Robust i-vector extraction for neural network adaptation in noisy environment

Chengzhu Yu\textsuperscript{1,2}, Atsunori Ogawa\textsuperscript{2}, Marc Delcroix\textsuperscript{2}, Takuya Yoshioka\textsuperscript{2},
Tomohiro Nakatani\textsuperscript{2}, John H. L. Hansen\textsuperscript{1}

\textsuperscript{1}The University of Texas at Dallas, Richardson, TX, U.S.A.
\textsuperscript{2}NTT Communication Science Laboratories, NTT corporation

\{chengzhu.yu, john.hansen\}@utdallas.edu,
\{ogawa.atsunori, marc.delcroix, yoshioka.takuya, nakatani.tomohiro\}@lab.ntt.co.jp

Abstract

In this study, we explore an i-vector based adaptation of deep neural network (DNN) in noisy environment. We first demonstrate the importance of encapsulating environment and channel variability into i-vectors for DNN adaptation in noisy conditions. To be able to obtain robust i-vector without losing noise and channel variability information, we investigate the use of parallel feature based i-vector extraction for DNN adaptation. Specifically, different types of features are used separately during two different stages of i-vector extraction namely universal background model (UBM) state alignment and i-vector computation. To capture noise and channel-specific feature variation, the conventional MFCC features are still used for i-vector computation. However, much more robust features such as Vector Taylor Series (VTS) enhanced as well as bottleneck features are exploited for UBM state alignment. Experimental results on Aurora-4 show that the parallel feature-based i-vectors yield performance gains of up to 9.2\% relative compared to a baseline DNN-HMM system and 3.3\% compared to a system using conventional MFCC-based i-vectors.

Index Terms: Automatic speech recognition, Deep neural networks, Acoustic model adaptation, i-vector extraction, Noise robustness

1. Introduction

Recently, deep neural network based acoustical models [1] have significantly surpassed the traditional HMM based models where each state is modeled using Gaussian mixture model (GMM). The success of DNN applications for speech recognition could largely be attributed to its ability to learn invariant features through layer-by-layer non-linear feature transformation [2]. However, even with such inherent robustness, DNN-HMM systems are still subject to variations caused by speaker and environment differences. Therefore, acoustic model adaptation is paramount even for the DNNs.

A number of approaches have been proposed for adapting DNN acoustic model to individual speakers or environments [3–12]. Of these methods, i-vector based input augmentation [7, 13–15] has gained much popularity due to its simplicity and compatibility with other adaptation approaches. The efficacy of the i-vector based adaptation has been verified using both utterance and speaker level i-vectors.

Previous studies on i-vector based DNN adaptation focused primarily on clean conditions [7, 16, 17]. Some studies have examined the i-vector based adaptation on noisy conditions and confirmed its effectiveness, but the i-vector extraction strategy used there was not necessary designed to deal with noisy conditions [13, 15]. A recent study in [14] have also attempted the use of jointly learned speaker and noise factors for DNN adaptation in order to cover wider range of variability. As the i-vectors used in these approaches are all computed as conditional expectation of i-vector given observation features, the characteristics of extracted i-vector would be significantly affected by the presence of noise. However, it has not been well understood how i-vectors should be extracted in noisy environment to improve DNN adaptation performance.

The conventional i-vector extraction [18] method, which was originally developed for speaker recognition, consists of two separate stage: UBM state alignment and i-vector computation. The role of UBM state alignment is to identify and cluster the similar acoustic content, e.g., frames belong to phoneme /AE/. The purpose of such alignment is to allow the following i-vector computation to be less affected by the phonetic variations between features. However, the existence of noise and channel variation could substantially affect the alignment quality and therefore the purity of extracted i-vector.

A straightforward approach to improve the robustness of UBM state alignment is to remove the environment and channel noise from input speech or features before i-vector extraction. While such direct enhancement approaches could improve the UBM state alignment, the environment and channel variability information within extracted i-vectors will be lost at the same time. For the task of speaker recognition, such i-vectors that are insensitive to the environment variability are desirable as only the speaker characteristics are needed to be captured. However, for DNN adaptation in noisy environment, it is helpful or even necessary to preserve the environment and channel variability information within the extracted i-vectors so that the DNN can exploit such information.

In order to make the UBM state alignment stage robust against noise while maintaining the environment and channel variability information within extracted i-vectors, we adopt a parallel feature based i-vector extraction strategy for DNN adaptation in noisy environments. The concept of parallel feature based i-vector extraction was proposed previously in the community of speaker recognition [19–23]. It is based on the fact that UBM state alignment can be performed independently of i-vector computation. Using this i-vector extraction strategy for DNN adaptation, we are able to use the original MFCC features for i-vector computation while using more robust features to perform UBM state alignment. In this paper, we evaluate two robust features, VTS enhanced features [24] and bottleneck features, for UBM state alignment. The bottleneck features [25] are used because of its invariance, which is gained from multiple layers of non-linear processing.
In Sec. 2, we overview the i-vector based DNN adaptation. We also introduce parallel feature based i-vector extraction that we use for DNN adaptation in noisy environment. Section 3 describes experimental results performed to evaluate the proposed approach. Section 4 concludes the paper.

2. I-vector based DNN adaptation

In this section, we present a short overview of the i-vector based DNN adaptation. In this framework, adaptation of DNN is achieved by concatenating i-vectors with input acoustic features e.g., filterbank (FBANK) features. The augmented features are then used for DNN training and prediction of senones. Both speaker and utterance level i-vector can be used for DNN adaptation.

In this study, we focus only on the utterance level i-vector for DNN adaptation as we are interested in representing variability originating from both speaker and environment (noise and channel). The structure of i-vector based DNN adaptation is shown in Fig. 1.

2.1. Conventional i-vector extraction

In conventional i-vector extraction framework, speaker and channel dependent GMM supervector is modeled as follows:

\[ M = m + Tw, \]

where \( m \) is the supervector obtained from the UBM, \( T \) is the low rank total variability matrix representing the basis of reduced total variability space, and \( w \) is the low rank factor loadings referred to as i-vectors.

The estimation of the total variability matrix \( T \) employs expectation maximization (EM) method as described in [26]. After training the total variability matrix, the i-vector of given speech utterance can be represented using Baum-Welch zeroth \((N_s)\) and centralized first \((F_s)\) order statistics:

\[ w^*_s = (T'N_s \Sigma^{-1}T + I)^{-1}T \Sigma^{-1} F_s, \]

where \( \Sigma \) is the covariance matrix obtained from UBM model and \( I \) is the identity matrix. Note that the zeroth \((N_s)\) and centralized first \((F_s)\) order statistics are expressed as

\[ N_s = \begin{bmatrix} N_s^{C=1} & 0 & 0 & 0 \\ 0 & N_s^{C=2} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & N_s^{C=\infty} \end{bmatrix}, \]

\[ F_s = \begin{bmatrix} F_s^{C=1} \\ F_s^{C=2} \\ \vdots \\ F_s^{C=\infty} \end{bmatrix}, \]

where

\[ N_s^{C=\infty} = \sum_t P(c|X_t, \theta_{UBM}), \]

\[ F_s^{C=\infty} = \sum_t P(c|X_t, \theta_{UBM})(X_t - \mu_c). \]

and \( c \) is the index of UBM mixture component, \( X_t \) is acoustic feature at time \( t \), \( \mu_c \) is the mean of \( c \)th Gaussian component. Note that the part \( X_t - \mu_c \) in Eq. (6) computes the variations within the input features.

2.2. Parallel feature based i-vector extraction

From the overview in previous section, we see that the zeroth and centralized first order statistics (see Eq. (5) and Eq. (6)) are two main components for total variability matrix training and i-vector extraction. Of those two, zeroth order statistics are completely determined by the mixture-wise posterior probability \( P(c|X_t, \theta_{UBM}) \) which is the soft alignment between input feature and mixtures of UBM. In parallel feature based i-vector extraction strategy, the expression of zeroth and centralized first order statistics are reformulated as

\[ N_s^{C=\infty} = \sum_t P(c|X_t, \widehat{\theta}_{UBM}), \]

\[ F_s^{C=\infty} = \sum_t P(c|X_t, \widehat{\theta}_{UBM})(X_t - \mu_c). \]

where \( \widehat{\theta}_{UBM} \) is the corresponding UBM trained with \( \widehat{X}_t \). The \( \widehat{X}_t \) is the features used for i-vector computation. It is important to note that the \( \mu_c \) represents the Gaussian means of all features \( X_t \) aligned with \( c \)th mixture of \( \widehat{\theta}_{UBM} \). As the \( \widehat{\theta}_{UBM} \) is trained with feature \( \widehat{X}_t \), the \( \mu_c \) can not be directly obtained from \( \widehat{\theta}_{UBM} \). Instead, \( \mu_c \) is obtained by averaging \( X_t \) weighted by \( P(c|X_t, \widehat{\theta}_{UBM}) \) over all training data.

In this study, we evaluated the use of two robust features as \( \widehat{X}_t \): VTS enhanced feature and bottleneck feature. Both the VTS enhanced and bottleneck features are robust against environment and channel noise. While the robustness of VTS enhanced features are obtained with noise and channel removal,
the invariance of bottleneck features are obtained using layer-
by-layer nonlinear feature transformation using DNN. For an-
other input features $X_t$, we use the noisy MFCC features with-
out any feature enhancement in order to maintain the environ-
ment and channel characteristics for i-vector computation. The
structure of DNN adaptation with parallel feature based i-vector
extraction using bottleneck features and noisy MFCC features
is shown in Fig. 2.

3. Experiments

3.1. Settings

The experiments are performed on Aurora-4 corpus which is
noisy version of Wall Street Journal (WSJ0) corpus. We con-
ducted our experiments on the multi-condition training set in-
cluding 7137 utterances from 83 speakers. Half of the train-
ing utterances are obtained from the primary Sennheiser micro-
phone. The other half are recordings from different secondary
microphones. Part of those utterances are clean speech with-
out noise and the other part are consists of corrupted utterances
with six different noises (street traffic, car, train station, babble,
airport, restaurant) at 10-20 dB SNR. The evaluation set is from
WSJ0 5k-word closed vocabulary test set including 330 test ut-
terances of 8 speakers. The evaluation set consists of two clean
sets recorded with Sennheiser microphone and the secondary
microphones. The types of noise for corrupting evaluation sets
are the same with those in training set but with different SNRs.

The baseline system was built with Kaldi open source toolkit [27]. The GMM-HMM system was trained to have 2026
distinct tied-state triphones using MFCC features along with
their linear discriminant analysis (LDA) and maximum likeli-
hood linear transform (MLLT). The alignment obtained from
GMM-HMM system is then used for training DNN-HMM sys-
tem. For the DNN-HMM systems, we first generatively pre-
train the DNN with 7 layers of stacked RBM with 2048 hidden
nodes in each layer. The DNN-HMM system was trained with
40 dimensional log Mel filterbank (FBANK) features. We use
256 mini-batch and 0.008 as the start learning rate. After each
epoch of training, the learning rate is reduced by half when the
improvements in development set are less than 0.5%. The tri-
gram language model was used for decoding.

The i-vector used in this experiments are extracted in ut-
terance level. The i-vector is set to have relatively small di-
-mensio of 25 in order to avoid overfitting problem observed in
study [13]. The 512 mixture full covariance UBM trained with
the training set is used for i-vector extraction.

3.2. Oracle experiments

To prove the importance of preserving noise and channel vari-
ability information for i-vector based DNN adaptation in noisy
environment, we first conduct a preliminary experiment using
clean MFCC features\(^2\). In this experiment, the clean MFCC,
an instance of perfect feature enhancement, are used instead of
noisy MFCC feature for extracting i-vectors with conventional
i-vector extraction framework. The extracted i-vectors from
clean MFCC features are robust in terms of UBM state align-
ment but the environment and channel variability information
within extracted i-vectors are also removed. The result of this
\(^2\)The corresponding clean speech of Aurora-4 multicondition train-
ing set can be obtained from clean training set. The data from clean
training set is used only for i-vector extraction. The extracted i-vectors
from clean MFCC features are augmented with noisy FBANK features
as the input to DNN system.

\(^3\)The result obtained with clean MFCC features is not the upper
bound of parallel feature based i-vector for DNN adaptation. Higher
performance could be achieved if other variability factors (e.g., speaker
variability) can also be compensated for UBM state alignment.

\(^4\)DNN adaptation with conventional i-vector extracted from noisy
MFCC features

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Improvement (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (DNN)</td>
<td>13.9</td>
</tr>
<tr>
<td>DNN+ivects (noisy MFCC)</td>
<td>12.7</td>
</tr>
<tr>
<td>DNN+ivects (clean MFCC)</td>
<td>16.1</td>
</tr>
<tr>
<td>DNN+ivects (clean MFCC ∪ noisy MFCC)</td>
<td>13.1</td>
</tr>
</tbody>
</table>

Table 1: Results of using clean MFCC for conventional i-vector extraction framework against its use for parallel feature based
i-vector extraction. The feature on the left of the symbol $\oplus$ is the type of feature used for UBM state alignment. The features
on the right of the symbol $\oplus$ indicates the type of feature used for i-vector computation.

\(^3\)The result obtained with clean MFCC features is not the upper
bound of parallel feature based i-vector for DNN adaptation. Higher
performance could be achieved if other variability factors (e.g., speaker
variability) can also be compensated for UBM state alignment.

\(^4\)DNN adaptation with conventional i-vector extracted from noisy
MFCC features

3.3. Robust UBM state alignment

While the previous experiments demonstrate the importance of
maintaining environment and channel variability within i-vector
using oracle experiments, in this experiment we use realistic ro-
bust features to perform DNN adaptation. We evaluate the use
of VTS enhanced and bottleneck features for UBM state align-
ment under the framework of parallel feature based i-vector ex-
traction. The bottleneck features are obtained from the output
of last hidden layer of DNN. The structure and the learning ap-
proach for obtaining bottleneck features are the same with train-
ing DNN for acoustical model except the size of last hidden
layer. The result of this experiment is shown in Table 2. The
dimension of bottleneck features used in this experiment is 120.
The results show that both VTS enhanced and bottleneck fea-
ture perform better than traditional i-vector based DNN adap-
tation with only MFCC features. It can also be observed that
using bottleneck feature for both UBM state alignment and i-
vector computation decrease the performance due to the loss of
environment and channel variability within extracted i-vectors.
This result indicates that improving UBM state alignment with-
out losing environment and channel variability could improve
the performance of i-vector based DNN adaptation in noisy en-
vironment. The best result of i-vector based adaptation obtained
using bottleneck feature under parallel feature based i-vector
extraction are shown in Table 1. It can be observed that the per-
formance drops significantly when using clean MFCC features
for both UBM state alignment and i-vector extraction. The de-
crease of performance is mainly attributed to the loss of chan-
nel and noise variability information within i-vectors. To fur-
ther prove this, we performed another experiment using parallel
feature based i-vector extraction. In this setup, the same clean
MFCC features are still used but only for UBM state alignment
while the noisy MFCC features are applied for i-vector compu-
tation. As we could see from the result in Table 1, the re-
results of using parallel feature based i-vector extraction scheme
gives much better results than using clean MFCC feature for
both UBM state alignment and i-vector computation. More-
over, the relative improvement over conventional i-vector based
DNN adaptation with noisy MFCC features indicates that it is
worthwhile to pursue robust UBM state alignment under paral-
lel feature based i-vector extraction scheme\(^4\).
extraction. The performance of using different size of bottleneck features are also illustrated in Table 3.

Table 2: Results of using parallel feature based i-vector for DNN adaptation. The features on the left of the symbol ⊕ is the type of feature used for UBM state alignment. The features on the right of the symbol ⊕ indicates the type of feature used for i-vector computation.

<table>
<thead>
<tr>
<th>Bottleneck Size</th>
<th>Baseline (DNN)</th>
<th>DNN+ivects (noisy MFCC)</th>
<th>DNN+ivects (Bottleneck)</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>13.9</td>
<td>13.1</td>
<td>13.3</td>
</tr>
<tr>
<td>80</td>
<td>12.7</td>
<td>13.0</td>
<td>13.0</td>
</tr>
<tr>
<td>120</td>
<td>12.9</td>
<td>13.0</td>
<td>12.7</td>
</tr>
<tr>
<td>160</td>
<td>12.7</td>
<td>13.0</td>
<td>13.3</td>
</tr>
</tbody>
</table>

Table 3: Results of using different size of bottleneck features for doing UBM state alignment during i-vector extraction.

<table>
<thead>
<tr>
<th>Bottleneck Size</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>12.9</td>
</tr>
<tr>
<td>80</td>
<td>13.0</td>
</tr>
<tr>
<td>120</td>
<td>12.7</td>
</tr>
<tr>
<td>160</td>
<td>13.3</td>
</tr>
</tbody>
</table>

4. Conclusion

In this study, we investigated robust i-vector extraction for DNN adaptation. We first show that direct speech and feature enhancement before i-vector extraction could not improve the DNN adaptation performance due to the loss of environment and channel variation information which is necessary for i-vector based DNN adaptation in noisy environment. To resolve this problem, we investigate the parallel feature based i-vector extraction for obtaining robust UBM state alignment while maintaining necessary variability information needed for environment adaptation. To perform UBM state alignment in a robust way, VTS enhanced features and bottleneck features were used and evaluated under the framework of parallel feature based i-vector extraction. The results show that the use of these features resulted in performance improvement compared to conventional i-vectors derived from MFCCs.

5. References


