Simplified Residual Factor Analysis for Text-Independent Speaker Verification

Lei Zhu¹, Rong Zheng¹, Bo Xu¹²

¹Digital Content Technology Research Center, Institute of Automation
²National Lab of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing 100190, China
E-mail: {lzhu, rzheng, xubo}@hitic.ia.ac.cn

ABSTRACT

Joint factor analysis (JFA) is a model used to treat the problem of speaker and session variability in GMM. In this paper, we add an additional direction to the joint factor analysis to model the homogeneous variability between the train utterance and the test utterance directly. This work is a natural extension of earlier work by P. Kenny [1] that shifted the model to test utterance along the channel subspace. We referred the new work as residual factor analysis (RFA). Then we proposed a new strategy called simplified residual factor analysis (S-RFA) to speedup the RFA procedure. We tested the proposed technique on the core condition of NIST 2008 speaker recognition evaluation (SRE08) and obtained about 4.4% relatively reduction in equal error rate (EER), compared to the JFA system with merely no extra CPU time.

Index Terms—Speaker Verification, Joint Factor Analysis, Residual Factor Analysis, Simplified Residual Factor Analysis

1. INTRODUCTION

In the recent NIST SRE, Joint Factor Analysis (JFA) has become the state-of-the-art technique to model speaker and session variability in Gaussian Mixture Models (GMM)[2].

In JFA system, the eigenvoice matrix is induced to model the homogeneous variability of all speakers, regardless the homogeneous variability just between the train utterance and the test utterance. So if the train utterance can’t be represented by the eigenvoice matrix well, the performance of JFA system will degrade. These considerations lead us to exploit a new way, in which we can take advantage of the homogeneous variability just between the train utterance and the test utterance. The new way is referred as Residual Factor Analysis (RFA), and the homogeneous variability is referred as residual vector.

Given a train utterance in RFA, we should calculate different residual models for different test utterance, which requires much more CPU time and seems to be an obvious short coming of the technology. To solve this problem, we propose an equivalent strategy, which is referred as the Simplified Residual Factor Analysis (S-RFA), to speed up the procedure of RFA with less degradation in performance.

Note that we expect the reader to be familiar with JFA. For an introduction to JFA, we refer the reader to [1, 3, 4, 5].

The remainder of the paper is organized as follows: The theory of RFA and other techniques are given in Sec 2. The detail of speech corpus is depicted in Sec 3 and then in Sec 4, we will present some experiments and results of the proposed algorithms using the SRE08 core test corpora (tel-tel). Final conclusion is given in Sec 5.

2. THEORETICAL BACKGROUND

2.1. Baum-Welch statistics

In order to be consistent with JFA algorithm, we define the same Baum-Welch statistics as in [1]. Given a speaker s and acoustic feature vectors, \( Y_1, Y_2, \ldots \), for each component \( c \), the centralized Baum-Welch statistics is:

\[
N_c = \sum_t \gamma_t(c)
\]

\[
F_c = \sum_t \gamma_t(c)(Y_t - m_c)
\]

Where \( \gamma_t(c) \) is the posterior probability of the event that the feature Vector \( Y_t \) is accounted for by the mixture component \( c \). We calculate these posteriors using the Universal Background Model.

2.2. Joint Factor Analysis

The basic assumption of joint factor analysis is that speaker- and channel-dependent GMM supervector are Gaussian distributed with most of the variance in these supervectors being
accounted for by a small number of hidden variable which is referred to as speaker and channel factors.

In JFA, the speaker- and channel-dependent supervector \( M \) can be decomposed into a sum of two supervectors (3), a speaker supervector \( s \) (4) and a channel supervector \( c \) (5).

\[
M = s + c \quad (3)
\]
\[
s = m + vy + dz \quad (4)
\]
\[
c = ux \quad (5)
\]

The diagonal matrix \( d \) is set as \( d^2 = \Sigma/\gamma \), which can speed-up the procedure dramatically with less degradation in system performance [1].

The factor \( x, y, z \) is calculated jointly using the equation (6)(7)(8), where \( w \) is defined as \( w = [ u \ v ] \), which is another form of equation (18) in [6].

\[
l = I + \sum_c \frac{\gamma N_c}{\gamma + N_c} w_c^* \Sigma^{-1} w_c \quad (6)
\]
\[
\begin{bmatrix} x \\ y \end{bmatrix} = l^{-1} \sum_c \frac{\gamma}{\gamma + N_c} w_c^* \Sigma^{-1} F_c \quad (7)
\]
\[
dz = (\gamma + N)^{-1}(F - Nux - Nvy) \quad (8)
\]

### 2.3. Residual Factor Analysis

The basic assumption of the residual factor analysis is that some of the homogeneity variability between the test utterance and enrol utterance cannot be represented by the eigenvoice matrix very well.

We induce the residual vector here to model those variability, which can’t be represented by the eigenvoice matrix.

**Def 1:** Given the eigenchannel matrix \( u \) and the eigenvoice matrix \( v \), for each test speech \( s_{te} \), the residual statistic is defined as

\[
F' = F - Nux - Nvy \quad (9)
\]

while the eigenchannel factor \( x_{te} \) and the eigenchannel factor \( y_{te} \) are calculated in the same way of the JFA system, and the residual vector is defined as

\[
r = N^{-1}F' \quad (10)
\]

**Def 2:** Given the eigenchannel matrix \( u \), the eigenvoice matrix \( v \) and the residual vector \( r \), for each train speech \( s_{tr} \), the residual model of \( s_{tr} \) is defined as

\[
s = m + vy + rz + dz \quad (11)
\]

where \( q \) is referred as the residual factor.

Fig. 1 illustrates the architecture of the residual factor analysis system. The main steps of the residual factor analysis are described as follows:

**Step 1.** Given a test utterance, we first calculate the eigenchannel factor \( x_{te} \) and the eigenvoice factor \( y_{te} \).

**Step 2.** Remove the channel supervector and speaker supervector from the centralized Baum-Welch statistics to generate the residual statistics \( F' \), and generate the residual vector \( r \) in such a way:

\[
r = N^{-1}F' = N^{-1}F_{te} = Nux_{te} - v_{yt} \quad (12)
\]

**Step 3.** For each train utterance, concatenate the eigenchannel matrix \( u \), the eigenvoice matrix \( v \) and the residual vector \( r \) to calculate factors \( x_{tr} \), \( y_{tr} \) and \( q_{tr} \). The equations here are similar to the equations (6)(7), where \( w \) is defined as \( w = [ u \ v \ r ] \).

We calculate \( z_{tr} \) from the following equation:

\[
dz = (\gamma + N)^{-1}(F - Nux_{tr} - Nvy_{tr} - Nq_{tr}) \quad (13)
\]

**Step 4.** We build up the residual model using the same method as that in JFA system:

\[
s_{tr} = m_{tr} + vy_{te} + rz_{tr} + dz_{tr} \quad (14)
\]

then update the residual model with the channel supervector \( c \) of the test utterance:

\[
m_{tr} = s_{tr} + u_{xe} \quad (15)
\]

**Step 5.** The true log-likelihood ratio (LLR) is calculated using the equation (16), which is an extension of the linear scoring method detailed in [7] and the tz-norm method is applied finally.

\[
LLK = \sum_c w_{ubmc,c}(v_{c,y_{te} + r_{te,c,q_{te} + d_{c,z_{te}}}) + \Sigma^{-1}(v_{c,y_{te} + d_{c,z_{te}}}) \quad (16)
\]

### 2.4. Simplified Residual Factor Analysis

Considering that the residual vector \( r \) is a test-dependent vector, the training procedure of the residual model for different test utterance will require enormous CPU time.
We induce the simplified residual factor analysis (S-RFA) here to release the heavy burden of CPU time. In RFA, Most of the CPU time is cost to invert the matrix \( l(17) \) in the equation (18), where \( w \) is defined as \([ u \ v \]\) and \( X \) is defined as \([ x \ y ]^T\).

\[
I + \sum_c \frac{\gamma N_c}{\gamma + N_c} w_c^c \Sigma_c^{-1} w_c - I + \sum_c \frac{\gamma N_c}{\gamma + N_c} r_c^* \Sigma_c^{-1} r_c = \lambda \Sigma \gamma
\]

To make the algorithm S-RFA more clearly, we rewritten \( l \) as \( \begin{pmatrix} \varphi^T \eta^T \delta^T \end{pmatrix} \). Considering the fact that most of the homogeneous variability can be represented by the eigenvoice matrix, we can assume (19) and rewrite the equation (18) as (20).

\[
\begin{cases}
\varphi X = \eta q \\
\varphi X = \sum_c \frac{\gamma}{\gamma + N_c} w_c^c \Sigma_c^{-1} F_c \\
\eta^* X + \delta q = \sum_c \frac{\gamma}{\gamma + N_c} r_c^* \Sigma_c^{-1} F_c
\end{cases}
\]

So we can calculate \( X \) from (21) firstly, which is another form of equation (7), and then calculate \( q \) from the equation (22).

\[
X = \varphi^{-1} \sum_c \frac{\gamma}{\gamma + N_c} w_c^c \Sigma_c^{-1} F_c
\]

\[
q = \delta^{-1} \sum_c \frac{\gamma}{\gamma + N_c} r_c^* \Sigma_c^{-1} (F_c - N_c w_c X)
\]

Because the calculation of the factor \( x \) and \( y \) in equation (21) has no relation with the residual vector \( r \), we can move this step to the train phase, which will release the heavy CPU burden of RFA in the verification phase.

In the train phase, we calculate the eigenchannel factor \( x_{te} \) and the eigenvoice factor \( y_{te} \) using the same way in JFA system.

In the verification phase, the main steps are described as follows:

**Step 1. and Step 2.** Calculate the eigenchannel factor \( x_{te} \) and the eigenvoice factor \( y_{te} \) to generate the residual factor \( r \), which are same as **Step 1. and Step 2.** in Sec 2.3.

**Step 3.** Calculate the residual factor \( q_{te} \) using the equation (22) and calculate \( z_{te} \) using the equation (13). Obviously, the CPU time required here is much less than that in Sec 2.3.

**Step 4. and Step 5.** Build up the residual model and calculate the score, which are same as **Step 4. and Step 5.** in Sec 2.3.

### 3. EXPERIMENTAL SETUP

#### 3.1. Test Set

The results of our experiments are reported on the core condition of NIST SRE08 lcon4w-1con4w corpus including 1993 female targets and 1270 male targets. Each train and test conversation has an average duration of 5 minutes and there are no cross-gender trials.

#### 3.2. Feature Extraction

The speech is pre-emphasized with a factor of 0.97 and segmented into frames by a 25 ms Hamming window shifting with 10-ms frame rate. The first 12 Mel frequency cepstral coefficients together with log energy are calculated with their first and second derivatives to form a 39-dimensional feature vector.

MVA filtering is used to remove the linear channel effects. A simple energy based VAD is used to prune out silence frames. Finally the Gaussianization is applied to all the MFCCs[9].

#### 3.3. Details of Joint Factor Analysis

We use gender-dependent UBM with 1024 Gaussian in our experiments. The UBM was trained with 400 records from SRE04 and SRE05 lcon4w-1con4w data for each gender.

The eigenchannel matrix are also trained gender-dependently from 184 female speakers with 2592 utterance and 128 male speakers with 1876 utterance in the SRE04 and SRE05. The eigenchannel factor number is 100.

The gender-dependent eigenvoice matrix is trained using LDC releases of Switchboard II Phase 2, Switchboard Cellular Parts 2, SRE04 and SRE05, including 815 female speakers with 10134 utterances and 568 male speakers with 7391 utterances. The eigenvoice factor number is 300.

The diagonal matrix \( d \) is set as \( d^2 = \Sigma / \gamma \), and the relevance factor \( \lambda \) is set to be 16.

---

**Fig. 2. Schematic diagram of the simplified residual factor analysis**

Fig. 2 illustrate the architecture of the simplified residual factor analysis system.
3.4. Normalization

All scores were normalized by a gender-dependent zt-norm. We adopt the SRE04 lcon4w-1con4w corpus as t-norm corpus and the SRE05 lcon4w-1con4w corpus as z-norm corpus. The number of the corpus for z-norm and t-norm is 1000.

3.5. Hardware and software

We measure the real-time factor on the machine Dell 5310, with Intel Xeon 2.33G CPU, and 3.99G RAM. The SSE2 operation is used all through the C++ code.

4. RESULTS AND ANALYSIS

4.1. Performance

Table 1 shows the performance of RFA, S-RFA and JFA systems on the core condition of the NIST SRE08 (tel-tel, all speakers, all trials).

<table>
<thead>
<tr>
<th></th>
<th>female</th>
<th></th>
<th>male</th>
<th></th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EER</td>
<td>DCF</td>
<td>EER</td>
<td>DCF</td>
<td>EER</td>
</tr>
<tr>
<td>JFA</td>
<td>6.55</td>
<td>3.26</td>
<td>4.90</td>
<td>2.42</td>
<td>6.08</td>
</tr>
<tr>
<td>RFA</td>
<td>6.28</td>
<td>3.15</td>
<td>4.51</td>
<td>2.32</td>
<td>5.87</td>
</tr>
<tr>
<td>S-RFA</td>
<td>6.26</td>
<td>3.15</td>
<td>4.59</td>
<td>2.33</td>
<td>5.81</td>
</tr>
</tbody>
</table>

We can find that the performance of RFA and S-RFA system is better than that on the JFA system, which is a proof of our assumption that some of the homogeneous variability between the test utterance and the enrol utterance can’t be represented by the eigenvoice matrix very well.

4.2. Speed

Table 2 shows the speed of RFA, S-RFA and JFA system on the core condition of the NIST SRE08 (female speakers, all trials).

<table>
<thead>
<tr>
<th></th>
<th>Time(min.)</th>
<th>RTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>JFA</td>
<td>288</td>
<td>0.017</td>
</tr>
<tr>
<td>RFA</td>
<td>4859</td>
<td>0.282</td>
</tr>
<tr>
<td>S-RFA</td>
<td>326</td>
<td>0.019</td>
</tr>
</tbody>
</table>

The time includes extracting features, estimating the required factors and calculating the likelihood ratio. We can find that speed of RFA seems to be intolerable and the speed of S-RFA seems to be close to the speed of JFA.

5. CONCLUSIONS

In this paper, we firstly propose a new technique RFA to model the homogeneous variability between the train utterance and the test utterance. The experiments shown that the RFA leads to 4.4% reduction in EER with mass CPU time. Then we proposed an equivalent strategy S-RFA to speed up the procedure of RFA, which leads to an equivalent reduction in EER with merely no extra CPU time.

6. ACKNOWLEDGEMENTS

We would like to thank other members in our group particularly Xiaorui Wang, Ce Zhang, Shuang Xu for a series of useful advice and discussion.

7. REFERENCES