \tilde{f}_{ic} = \sum_{t} P_{\text{DNN}}(c | s_{it})(\text{BN}(s_{it}) - \mu_c),
  \]
  \[
  S_{ic} = \sum_{t} P_{\text{DNN}}(c | s_{it})(\text{BN}(s_{it}) - \mu_c)(\text{BN}(s_{it}) - \mu_c)^T,
  \]
  where:
  \[
  \mu_c = \frac{\sum_{i} S_{ic}}{\sum_{i} N_{ic}},
  \]
  \[
  \Sigma_c = \frac{1}{\sum_{i} N_{ic}} \sum_{i} S_{ic}.
  \]

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 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-
null
5. References


