An i-Vector based Approach to Acoustic Sniffing for Irrelevant Variability Normalization based Acoustic Model Training and Speech Recognition

Jian Xu[1,2*], *Yu Zhang*[1,3*], *Zhi-Jie Yan*[1], *Qiang Huo*[1]

[1]Microsoft Research Asia, Beijing, China
[2]Department of Automation, University of Science and Technology of China, Hefei, China
[3]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
{v-jiaxu,v-yzzhan,zhijiey,qianghuo}@microsoft.com

Abstract

This paper presents a new approach to acoustic sniffing for irrelevant variability normalization (IVN) based acoustic model training and speech recognition. Given a training corpus, a so-called i-vector is extracted from each training speech segment. A clustering algorithm is used to cluster the training i-vectors into multiple clusters, each corresponding to an acoustic condition. The acoustic sniffing can then be implemented as finding the most similar cluster by comparing the i-vector extracted from a speech segment with the centroid of each cluster. Experimental results on Switchboard-1 conversational telephone speech transcription task suggest that the i-vector based acoustic sniffing outperforms our previous Gaussian mixture model (GMM) based approach. The proposed approach is very efficient therefore can deal with very large scale training corpus on current mainstream computing platforms, yet has very low run-time cost.

Index Terms: i-vector, acoustic modeling, irrelevant variability normalization, unsupervised online adaption, LVCSR

1. Introduction

In a state-of-the-art Large Vocabulary Continuous Speech Recognition (LVCSR) system, robust acoustic model is usually trained 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 Irrelevant Variability Normalization (IVN) based approach can be used (e.g., [7, 9]). Fig. 1 illustrates how it works for acoustic modeling, training and adaptation. In the off-line training stage (upper part), a set of feature transforms along with the generic Hidden Markov Models (HMMs) are trained using a Maximum Likelihood (ML) or Discriminative Training (DT) 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.

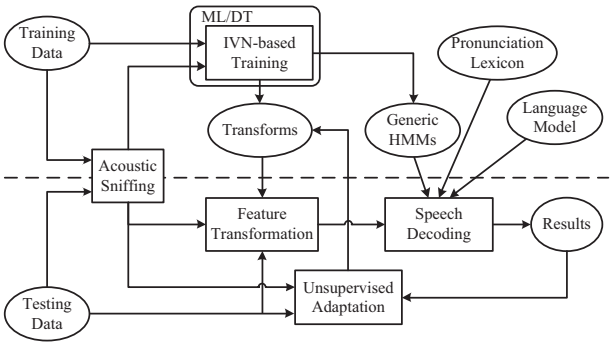

Figure 1: *An illustration of IVN-based framework for acoustic modeling, training and adaptation.*

Apparently, in IVN-based framework, the "acoustic sniffing" module is essential for both training and recognition. This module should be able to detect different acoustic conditions effectively and efficiently so that different feature transforms can be learned from the training data of each condition, and the most appropriate transforms can be chosen in recognition. Previously, we have studied two acoustic sniffing methods, namely a moving-window based frame labeling method in [7], and a Gaussian mixture model (GMM) based data-clustering and selection method in [9]. Better results are achieved by using the second approach on Switchboard-1 conversational telephone speech transcription task. However, there are two major drawbacks of the GMM-based approach: 1) The GMM-based likelihood score is not independent of the phonetic content of the speech segment concerned. This could be even more serious for a short utterance (e.g., in voice search scenario); 2) when the number of acoustic conditions (therefore the number of feature transforms) increases, the GMM-based method does not scale up well in both training and recognition due to its expensive computational cost for likelihood evaluation. Therefore, an improved "acoustic sniffing" module for IVN-based framework is desirable.

Inspired by the recent success of a so-called i-vector based approach for speaker recognition [1], we found that i-vector methodology can be easily modified to come out new efficient approaches for both training data clustering and acoustic sniffing. In a companion paper [10], we present an i-vector based

*This work was done when Jian Xu and Yu Zhang were interns in Speech Group, Microsoft Research Asia, Beijing, China.

approach to clustering training data for training multiple acoustic models to improve speech recognition accuracy. In this paper, we present an i-vector based approach to acoustic sniffing for IVN-based framework.

The rest of the paper is organized as follows. In Section 2, we present the i-vector based acoustic sniffing approach. In Section 3, we report experimental results. Finally, we conclude the paper in Section 4.

2. i-Vector based Approach to Acoustic Sniffing

Although we borrowed the main idea of i-vector extraction from [1], our hyperparameter estimation procedure is different from the one in [3], which was used in [1]. In [10], we have explained in detail the theoretical justification of our version of i-vector approach. In the following, we describe our i-vector based approach to acoustic sniffing.

2.1. Extracting i-Vectors from Training Data

Let $\mathcal{Y} = \{\boldsymbol{Y}_i | i = 1, 2, \ldots, I\}$ denote the training data set, where $\boldsymbol{Y}_i = (\boldsymbol{y}_1^{(i)}, \boldsymbol{y}_2^{(i)}, \ldots, \boldsymbol{y}_{T_i}^{(i)})$ is a sequence of T_i D-dimensional feature vectors extracted from the i-th speech segment. From $\mathcal{Y}$, a Gaussian mixture model (GMM) can be trained using a maximum likelihood (ML) 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 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 GMM-UBM parameters.

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 $\boldsymbol{M}_0$ as follows:

$$\boldsymbol{M}(i) = \boldsymbol{M}_0 + \boldsymbol{T}\boldsymbol{w}(i) \tag{2}$$

where $\boldsymbol{T}$ is a fixed but unknown $(D \cdot K) \times F$ rectangular matrix of low rank (i.e., $F \ll (D \cdot K)$), and $\boldsymbol{w}(i)$ is an F-dimensional random vector having a prior distribution of standard normal distribution $\mathcal{N}(\cdot; \boldsymbol{0}, \boldsymbol{I})$. In [1], $\boldsymbol{T}$ is called the total variability matrix.

Given Ω and $\boldsymbol{T}$, an i-vector can be extracted from $\boldsymbol{Y}_i$ as follows:

$$\hat{\boldsymbol{w}}(i) = \boldsymbol{l}^{-1}(i) \boldsymbol{T}^\top {\boldsymbol{R}_0}^{-1} \boldsymbol{\Gamma}_{\boldsymbol{y}}(i) \tag{3}$$

where

$$\boldsymbol{l}(i) = \boldsymbol{I} + \boldsymbol{T}^\top \boldsymbol{\Gamma}(i) \boldsymbol{R}_0^{-1} \boldsymbol{T}; \tag{4}$$

$\boldsymbol{\Gamma}(i)$ 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:

$$\gamma_k(i) = \sum_{t=1}^{T_i} P(k | \boldsymbol{y}_t^{(i)}, \Omega) \tag{5}$$

$$\Gamma_{\boldsymbol{y},k}(i) = \sum_{t=1}^{T_i} P(k | \boldsymbol{y}_t^{(i)}, \Omega)(\boldsymbol{y}_t^{(i)} - \boldsymbol{m}_k) \tag{6}$$

where

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

To facilitate i-vector clustering, we normalize each i-vector to have a unit norm.

Given the training data $\mathcal{Y}$ and the pre-trained GMM-UBM Ω, the hyperparameters (i.e., total variability matrix) $\boldsymbol{T}$ can be estimated by using the following procedure:

Step 1: *Initialization*

Set the initial value of each element in $\boldsymbol{T}$ randomly from $[Th_1, Th_2]$, where Th_1 and Th_2 are two control parameters ($Th_1 = 0, Th_2 = 0.1$ in our experiments). For each training speech segment, calculate the corresponding "Baum-Welch" statistics as in Eq. (5) and Eq. (6).

Step 2: *E-step*

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

$$\begin{aligned} E[\boldsymbol{w}(i)] &= \boldsymbol{l}^{-1}(i) \boldsymbol{T}^\top {\boldsymbol{R}_0}^{-1} \boldsymbol{\Gamma}_{\boldsymbol{y}}(i) \\ E[\boldsymbol{w}(i)\boldsymbol{w}^\top(i)] &= E[\boldsymbol{w}(i)] E[\boldsymbol{w}^\top(i)] + \boldsymbol{l}^{-1}(i) \end{aligned} \tag{7}$$

where $\boldsymbol{l}(i)$ is defined in Eq. (4).

Step 3: *M-step*

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

$$\sum_{i=1}^{I} \boldsymbol{\Gamma}(i) \boldsymbol{T} E[\boldsymbol{w}(i)\boldsymbol{w}^\top(i)] = \sum_{i=1}^{I} \boldsymbol{\Gamma}_{\boldsymbol{y}}(i) E[\boldsymbol{w}^\top(i)]. \tag{8}$$

Step 4: *Repeat or stop*

Repeat **Step 2** to **Step 3** for a fixed number of iterations or until the algorithm converges [10].

2.2. Acoustic Condition Clustering using i-Vectors

After extracting the unit-norm i-vectors from all the training speech segments, the Linde-Buzo-Gray (LBG) clustering algorithm [5] can be used to cluster them into several clusters, each corresponding to a homogeneous acoustic condition. Following cosine similarity is used to measure the similarity of two speech segments in the i-vector space:

$$sim(\hat{\boldsymbol{w}}(i), \hat{\boldsymbol{w}}(j)) = \hat{\boldsymbol{w}}(i)^\top \hat{\boldsymbol{w}}(j). \tag{9}$$

Given the above similarity measure, it can be proven that the centroid, $\boldsymbol{cw}$, 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:

$$\begin{aligned} \boldsymbol{cw} &= \operatorname*{argmax}_{\boldsymbol{cw}} \sum_{i=1}^{n} sim(\hat{\boldsymbol{w}}(i), \boldsymbol{cw}) \\ &= \begin{cases} \dfrac{\sum_{i=1}^{n} \hat{\boldsymbol{w}}(i)}{||\sum_{i=1}^{n} \hat{\boldsymbol{w}}(i)||} & \text{if } \sum_{i=1}^{n} \hat{\boldsymbol{w}}(i) \neq \boldsymbol{0} \\ \boldsymbol{0} & \text{otherwise} \end{cases}. \end{aligned} \tag{10}$$

After the LBG clustering converges, we obtain E clusters of i-vectors with their centroids denoted as $\boldsymbol{cw}_1, \boldsymbol{cw}_2, \ldots, \boldsymbol{cw}_E$, 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 and the total number of feature transforms equals the number of clusters.

2.3. i-Vector based Acoustic Sniffing

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 Eq. (5) and Eq. (6) using GMM-UBM.

Step 2: Extract an i-vector $\hat{\boldsymbol{w}}$ from $\boldsymbol{Y}$ using the calculated sufficient statistics and the pre-trained total variability matrix $\boldsymbol{T}$. Normalize the extracted i-vector to have a unit norm.

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

$$e = \underset{l=1,2,\ldots,E}{\operatorname{argmax}} \; sim(\hat{\boldsymbol{w}}, \boldsymbol{cw}_l) \tag{11}$$

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.

3. Experiments and Results

3.1. Experimental Setup

Switchboard-1 conversational telephone speech transcription task [2] 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 conversation sides are 4.84s, 547.16s, and 229.61s in the training set and 73.12s, 279.77s, and 184.47s in the testing set, respectively.

For front-end feature extraction, we used 39 PLP_E_D_A (in HTK's terminology [8]) 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 [9] and the results were evaluated by using the NIST Scoring Toolkit SCTK [6].

For i-vector based acoustic sniffing, both utterance based (denoted as "U" hereafter) and conversation-side based (denoted as "CS" hereafter) i-vectors were extracted in training and recognition stages. The settings of relevant control parameters were as follows: The number of GMM components $K = 1,024$, the dimension of i-vector $F = 400$, the number of iterations for updating $\boldsymbol{T}$ was 15. Furthermore, $\boldsymbol{T}$ was initialized by random values ranging from 0 to 0.1, where the thresholds are determined under the guidance of the dynamic range of the variance values in GMM-UBM. It is noted that too large initial values may lead to numerical problems in training $\boldsymbol{T}$.

Table 1: *Comparison of different approaches when using 8 IVN transforms (AS: acoustic sniffing approach; GMM: GMM based approach; IVEC: i-vector based approach; UA: unsupervised adaptation).*

Method		w/o UA		UA	
HMM	AS	WER(%)	Rel.(%)	WER(%)	Rel.(%)
ML	-	30.0	N/A	28.4	N/A
	GMM	27.6	8.0	25.0	12.0
	IVEC	27.1	9.7	24.7	13.0
DT	-	26.2	N/A	24.8	N/A
	GMM	24.9	5.0	22.9	7.7
	IVEC	24.1	8.0	22.3	10.1

Our ML- and DT-trained baseline systems achieved Word Error Rates (WERs) of 30.0% and 26.2% respectively. For all the IVN-based training experiments (ML and DT), we followed the settings used in [9].

3.2. i-Vector vs. GMM based Approach to Acoustic Sniffing

We compared i-vector based acoustic sniffing with our previous GMM-based approach [9]. The results are shown in Table 1. In this set of experiments, each conversation-side had been chosen as the speech segment in extracting i-vector and 8 acoustic conditions (therefore 8 IVN feature transforms) were used. After 40 main cycles of IVN-based ML training [7], the i-vector based acoustic sniffing method achieves a WER of 27.1%, which is slightly better than the previously reported WER of 27.6% using GMM-based acoustic sniffing. The performance gain is maintained after unsupervised ML adaptation of feature transforms. The IVN-based DT training (only for HMMs) and the corresponding unsupervised ML adaptation [9] have also shown similar performance gains. After adaptation, the DT-IVN method using i-vector based acoustic sniffing achieved a WER of 22.3% (10.1% relative WER reduction from the "DT baseline + UA"), while the GMM-based approach achieved a higher WER of 22.9% (7.7% relative WER reduction from the "DT baseline + UA"). This set of experiments demonstrated clearly the effectiveness of the i-vector based acoustic sniffing approach.

For comparison, starting from the ML- and DT-trained baseline systems, we performed conversation-side based unsupervised HMM adaptation using MLLR approach [4]. Eight regression classes are used and 3 EM iterations are performed to estimate the linear transforms. After two cycles of recognition and adaptation, the WERs are reduced to 28.4% and 24.8% respectively, which are worse than their IVN-based counterparts.

3.3. Effect of Using More IVN Transforms

Compared with the previous GMM-based acoustic sniffing approach, it is much easier for the i-vector based approach to scale up and handle more IVN transforms in run-time decoding. This is because in the i-vector based approach, the computation only involves extracting the i-vector for a given input speech segment, and calculating the cosine similarity with all the centroids of the clustered acoustic conditions. This is much more efficient than the required likelihood evaluations in the GMM-based approach.

Table 2: *Comparison of different approaches when using 128 IVN transforms.*

Method		w/o UA		UA	
HMM	AS	WER(%)	Rel.(%)	WER(%)	Rel.(%)
ML	-	30.0	N/A	28.4	N/A
	GMM	26.9	10.3	24.5	13.7
	IVEC	26.3	12.3	24.3	14.4
DT	-	26.2	N/A	24.8	N/A
	GMM	24.3	7.2	22.3	10.1
	IVEC	23.5	10.3	22.0	11.3

Table 3: *Comparison of choosing different types of speech segment for i-vector extraction (CS: conversation-side; U: utterance).*

Training	Recognition	WER (%)
CS	CS	27.1
CS	U	27.2
U	U	27.9

In this set of experiments, we increased the number of IVN feature transforms to 128, and compared the results of different approaches in Table 2. Again, the i-vector based acoustic sniffing achieved slightly better performance than the GMM-based approach. Compared with the results in Table 1, using more transforms is helpful.

3.4. Effect of Speech Segment Length

In this set of experiments, different granularities for i-vector extraction were compared. For Switchboard-1 task, utterance and conversation-side based speech segments are two natural choices: in training, it is quite flexible to choose either utterance or conversation-side as the speech segment unit for extracting i-vectors; in recognition, depending on the decoding scenarios, utterance based i-vector can be extracted for utterance-by-utterance recognition scenario, while conversation-side based i-vector is extracted for speech transcription scenario.

Table 3 shows the performance comparison of using different granularities of the speech segments in training and recognition for i-vector extraction. 8 IVN transforms were used and 40 main cycles of IVN-based ML training were performed. Best performance (27.1% WER) was obtained in the most favorable scenario, where CS-based i-vector extraction can be performed in both training and recognition. When we can only extract i-vectors utterance-by-utterance for testing sentences, the recognition performance varies with different training granularities: When the i-vectors of the training set were extracted on CS basis, the WER was 27.2%, which is close to the "CS-CS" scenario; However, if the training i-vectors were extracted on utterance basis, a slight performance degradation was observed (27.9% WER). These results suggest that using the metadata (conversation-side information in this case) in training may lead to more stable and reliable estimate of the i-vectors as well as the resultant acoustic condition clustering result. In recognition scenarios where utterance-by-utterance decoding is required, the i-vector approach can also give a reasonable recognition performance (e.g. the "CS-U" case).

4. Conclusion and Discussion

In this paper, we have proposed and investigated an i-vector based acoustic sniffing method in IVN-based framework. Compared with the previous GMM-based approach, the i-vector based approach is confirmed to perform better. At the same time, the proposed approach is very efficient therefore can deal with very large scale training corpus on current mainstream computing platforms, yet has very low run-time cost, therefore a large number of transforms can be used to exploit the full potential offered by IVN-based framework. Actually, to handle the large-scale training data, the i-vector extraction and model training tools 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.

Ongoing and future works on this topic include:

- to verify the effectiveness of the IVN-based framework for a large-scale voice search task with 7,500 hours of speech training data;
- to improve the i-vector based acoustic sniffing approach for different LVCSR application scenarios and deployment requirements.

We will report those results elsewhere once they become available.

5. References

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

[2] 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.

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

[4] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, Vol. 9, pp.171-185, 1995.

[5] 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.

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

[7] 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.

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

[9] 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.

[10] 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*.