A NEW I-VECTOR APPROACH AND ITS APPLICATION TO IRRELEVANT VARIABILITY NORMALIZATION BASED ACOUSTIC MODEL TRAINING

Yu Zhang[1,2], Zhi-Jie Yan[2], Qiang Huo[2]

[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[2]Microsoft Research Asia, Beijing, China

sjtuzy@gmail.com, {zhijiey, qianghuo}@microsoft.com

ABSTRACT

This paper presents a new approach to extracting a low-dimensional i-vector from a speech segment to represent acoustic information irrelevant to phonetic classification. Compared with the traditional i-vector approach, a full factor analysis model with a residual term is used. New procedures for hyperparameter estimation and i-vector extraction are derived and presented. The proposed i-vector approach is applied to acoustic sniffing for irrelevant variability normalization based acoustic model training in large vocabulary continuous speech recognition. Its effectiveness is confirmed by experimental results on Switchboard-1 conversational telephone speech transcription task.

Index Terms— i-vector, acoustic model, irrelevant variability normalization, unsupervised adaption, LVCSR

1. INTRODUCTION

Recently, a so-called i-vector approach [1] was proposed to extract a low-dimensional feature vector from a speech segment to represent speaker information, which has been successfully applied to speaker verification and become popular in speaker recognition community (e.g., [2, 11]). In [1], important information on how to estimate hyperparameters (a.k.a. total variability matrix [1]) was missing and readers were referred to [7] for such technical details instead. However, because so-called "Baum-Welch" statistics (instead of "Viterbi" ones) were used to extract an i-vector from each speech segment, the theoretical justification and derivation in [7] cannot be used to justify the practice in [1] for both i-vector extraction and hyperparameter estimation. In [18], we explain the theoretical justification of the i-vector extraction approach borrowed from [1] and present our version of hyperparameter estimation procedure. In [2, 11], readers were referred to [6] for technical details of hyperparameter estimation, but it seems the method used in [2] for hyperparameter estimation is the same as we did and described in [18]. In [18], an i-vector based approach was applied to clustering training data so that multiple sets of acoustic models can be trained to improve speech recognition accuracy. In [15], an i-vector based approach was used for acoustic sniffing in irrelevant variability normalization (IVN) based acoustic model training (e.g., [4, 5, 13, 17]) for large vocabulary continuous speech recognition (LVCSR). In all of the above work, a simplified factor analysis model without residual term is used. In this paper, we extend the i-vector approach by using a full factor analysis model with a residual term. New procedures for hyperparameter estimation and i-vector extraction are derived and presented. The proposed i-vector approach is applied to acoustic sniffing for IVN-based acoustic model training in LVCSR.

The rest of the paper is organized as follows. In Section 2, we present the formulation of the new i-vector approach. In Section 3, we describe how we apply i-vector approach to IVN-based framework. In Section 4, we report experimental results on Switchboard-1 conversational telephone speech transcription task. Finally, we conclude the paper in Section 5.

2. NEW I-VECTOR APPROACH

2.1. Data Model

Suppose we are given a set of training data denoted as $\mathcal{Y} = \{\boldsymbol{Y}_i | i = 1, 2, \ldots, I\}$, where $\boldsymbol{Y}_i = (\boldsymbol{y}_1^{(i)}, \boldsymbol{y}_2^{(i)}, \ldots, \boldsymbol{y}_{T_i}^{(i)})$ is a sequence of D-dimensional feature vectors extracted from the i-th training speech segment. From $\mathcal{Y}$, a Gaussian mixture model can be trained using a maximum likelihood approach to serve as a so-called Universal Background Model (UBM):

$$p(\boldsymbol{y}) = \sum_{k=1}^{K} c_k \mathcal{N}(\boldsymbol{y}; \boldsymbol{m}_k, \boldsymbol{R}_k) \tag{1}$$

where c_k's are mixture coefficients, $\mathcal{N}(\cdot; \boldsymbol{m}_k, \boldsymbol{R}_k)$ is a normal distribution with a D-dimensional mean vector $\boldsymbol{m}_k$ and a $D \times D$ diagonal covariance matrix $\boldsymbol{R}_k$. Let $\boldsymbol{M}_0$ denote the $(D \cdot K)$-dimensional supervector by concatenating the $\boldsymbol{m}_k$'s

This work was done when Yu Zhang was intern in Microsoft Research Asia, Beijing, China.

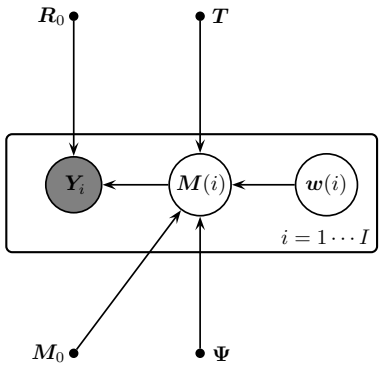

Fig. 1. *A graphical model representation of our new i-vector approach.*

and $\boldsymbol{R}_0$ denote the $(D \cdot K) \times (D \cdot K)$ block-diagonal matrix with $\boldsymbol{R}_k$ as its k-th block component. Let's use $\Omega = \{c_k, \boldsymbol{m}_k, \boldsymbol{R}_k | k = 1, \ldots, K\}$ to denote the set of UBM-GMM parameters.

2.2. i-Vector Extraction

Given a speech segment $\boldsymbol{Y}_i$, let's use a $(D \cdot K)$-dimensional random supervector $\boldsymbol{M}(i)$ to characterize its variability independent of linguistic content, which relates to $\boldsymbol{M}_0$ according to the following full factor analysis model:

$$\begin{cases} \boldsymbol{M}(i) = \boldsymbol{M}_0 + \boldsymbol{T}\boldsymbol{w}(i) + \boldsymbol{\epsilon}(i), \\ \boldsymbol{w}(i) \sim \mathcal{N}(\cdot; \boldsymbol{0}, \boldsymbol{I}), \boldsymbol{\epsilon}(i) \sim \mathcal{N}(\cdot; \boldsymbol{0}, \boldsymbol{\Psi}), \end{cases} \tag{2}$$

where $\boldsymbol{T}$ is a fixed but unknown $(D \cdot K) \times F$ rectangular matrix of low rank (i.e., $F \ll (D \cdot K)$), $\boldsymbol{w}(i)$ is an F-dimensional random vector, $\boldsymbol{\epsilon}(i)$ is a $(D \cdot K)$-dimensional random vector, and $\boldsymbol{\Psi} = \mathrm{diag}\{\psi_1, \psi_2, \ldots, \psi_{DK}\}$ is a positive definite diagonal matrix. A graphical model representation is shown in Fig. 1. In [1], $\boldsymbol{T}$ is called the total variability matrix. Different from [1], we add a residual term $\boldsymbol{\epsilon}$ to model the variabilities not captured by the total variability matrix.

Given $\boldsymbol{Y}_i$, Ω, $\boldsymbol{T}$ and $\boldsymbol{\Psi}$, the i-vector is defined as the solution of the following optimization problem:

$$\hat{\boldsymbol{w}}(i) = \underset{\boldsymbol{w}(i)}{\mathrm{argmax}} \prod_{t=1}^{T_i} \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{y}_t^{(i)}; \boldsymbol{M}_k(i), \boldsymbol{R}_k)^{P(k|\boldsymbol{y}_t^{(i)}, \Omega)} p(\boldsymbol{w}(i)) \tag{3}$$

where $\boldsymbol{M}_k(i)$ is the k-th D-dimensional subvector of $\boldsymbol{M}(i)$, and

$$P(k|\boldsymbol{y}_t^{(i)}, \Omega) = \frac{c_k \mathcal{N}(\boldsymbol{y}_t^{(i)}; \boldsymbol{m}_k, \boldsymbol{R}_k)}{\sum_{l=1}^{K} c_l \mathcal{N}(\boldsymbol{y}_t^{(i)}; \boldsymbol{m}_l, \boldsymbol{R}_l)}.$$

The closed-form solution of the above problem gives the i-vector extraction formula as follows:

$$\hat{\boldsymbol{w}}(i) = \boldsymbol{\zeta}^{-1} \boldsymbol{T}^\top \boldsymbol{\gamma}^{-1} \boldsymbol{\Psi}^{-1} \boldsymbol{R}^{-1} \boldsymbol{\Gamma}_{\boldsymbol{y}}(i) \tag{4}$$

where

$$\boldsymbol{\zeta} = (\boldsymbol{I} + \boldsymbol{T}^\top (\boldsymbol{\Psi} + \boldsymbol{\Gamma}(i)^{-1} \boldsymbol{R})^{-1} \boldsymbol{T})^{-1} \tag{5}$$

$$\boldsymbol{\gamma} = \boldsymbol{\Gamma}(i) \boldsymbol{R}^{-1} + \boldsymbol{\Psi}^{-1}. \tag{6}$$

In the above equations, $\boldsymbol{\Gamma}(i)$ is a $(D \cdot K) \times (D \cdot K)$ block-diagonal matrix with $\gamma_k(i) \boldsymbol{I}_{D \times D}$ as its k-th block component; $\boldsymbol{\Gamma}_{\boldsymbol{y}}(i)$ is a $(D \cdot K)$-dimensional supervector with $\Gamma_{\boldsymbol{y},k}(i)$ as its k-th D-dimensional subvector. The "Baum-Welch" statistics $\gamma_k(i)$ and $\Gamma_{\boldsymbol{y},k}(i)$ are calculated as follows:

$$\gamma_k(i) = \sum_{t=1}^{T_i} P(k|\boldsymbol{y}_t^{(i)}, \Omega) \tag{7}$$

$$\Gamma_{\boldsymbol{y},k}(i) = \sum_{t=1}^{T_i} P(k|\boldsymbol{y}_t^{(i)}, \Omega)(\boldsymbol{y}_t^{(i)} - \boldsymbol{m}_k) \, . \tag{8}$$

2.3. Hyperparameter Estimation

Given the training data $\mathcal{Y}$ and the pre-trained UBM-GMM Ω, the hyperparameters $\boldsymbol{T}$ and $\boldsymbol{\Psi}$ can be estimated by maximizing the following objective function:

$$\mathcal{F}(\boldsymbol{T}, \boldsymbol{\Psi}) = \prod_{i=1}^{I} \int p(\boldsymbol{Y}_i | \boldsymbol{M}(i)) p(\boldsymbol{M}(i) | \boldsymbol{T}, \boldsymbol{\Psi}) \mathrm{d}\boldsymbol{M}(i). \tag{9}$$

Although it is possible to use variational Bayesian approach to solve the above problem, for simplicity, we use the following approximation to ease the problem:

$$p(\boldsymbol{Y}_i | \boldsymbol{M}(i)) \simeq \prod_{t=1}^{T_i} \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{y}_t^{(i)}; \boldsymbol{M}_k(i), \boldsymbol{R}_k)^{P(k|\boldsymbol{y}_t^{(i)}, \Omega)}.$$

Consequently, an EM-like algorithm can be used to solve the above simplified problem. The procedure for estimating $\boldsymbol{T}$ and $\boldsymbol{\Psi}$ is described as follows:

Step 1: *Initialization*

Set the initial value of each element in $\boldsymbol{T}$ randomly from $[Th_1, Th_2]$ and the initial value of each element in $\boldsymbol{\Psi}$ randomly from $[Th_3, Th_4] + Th_5$, where Th_1, Th_2, $Th_3 \geq 0$, $Th_4 > 0$, and $Th_5 > 0$ are five control parameters. For each training speech segment, calculate the corresponding "Baum-Welch" statistics as in Eqs. (7) and (8).

Step 2: *E-step*

For each training speech segment $\boldsymbol{Y}_i$, calculate the posterior expectation of the relevant terms using the sufficient statistics and the current estimation of $\boldsymbol{T}$ and $\boldsymbol{\Psi}$ as follows:

$$E[\boldsymbol{w}(i)] = \boldsymbol{\zeta}^{-1} \boldsymbol{T}^\top \boldsymbol{\gamma}^{-1} \boldsymbol{\Psi}^{-1} \boldsymbol{R}^{-1} \boldsymbol{\Gamma}_{\boldsymbol{y}}(i)$$

$$E[\boldsymbol{\epsilon}(i)] = \boldsymbol{\gamma}^{-1} (-\boldsymbol{\beta}^\top \boldsymbol{\zeta}^{-1} \boldsymbol{T}^\top \boldsymbol{\gamma}^{-1} \boldsymbol{\Psi}^{-1} + \boldsymbol{I}) \boldsymbol{R}^{-1} \boldsymbol{\Gamma}_{\boldsymbol{y}}(i)$$

$$E[\boldsymbol{w}(i)\boldsymbol{w}(i)^\top] = E[\boldsymbol{w}(i)] E[\boldsymbol{w}(i)^\top] + \boldsymbol{\zeta}^{-1}$$

$$E[\boldsymbol{\epsilon}(i)\boldsymbol{\epsilon}(i)^\top] = E[\boldsymbol{\epsilon}(i)] E[\boldsymbol{\epsilon}(i)^\top] + \boldsymbol{\gamma}^{-1} (\boldsymbol{I} + \boldsymbol{\beta}^\top \boldsymbol{\zeta}^{-1} \boldsymbol{\beta} \boldsymbol{\gamma}^{-1})$$

$$E[\boldsymbol{\epsilon}(i)\boldsymbol{w}(i)^\top] = E[\boldsymbol{\epsilon}(i)] E[\boldsymbol{w}(i)^\top] - \boldsymbol{\gamma}^{-1} \boldsymbol{\beta}^\top \boldsymbol{\zeta}^{-1}$$

where $\boldsymbol{\zeta}$ and $\boldsymbol{\gamma}$ are defined in Eqs. (5) and (6), and

$$\boldsymbol{\beta} = \boldsymbol{T}^\top \boldsymbol{R}^{-1} \boldsymbol{\Gamma}(i) \ .$$

Step 3: *M-step*

Update $\boldsymbol{\Psi}$ directly as follows:

$$\boldsymbol{\Psi} = \frac{1}{I} \sum_{i=1}^{I} E[\boldsymbol{\epsilon}(i)\boldsymbol{\epsilon}(i)^\top] \tag{10}$$

and solve the following equation to update $\boldsymbol{T}$:

$$\begin{aligned} &\sum_{i=1}^{I} \boldsymbol{\Gamma}(i) \boldsymbol{T} E[\boldsymbol{w}(i)\boldsymbol{w}(i)^\top] \\ &= \sum_{i=1}^{I} (\boldsymbol{\Gamma}_{\boldsymbol{y}}(i) E[\boldsymbol{w}(i)^\top] - \boldsymbol{\Gamma}(i) E[\boldsymbol{\epsilon}(i)\boldsymbol{w}(i)^\top]). \end{aligned} \tag{11}$$

Step 4: *Repeat or stop*

Repeat **Step 2** to **Step 3** for a fixed number of iterations or until the objective function in Eq. (9) converges.

3. I-VECTOR APPROACH TO ACOUSTIC SNIFFING FOR IVN-BASED TRAINING

3.1. Feature Extraction using LDA

As described above, given the training corpus, a raw F-dimensional i-vector can be extracted from each training speech segment. If meta data (e.g., speaker ID in our experiments) for each speech segment is available, this information can be used (e.g., each speaker ID can be used as a class label in our experiments) to train an $F_1 \times F$ LDA transform matrix, which can be used to transform each raw i-vector into a lower dimensional (i.e., $F_1 \leq F$) yet more discriminative feature space.

3.2. Acoustic Condition Clustering using i-Vectors

Given the set of raw or LDA-transformed training i-vectors, we use a hierarchical divisive clustering algorithm, namely LBG algorithm [9], to cluster them into multiple clusters. Either a Euclidean distance is used to measure the dissimilarity between two i-vectors, $\hat{\boldsymbol{w}}(i)$ and $\hat{\boldsymbol{w}}(j)$, or a cosine measure is used to measure the similarity between two i-vectors. In the latter case, we normalize each i-vector to have a unit norm so that the following cosine similarity measure can be used:

$$sim(\hat{\boldsymbol{w}}(i), \hat{\boldsymbol{w}}(j)) = \hat{\boldsymbol{w}}(i)^\top \hat{\boldsymbol{w}}(j). \tag{12}$$

Furthermore, given the above cosine similarity measure, it can be proven that the centroid, $\boldsymbol{c}^{(w)}$, of a cluster consisting of n unit-norm vectors, $\hat{\boldsymbol{w}}(1), \hat{\boldsymbol{w}}(2), \ldots, \hat{\boldsymbol{w}}(n)$, can be calculated as follows:

$$\begin{aligned} \boldsymbol{c}^{(w)} &= \underset{\boldsymbol{c}}{\operatorname{argmax}} \sum_{i=1}^{n} sim(\hat{\boldsymbol{w}}(i), \boldsymbol{c}) \\ &= \begin{cases} \frac{\sum_{i=1}^{n} \hat{\boldsymbol{w}}(i)}{\| \sum_{i=1}^{n} \hat{\boldsymbol{w}}(i) \|} & \text{if } \sum_{i=1}^{n} \hat{\boldsymbol{w}}(i) \neq 0 \\ 0 & \text{otherwise} \end{cases} . \end{aligned} \tag{13}$$

After the convergence of the LBG clustering algorithm, we obtain E clusters of i-vectors with their centroids denoted as $\boldsymbol{c}_1^{(w)}, \boldsymbol{c}_2^{(w)}, \ldots, \boldsymbol{c}_E^{(w)}$, respectively. Then the speech segments in training set can be distributed to different clusters according to the one-to-one relationship with the corresponding i-vectors. By doing so, all the feature vectors from the same cluster will share a single linear feature transform in IVN-based acoustic model training (to be explained in the next subsection) and the total number of feature transforms equals the number of clusters.

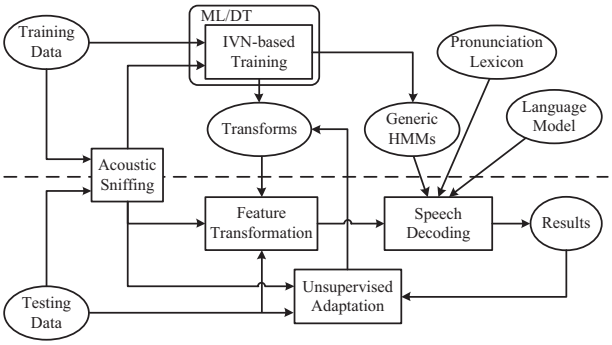

Fig. 2. *An illustration of IVN-based framework for acoustic modeling, training and adaptation.*

3.3. i-Vector Approach to Acoustic Sniffing for IVN-based Training

In a state-of-the-art LVCSR system, robust acoustic model is usually trained by using a large amount of diversified training utterances. However, due to various kind of variabilities (e.g., speakers, environments, channels), conventional model training procedures may lead to a set of diffused models fitting the variabilities irrelevant to phonetic classification. To address this problem, an IVN-based approach can be used (e.g., [4, 5, 13, 17]). Fig. 2 illustrates how it works for acoustic modeling, training and adaptation. In the offline training stage (upper part), a set of feature transforms along with the generic Hidden Markov Models (HMMs) are trained using a Maximum Likelihood (ML) [4, 13] or Discriminative Training (DT) [17] criterion. The feature transforms are used to normalize the irrelevant variabilities of different acoustic conditions. Given a speech segment (e.g., several frames of speech, an utterance, or several utterances), the "acoustic sniffing" module is responsible for detecting the

corresponding acoustic condition and choosing the most appropriate transform(s) accordingly. In the recognition stage (lower part), given an unknown speech segment, the "acoustic sniffing" module is used again for choosing the pre-trained IVN transform(s). The transformed feature vector sequence is then decoded using a conventional LVCSR decoder. After the first-pass recognition, unsupervised adaptation can be performed to adapt the selected feature transform(s). Therefore, an improved recognition accuracy can be achieved in the second-pass decoding.

In this study, the following feature transformation (FT) function is used:

$$\boldsymbol{x}_t = \mathcal{F}(\boldsymbol{y}_t; \boldsymbol{\Theta}) = \boldsymbol{A}^{(e)}\boldsymbol{y}_t + \boldsymbol{b}^{(e)} \tag{14}$$

where $\boldsymbol{y}_t$ is the t-th D-dimensional feature vector of the input feature vector sequence $\boldsymbol{Y}$; $\boldsymbol{x}_t$ is the transformed feature vector; e is a label (transform index) informed by the "Acoustic Sniffing" module for the $D \times D$ nonsingular transformation matrix $\boldsymbol{A}^{(e)}$ and D-dimensional bias vector $\boldsymbol{b}^{(e)}$; and $\boldsymbol{\Theta} = \{\boldsymbol{A}^{(e)}, \boldsymbol{b}^{(e)} | e = 1, 2, \cdots, E\}$ denotes the set of feature transformation parameters with E being the total number of tied linear transforms. For the convenience of notation, we also use hereinafter $\mathcal{F}(\boldsymbol{Y}; \boldsymbol{\Theta})$ to denote the transformed version of a speech segment $\boldsymbol{Y}$ by transforming individual feature vector $\boldsymbol{y}_t$ of $\boldsymbol{Y}$ as defined in Eq. (14).

In IVN-based framework, the acoustic sniffing module is essential for both training and recognition. As mentioned previously, in [15], the old i-vector based approach was used for acoustic sniffing and promising results were achieved. In this study, we compare the effectiveness of the newly proposed i-vector approach with the old one in this context. Given a speech segment $\boldsymbol{Y}$, i-vector based acoustic sniffing can be done as follows:

Step 1: Calculate Baum-Welch sufficient statistics defined by Eqs. (7) and (8) using UBM-GMM.

Step 2: Extract an i-vector from $\boldsymbol{Y}$ using the calculated sufficient statistics and the pre-trained hyperparameters $\boldsymbol{T}$ and $\boldsymbol{\Psi}$. Do LDA feature transformation if applicable. Further normalize the i-vector to have a unit norm if cosine similarity measure is used. Let's use $\hat{\boldsymbol{w}}$ to denote the final processed i-vector.

Step 3: Classify the i-vector $\hat{\boldsymbol{w}}$ into a cluster, e, as follows:

- If Euclidean distance is used as a dissimilarity measure,

$$e = \underset{l=1,2,\ldots,E}{\operatorname{argmin}} \; EuclideanDistance(\hat{\boldsymbol{w}}, \boldsymbol{c}_l^{(w)});$$

- If cosine similarity measure is used,

$$e = \underset{l=1,2,\ldots,E}{\operatorname{argmax}} \; sim(\hat{\boldsymbol{w}}, \boldsymbol{c}_l^{(w)}) \; .$$

The pre-trained linear feature transform from the corresponding cluster e will be used for feature transformation.

The same acoustic sniffing procedure is used in both training and recognition stages.

Let's assume that each basic speech unit in our speech recognizer is modeled by a Gaussian mixture continuous density HMM (CDHMM), whose parameters are denoted as λ. Let $\boldsymbol{\Lambda} = \{\lambda\}$ denote the set of CDHMM parameters. By using the above acoustic sniffing technique, a set of labels for linear transforms $\mathcal{E} = \{e_i | i = 1, 2, \ldots, I\}$ can be derived from the training data $\mathcal{Y}$. The IVN-based ML training is to maximize, by adjusting feature transform parameters $\boldsymbol{\Theta}$ and HMM parameters $\boldsymbol{\Lambda}$, the following likelihood function

$$\begin{aligned} F(\Theta, \Lambda) &= \prod_{i=1}^{I} p(\boldsymbol{Y}_i | \boldsymbol{\Theta}, \boldsymbol{\Lambda}, \mathcal{E}) \\ &= \prod_{i=1}^{I} \{p(\mathcal{F}(\boldsymbol{Y}_i; \boldsymbol{\Theta}) | \boldsymbol{\Lambda}) \cdot |\det(\boldsymbol{A}^{(e_i)})|^{T_i} \end{aligned} \tag{15}$$

where e_i is the acoustic condition label identified by acoustic sniffing for the training speech segment $\boldsymbol{Y}_i$. A *method of alternating variables* can then be used to maximize the above objective function as described in [4, 13].

4. EXPERIMENTS AND RESULTS

4.1. Experimental Setup

Switchboard-1 conversational telephone speech transcription task [3] was used in our experiments. We used 4,870 sides of conversations (about 300 hours of speech) from 520 speakers in training, and 40 sides of conversations (about 2 hours of speech) from the 2000 Hub5 evaluation for testing. The minimum, maximum and average lengths of the utterances are 0.21s, 21.02s, and 4.47s in the training set and 0.53s, 15.50s, and 4.01s in the testing set, respectively.

For front-end feature extraction, we used 39 PLP_E_D_A (in HTK's terminology [16]) features. Conversation-side based mean and variance normalization was applied for both training and testing utterances. For acoustic modeling, we used phonetic decision tree based tied-state triphone GMM-HMMs with 9,302 states and 40 Gaussian components per state. Our recognition vocabulary contained 22,641 unique words. The pronunciation lexicon contained multiple pronunciations per word with a total of 28,649 unique pronunciations. A trigram language model trained on the transcription of the Switchboard-1 training data and broadcast news data was used in decoding. All of the recognition experiments were performed with a Microsoft in-house decoder as in [17] and the results were evaluated by using the NIST Scoring Toolkit SCTK [12]. Our ML-trained baseline system achieves a word error rate (WER) of 30.0%.

Table 1. Comparison of two i-vector based approaches for utterance-based acoustic condition clustering by using average speaker purity (in %) as a quality measure of clustering result on training set.

(Dis)similarity Measure	Cosine		Euclidean	
i-Vector Approach	New	Old	New	Old
No LDA, $F = 600$	37.8	36.8	38.7	35.2
LDA, $F_1 = 600$	58.6	51.5	57.5	55.0
LDA, $F_1 = 400$	51.0	50.3	51.5	50.0
LDA, $F_1 = 200$	41.2	38.1	44.9	43.0

Table 2. Comparison of two i-vector based approaches for IVN-based ML training by using recognition word error rate (WER in %) as performance metric. Our ML-trained baseline system achieves a WER of 30.0%.

(Dis)similarity Measure	Cosine		Euclidean	
i-Vector Approach	New	Old	New	Old
No LDA, $F = 600$	27.1	27.3	27.1	27.3
LDA, $F_1 = 600$	26.7	26.8	26.7	26.9
LDA, $F_1 = 400$	26.5	27.0	26.6	26.9
LDA, $F_1 = 200$	26.8	27.5	27.0	27.4

For each speech utterance in both training and testing data, two raw i-vectors are extracted by using the new and old i-vector approaches, respectively. The settings of relevant control parameters are as follows: The number of UBM-GMM components $K = 1,024$; The dimension of raw i-vector $F = 600$; The number of iterations for updating $\boldsymbol{T}$ and $\boldsymbol{\Psi}$ is 15; The thresholds for initializing $\boldsymbol{T}$ and $\boldsymbol{\Psi}$ are set as $Th_1 = Th_3 = 0, Th_2 = Th_4 = 0.01, Th_5 = 0.001$ under the guidance of the dynamic range of the variance values in UBM-GMM. It is noted that too large initial values may lead to numerical problems in training $\boldsymbol{T}$.

To handle large-scale training data, the hyperparameter estimation tool for i-vector extraction, tools for LBG clustering and GMM training have been implemented based upon MSR Asia's HPC-based speech training platform. This training platform was developed on top of Microsoft Windows HPC Server, and optimized for various speech training and other machine learning algorithms. With this high-performance parallel computing platform, we can run experiments very efficiently for large-scale tasks.

4.2. Comparison of i-Vector Approaches for Acoustic Condition Clustering

For Switchboard-1 corpus, the speaker variability is probably the primary factor we need to deal with. To compare the effect of new and old i-vector approaches for acoustic condition clustering, we use the following Average Speaker Purity (ASP) criterion adapted from [8] to measure the quality of clustering result:

$$ASP = \frac{\sum_{s=1}^{S} p_s \cdot n_s}{\sum_{s=1}^{S} n_s} \tag{16}$$

where S is the number of speakers, n_s is the number of utterances spoken by the speaker s, and p_s is the speaker purity for the speaker s defined as

$$p_s = \sum_{e=1}^{E} \frac{n_{es}^2}{n_s^2} \tag{17}$$

with n_{es} being the number of utterances in cluster e spoken by the speaker s. The higher the ASP, the lesser the degree of splitting utterances from the same speaker across multiple clusters.

Table 1 gives a comparison of the new and old i-vector approaches for utterance-based acoustic condition clustering in terms of ASP (in %) for the cases of using cosine similarity measure and Euclidean distance dissimilarity measure respectively. Eight clusters are generated. It is observed that the new i-vector approach achieves consistently better ASP scores in comparison with that of old i-vector approach. Understandably, after LDA transformation, much better ASP scores are achieved in comparison with the cases without using LDA, because the LDA-transformed i-vectors are more "speaker discriminative". When LDA is used, the lower the i-vector dimensions, the worse the ASP scores. According to the above results, we conjecture that the new i-vector approach may perform better than the old i-vector approach for speaker recognition applications.

4.3. Comparison of i-Vector Approaches for IVN-based Training

We also compared two i-vector based approaches to acoustic sniffing for IVN-based ML training of acoustic models when the cosine similarity measure and Euclidean distance dissimilarity measure are used respectively. The results (WER in %) are summarized in Table 2. In this set of experiments, again 8 acoustic conditions (therefore 8 IVN feature transforms) were used. For the case of using cosine similarity measure and no LDA, after 40 main cycles of IVN-based ML training [13], the new i-vector based acoustic sniffing method achieves a WER of 27.1%, which is slightly better than the WER of 27.3% using the old i-vector approach. After LDA transform, the WER of the new i-vector approach reduces to 26.7% and 26.5% for the dimensions of 600 and 400 respectively. Further reduction of the i-vector dimension to 200 incurs significant WER increase for the old approach. The new i-vector approach works well in a wider range of i-vector dimension. Similar observations can be made for the cases of using Euclidean distance dissimilarity measure. All the IVN-trained systems perform significantly better than the baseline system.

5. CONCLUSION AND DISCUSSION

In this paper, we have proposed a new approach to extracting a low-dimensional i-vector from a speech segment to represent acoustic information irrelevant to phonetic classification. Compared with the traditional i-vector approach, a full factor analysis model with a residual term is used. New procedures for hyperparameter estimation and i-vector extraction are derived and presented. The experimental results on Switchboard-1 corpus demonstrated that the proposed i-vector approach performs better than the old approach for improving speaker clustering result as measured by a so-called Average Speaker Purity (ASP) criterion, and for improving recognition accuracy in an IVN-based framework for speech recognition.

Ongoing and future works on this topic include:

- to verify the effectiveness of the IVN-based framework for even larger scale ASR tasks;
- to investigate better discriminative feature extraction methods (e.g., [10, 14]) when the cosine measure is used to compare the similarity of two i-vectors;
- to study the effectiveness of the new i-vector approach for speaker recognition applications.

We will report those results elsewhere once they become available.

6. REFERENCES

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

[2] O. Glembek, L. Burget, P. Matejka, M. Karafiat, and P. Kenny, "Simplification and optimization of i-vector extraction," *Proc. ICASSP-2011*, pp.4516-4519.

[3] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," *Proc. ICASSP-1992*, pp.517-520. See also LDC website: http://www.ldc.upenn.edu for more details.

[4] Q. Huo and D. Zhu, "A maximum likelihood training approach to irrelevant variability compensation based on piecewise linear transformations," *Proc. Interspeech-2006*, pp.1129-1132.

[5] Q. Huo and D. Zhu, "Robust speech recognition based on structured modeling, irrelevant variability normalization and unsupervised online adaptation," *Proc. ICASSP-2009*, pp.4637-4640.

[6] P. Kenny, "Joint factor analysis of speaker and session variability: theory and algorithms," Technical Report CRIM-06/08-13, CRIM, Montreal, 2006.

[7] P. Kenny, G. Boulianne, P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Trans. on Speech and Audio Processing*, Vol. 13, No. 3, pp.345-354, 2005.

[8] I. Lapidot, "SOM as likelihood estimator for speaker clustering," *Proc. Eurospeech-2003*, pp.3001-3004.

[9] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communication*, Vol. COM-28, pp.84-95, 1980.

[10] Y. Ma, S. Lao, E. Takikawa, and M. Kawade, "Discriminant analysis in correlation similarity measure space," *Proc. ICML-2007*, pp.577-584.

[11] P. Matejka, O. Glembek, F. Castaldo, M. J. Alam, O. Plchot, P. Kenny, L. Burget, and J. Cernocky, "Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification," *Proc. ICASSP-2011*, pp.4828-4831.

[12] NIST Scoring Toolkit SCTK, see the following site for details:
http://itl.nist.gov/iad/mig/tests/rt/2002/software.htm.

[13] G.-C. Shi, Y. Shi, and Q. Huo, "A study of irrelevant variability normalization based training and unsupervised online adaptation for LVCSR," *Proc. Interspeech-2010*, pp.1357-1360.

[14] H. Tang, S. M. Chu, T. S. Huang, "Spherical discriminant analysis in semi-supervised speaker clustering," *Proc. NAACL HLT-2009: Short Papers*, pp.57-60.

[15] J. Xu, Y. Zhang, Z.-J. Yan, and Q. Huo, "An i-vector based approach to acoustic sniffing for irrelevant variability normalization based acoustic model training and speech recognition," *Proc. Interspeech-2011*.

[16] S. Young, *et al.*, The HTK Book (for HTK version 3.4), 2006.

[17] Y. Zhang, J. Xu, Z.-J. Yan, and Q. Huo, "A study of irrelevant variability normalization based discriminative training approach for LVCSR," *Proc. ICASSP-2011*, pp.5308-5311.

[18] Y. Zhang, J. Xu, Z.-J. Yan, and Q. Huo, "An i-vector based approach to training data clustering for improved speech recognition," *Proc. Interspeech-2011*.