Modeling High-level Information by Using Gaussian Mixture Correlation for GMM-UBM Based Speaker Recognition

Jing Deng, Thomas Fang Zheng, Zhanjiang Song, Jian Liu

Center for Speech Technology, State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science & Technology, Tsinghua University, Beijing, 100084, P.R. China
1Beijing d-Ear Technologies Co., Ltd., http://www.d-Ear.com
dengj@cst.cs.tsinghua.edu.cn, fzheng@tsinghua.edu.cn, zjsong@d-Ear.com, liuj@cst.cs.tsinghua.edu.cn

Abstract

The Gaussian mixture model-universal background model (GMM-UBM) has been dominant in text-independent speaker recognition tasks. However the conventional GMM-UBM method assumes that each Gaussian mixture is independent and ignores the fact that within Gaussian mixtures, there do exist some useful high-level speaker-dependent characteristics, such as word usage or speaking habits. Based on the GMM-UBM method, a method is proposed to use Gaussian mixture correlation to model the high-level information for speaker recognition tasks. In this method, we first cluster the Gaussian mixtures of the UBM into a small number of classes in terms of the mean vectors; in the following step, a universal class transition probability matrix (UCTPM) is learned which is helpful in modeling the high-level speaker’s characteristics embedded in Gaussian mixture correlation. During the training phase, a speaker-dependent class transition probability matrix is adapted from the UCTPM. Experiments over two different databases show that an average 20.38% error rate reduction (ERR) can be achieved compared with the conventional GMM-UBM method.

1. Introduction

In recent years, the Gaussian mixture model (GMM) has become one of the dominant approaches to speaker recognition. In general, these systems use GMMs to perform likelihood functions. A universal background model (UBM) is adopted to model out-of-set speakers and Bayesian adaptation is used to obtain speaker models from the UBM using the training data. This method is referred to as the GMM-UBM [1, 2] method for short.

However, GMM-UBM based speaker recognition systems typically use only short-term, low-level acoustic information, such as cepstral features. While these systems produce low error rates, they ignore higher-level information, such as the particular word usage or the idiolect (related to learned habits and style)[3]. Though high-level information holds the promise not only for improvement in basic recognition accuracy (by adding complementary knowledge), but also for robustness in acoustic degradations from channel and noise effects (to which low-level features are highly susceptible), it is difficult to extract and apply to text-independent speaker recognition tasks [4, 5, 6, 7]. HMM based speech recognition systems succeed in words recognition, which incorporate Gaussian mixtures correlation information. However multi-state HMMs are difficult to adapt to text-independent applications [8].

In this paper, we assume that the relation between Gaussian mixtures contains some high-level information, such as word usage or speaking habits. We propose a method to model Gaussian mixture correlation based on the conventional GMM-UBM method. Experiments over two different databases show that an average 20.38% error rate reduction can be achieved compared with the conventional GMM-UBM method.

The rest of this paper is organized as follows. In Section 2 we briefly review the commonly used GMM-UBM based method. In Section 3 we introduce our new approach based on the GMM-UBM method. Experiments and results are described in Section 4. Finally, conclusions are given in Section 5.

2. GMM-UBM based speaker recognition

For text-independent speaker recognition, the most successful model has been the Gaussian mixture model, used in conjunction with a universal background model for rejection purposes in open-set tasks.

In general, a UBM is trained using utterances from as many speakers as possible. To make the model more generalizable, the data used to train the UBM should not be included in, or overlapped with, that used to train individual speaker models. The UBM adopted in our experiments is also modeled by a GMM whose Gaussian mixture density can be described as

\[ p(x|ubm) = \sum_{m=1}^{M} g_m \cdot f\left(\mu_m^{(ubm)}, \Sigma_m\right) \]  \hspace{1cm} (1)

where \( x \) is a \( D \)-dimensional feature vector, \( \mu_m^{(ubm)}, \Sigma_m \), and \( g_m \) \( (m = 1, 2, ..., M) \) are the mean vector, the covariance matrix, and the weight for the \( m \)-th Gaussian mixture, and \( f(\cdot, \cdot) \) is the Gaussian density function. In this case, the UBM can be denoted by

\[ ubm = \{ g_m, \mu_m^{(ubm)}, \Sigma_m | m = 1, 2, ..., M \} \]  \hspace{1cm} (2)

A speaker model has the same form

\[ \lambda = \{ g_m, \mu_m^{(a)}, \Sigma_m | m = 1, 2, ..., M \} \]  \hspace{1cm} (3)

where the mean vectors are adapted to the training data of the given speaker from the UBM using the Maximum a Posteriori (MAP) adaptation method [9, 10], while the weights and covariance matrices remain unchanged, and the Gaussian mixture density is defined similar to Equation 1 as
Given a sequence of $T$ feature vectors $X = \{x_1, x_2, \ldots, x_T\}$ for a test speaker, the matching score with a certain model $\lambda$ (either a UBM or a speaker model) is taken as the normalized log-likelihood

$$S(X | \lambda) = \frac{1}{T} \sum_{t=1}^{T} \log p(x_t | \lambda)$$

while the normalized log-likelihood ratio will be

$$LLR(X | \lambda) = S(X | \lambda) - S(X | \text{ubm})$$

$$= \frac{1}{T} \sum_{t=1}^{T} \log \frac{p(x_t | \lambda)}{p(x_t | \text{ubm})}$$

Therefore the final recognized speaker is the one with the highest LLR score, or $\hat{\lambda}_{\text{max}} = \arg \max_{\lambda} LLR(X | \lambda)$. Given a predefined threshold $TSH_{\text{LLR}}$, the result should be accepted if $\hat{\lambda}_{\text{max}} \geq TSH_{\text{LLR}}$ or rejected otherwise.

The use of UBM makes it easier to define the threshold and hence the UBM is widely used in many open-set speaker recognition tasks.

3. Using Gaussian mixture correlation

In the conventional GMM-UBM framework, the assumption is that each Gaussian mixture is independent of every other and therefore the relationship among them is not considered. We assume that the Gaussian mixture correlation contains some speaker-dependent information and thus will be useful for speaker recognition. On the other hand, too detailed modeling of the correlation might make the recognizer more text-dependent. Based on this, we propose to model the correlation based on the Gaussian mixture classes instead of based on all the individual Gaussian mixtures. In other words, the $M$ Gaussian mixtures of the UBM will be clustered into a smaller number, $K$, of classes each of which is represented by one vector and the transition probability among these $K$ classes will be used to model the Gaussian mixture correlation.

![Figure 1: Clustering the Gaussians of the UBM](image)

The details of this method are described as follows. Given a UBM $\text{ubm} = \{g_m, \mu_{mubm}, \Sigma_m | m = 1, 2, \ldots, M\}$, the $K$-means [11] method is used to cluster the $M$ Gaussians into $K$ classes in terms of the mean vectors, say $\{C_{kubm}^{\text{ubm}}, w_k^{\text{ubm}} | k = 1, 2, \ldots, K\}$, where $w_k^{\text{ubm}}$ is the class weight defined as the percentage of the Gaussians that fall into the $k$-th class (see Figure 1). The mapping from a specific Gaussian mixture to its corresponding class is kept for future use.

Assume the training feature vector sequence for the UBM is $X = \{x_1, x_2, \ldots, x_T\}$. A universal class transition probability matrix (UCTPM), $\{a_{ij}^{\text{ubm}} | i, j \leq K\}$, which will be used to model the Gaussian mixture correlation, is learned as follows:

(i) For each feature vector $x_t$, calculate the $N$-best results in terms of the value of $g_n f(x, \mu_{nubm}, \Sigma_n)$, resulting in $N$ sequences of Gaussian indices, denoted by

$$\{IC_{nubm}^{\text{ubm}}(t) | t = 1, 2, \ldots, T\}, \ 1 \leq n \leq N$$

(ii) Look up the Gaussian-to-class mapping table obtained in the clustering procedure to obtain $N$ sequences of Gaussian class indices,

$$\{IC_{nubm}^{\text{ubm}}(t) | t = 1, 2, \ldots, T\}, \ 1 \leq n \leq N$$

(iii) Estimate the class transition probabilities as follows

$$a_{ij}^{\text{ubm}} = \frac{\text{Count}_{1 \leq n \leq N} (IC_{nubm}^{\text{ubm}}(t-1), IC_{nubm}^{\text{ubm}}(t))}{\text{Count}_{1 \leq n \leq N} (IC_{nubm}^{\text{ubm}}(t-1))}, \quad (1 \leq i, j \leq K)$$

A simple method to avoid zero class transition probabilities is to initialize each $\text{Count}_{\text{ubm}}^{\text{ubm}}(i, j)$ and $\text{Count}_{\text{ubm}}^{\text{ubm}}(i)$ to 1 before the counting procedure starts. A more complicated method can also be employed.

In the training phase, assume that the training feature vector sequence for a speaker model is $X = \{x_1, x_2, \ldots, x_T\}$. This speaker’s model is adapted from the UBM using the MAP adaptation and is denoted as $\lambda = \{g_n, \mu_{nubm}, \Sigma_m | m = 1, 2, \ldots, M\}$ with the Gaussian weight $g_n$ and the covariance matrix $\Sigma_m$ unchanged. A similar process is applied to get $\{IC_{nubm}^{\text{ubm}}(t) | t = 1, 2, \ldots, T\}, \ 1 \leq n \leq N$ and $\{IC_{nubm}^{\text{ubm}}(t) | t = 1, 2, \ldots, T\}, \ 1 \leq n \leq N$. The speaker-dependent class transition probability is estimated as

$$a_{ij}^{\text{ubm}} = \frac{\beta \cdot a_{ij}^{\text{ubm}} + a_{ij}^{\text{ubm}}}{\beta + 1}$$

where $\beta$ is an empirical value set to 0.8 for experiment group 1 and 0.5 for experiment group 2, and

$$a_{ij}^{\text{ubm}} = \frac{\text{Pr} (IC_{nubm}^{\text{ubm}}(t) = j | IC_{nubm}^{\text{ubm}}(t-1) = i)}{\text{Count}_{1 \leq n \leq N} (IC_{nubm}^{\text{ubm}}(t-1))}$$

$$= \frac{\text{Count}_{1 \leq n \leq N} (IC_{nubm}^{\text{ubm}}(t-1), IC_{nubm}^{\text{ubm}}(t))}{\text{Count}_{1 \leq n \leq N} (IC_{nubm}^{\text{ubm}}(t-1))}$$
Here, the speaker-dependent Gaussian class weight, $w^k_\lambda$, is assigned with the same value as that in the UBM, $w^\lambda_{ubm}$.

To calculate the matching score of the UBM or a speaker model $\lambda$ with a test feature sequence $X = \{x_1, x_2, \ldots, x_T\}$ in the recognition phase, the first step is to perform conventional GMM-UBM decoding. The decoding results include the best class sequences $\{IC^k_{ubm}(t)| t \le T\}$, $1 \le n \le N$ and $\{IC^i\}(t)| t \le T\}$, $1 \le n \le N$. According to Equation 5, we rescore the $n$-th best result of speaker model $\lambda$ using the following equation

$$S_n(X|\lambda) = \frac{1}{T} \sum_{t=1}^{T} \log\left(p(x_t|\lambda) \cdot w^k_\lambda \cdot a^k_{IC(t)} \cdot a^\lambda_{IC(t)}\right)$$  \hspace{1cm} (12)$$

where the speaker-dependent class transition probabilities are integrated. Similarly, we can rescore the $N$-best results for the UBM and we then will have $S_n(X|ubm)$ ($1 \le n \le N$) where the universal class transition probabilities are integrated. Afterwards, the new log-likelihood score is easily calculated as

$$LLR(X|\lambda) = \max_{n=1}^{N} (S_n(X|\lambda) - S_n(X|ubm))$$ \hspace{1cm} (13)$$

All the remaining procedures, including the acceptance or rejection decision, remain the same as in the conventional method.

4. Experiments

Our experiments were conducted in both the microphone and the telephone environments. The speech was sampled at 8 kHz and the frame length was 24ms with a 12ms frame shift. The value used for the pre-emphasis was 0.97 and hamming windowing was applied to each frame. After that, a 256-point FFT was calculated. The number of triangular Mel filters used in the MFCC calculation was 30. After the logarithm operation and the discrete cosine transformation (DCT), 16 MFCC coefficients were extracted. For each frame, the MFCC coefficients and their first derivative formed a 32-dimensional feature vector (i.e. $D=32$). The UBM or each speaker model was represented by $M=1,024$ Gaussian mixture density, where the value of $M$ was chosen empirically. Comparison experiments show that clustering the Gaussian mixtures of the UBM or any speaker model into $K=32$ classes can achieve the best result. During the training and testing, 4 was taken as the value of $N$ in the $N$-best rescoring procedure.

Table 1: Recognition Result (%) (Database I, training=24 seconds, testing=1 second)

<table>
<thead>
<tr>
<th># of top candidates</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>80.08</td>
<td>86.59</td>
<td>90.23</td>
<td>91.57</td>
<td>93.10</td>
</tr>
<tr>
<td>K=8</td>
<td>80.08</td>
<td>88.12</td>
<td>90.23</td>
<td>93.10</td>
<td>94.06</td>
</tr>
<tr>
<td>K=16</td>
<td>81.42</td>
<td>88.12</td>
<td>91.57</td>
<td>93.10</td>
<td>94.06</td>
</tr>
<tr>
<td>K=32</td>
<td>83.33</td>
<td>90.23</td>
<td>93.10</td>
<td>94.25</td>
<td>95.21</td>
</tr>
<tr>
<td>K=64</td>
<td>81.42</td>
<td>88.12</td>
<td>93.10</td>
<td>93.10</td>
<td>94.25</td>
</tr>
<tr>
<td>K=128</td>
<td>79.31</td>
<td>81.42</td>
<td>86.40</td>
<td>90.23</td>
<td>91.57</td>
</tr>
</tbody>
</table>

A conventional GMM-UBM based system was the baseline. Both the baseline and our proposed method shared the same front-end processing and used the same MAP adaptation method, and the Gaussian mixture density for the UBM or each speaker in our proposed method was the same as that in the baseline. The only difference was the log-likelihood rescoring which used the class transition probabilities to model the Gaussian mixture correlation.

Experiment Group 1: A 522-speaker speech database was used and is referred to as Database I. All the speech data is clean, and was recorded in an ordinary laboratory environment at 8 kHz sampling rate with 16-bit precision. For each speaker, a 24-second valid speech segment was taken for training while another 1-second valid speech segment was taken for testing. The performance of the baseline system is listed in Table 1. In order to choose an optimal $K$ value, several experiments were done and the results are also listed in Table 1 and shown in Figure 2. It can be seen that $K=32$ achieves the best result.

Results when taking 3-second valid speech for testing are given in Table 2.

Table 2: Recognition result (%) (Database I, training=24 seconds, testing=3 seconds)

<table>
<thead>
<tr>
<th># of top candidates</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>93.87</td>
<td>95.98</td>
<td>97.32</td>
<td>97.51</td>
<td>97.89</td>
</tr>
<tr>
<td>New method</td>
<td>94.83</td>
<td>96.36</td>
<td>97.32</td>
<td>97.51</td>
<td>97.89</td>
</tr>
<tr>
<td>Error Rate Reduction</td>
<td>15.66</td>
<td>9.45</td>
<td>14.18</td>
<td>22.89</td>
<td>18.48</td>
</tr>
</tbody>
</table>

Experiment Group 2: CallFriend Hub5 was used [12] and is referred to as Database II. In this database, the data format is 8 kHz 16-bit stereo; each channel contains speech collected from one end of the telephone. Every conversation in the database lasts around 30 minutes. The speech in different channels belongs to different speakers. So we can get two speakers’ speech from each conversation. The total number of speakers is 56. As above, for each speaker, 24-seconds valid speech was taken to train the speaker model. Results when
using 1-second speech and 3-second speech for testing are given in Tables 3 and 4 respectively.

Table 3: Recognition result (%) (Database II, training=24 seconds, testing=1 second)

<table>
<thead>
<tr>
<th># of top candidates</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>78.01</td>
<td>85.71</td>
<td>85.71</td>
<td>89.29</td>
<td>91.07</td>
</tr>
<tr>
<td>New method</td>
<td>81.57</td>
<td>89.29</td>
<td>91.07</td>
<td>91.07</td>
<td>92.85</td>
</tr>
<tr>
<td>Error Rate Reduction</td>
<td>16.19</td>
<td>25.05</td>
<td>37.51</td>
<td>16.62</td>
<td>19.93</td>
</tr>
</tbody>
</table>

Table 4: Recognition result (%) (Database II, training=24 seconds, testing=3 seconds)

<table>
<thead>
<tr>
<th># of top candidates</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>89.29</td>
<td>91.07</td>
<td>91.07</td>
<td>96.43</td>
<td>98.21</td>
</tr>
<tr>
<td>New method</td>
<td>92.86</td>
<td>94.64</td>
<td>96.64</td>
<td>98.21</td>
<td>100.00</td>
</tr>
<tr>
<td>Error Rate Reduction</td>
<td>33.33</td>
<td>39.98</td>
<td>62.37</td>
<td>49.86</td>
<td>100.00</td>
</tr>
</tbody>
</table>

We can see from the results that the proposed method is more effective than the conventional GMM-UBM based method in speaker recognition. The proposed method achieves a 15.96% error rate reduction in Database I and a 24.76% error rate reduction in Database II compared with the baseline. The average error rate reduction is 20.38%.

5. Conclusions

In this paper we propose a new approach to model high-level information within Gaussian mixture correlation by using a class transition probability matrix for text-independent speaker recognition. The proposed method modeling the Gaussian mixture correlation within this framework, the new method can help to remarkably improve the recognition results in practice. Our experiment proved that within Gaussian mixture correlation, there do exist some useful speaker-dependent information for speaker recognition.

Though our method could improve the performance of speaker recognition where the acoustic conditions for testing match those for training, it holds the promise for robustness to acoustic degradations from channel and noise effects and this needs to be verified in future experiments.

6. References