HIERARCHICAL GAUSSIAN MIXTURE MODEL FOR SPEAKER VERIFICATION

Ming Liu*,†, Eric Chang and Bei-qian Dai*

Microsoft Research Asia
*Department of Electronic Science and Technology,
University of Science and Technology of China
mliu197@hotmail.com, echang@microsoft.com

ABSTRACT
A novel type of Gaussian mixture model for text-independent speaker verification, Hierarchical Gaussian Mixture Model (HGMM) is proposed in this paper. HGMM aims at maximizing the efficiency of MAP training on the Universal Background Model (UBM). Based on the hierarchical structure, the parameters of one Gaussian component can also be adapted by the observation vectors of neighboring Gaussian components. HGMM can also be considered as a generalized GMM which replaces the Gaussian component in the GMM models with a local GMM. The hierarchical Gaussian mixture description of the local observation space is better than one single Gaussian distribution. Experiment on NIST 99 Evaluation corpus shows that the HGMM achieves an 18% relative reduction in EER compared with the conventional GMM.

1. INTRODUCTION
Security and authentication in speech-driven telephony applications can be achieved effectively through speaker verification (SV). Among the most successful approaches to robust text-independent speaker verification is the Gaussian Mixture Model (GMM) employed in many state-of-the-art systems [1, 2, 3]. For example, the system employing Bayesian adaptation of speaker models from a Universal Background Model (UBM) and handset-based score normalization (HNORM) has been the basis of the top performing systems in the past several NIST Speaker Recognition Evaluations [3].

In current UBM, there are a large number of Gaussian components (such as 2048 mixtures), and using a large number of mixtures requires more training and adaptation data. But in speaker adaptation scenario, the amount of adaptation data is always limited. So increasing the adaptation efficiency is a key problem for the performance of UBM+MAP[3] system. In speaker adaptation of speech recognition system, a different type of augmented MAP approach to tackle the speed problem is termed structural MAP (SMAP)[5]. The Gaussians in the system are organized in a tree structure. The MAP adaptation procedure starts at the root node, and then descends down to leaf nodes. Motivated by this method, we propose a novel Hierarchical Gaussian Mixture Model to embed the SMAP into its hierarchical structure. In our implementation of this new model, we simply build two hierarchies: the first hierarchy is root nodes that are conventional Gaussian components like the GMM, and the second hierarchy is leaf nodes of each root node. These leaf nodes are also Gaussian components, so the leaf nodes form a local GMM for each root node. By the local GMM, we can model the fine structure of the local area defined by the root Gaussian component. Thus in adaptation procedure, all leaf nodes of one root node can share their adaptation vectors through this hierarchical structure. In this way, adaptation efficiency is better than conventional MAP algorithm. The HGMM can also be considered as a general Gaussian Mixture Model. When the number of leaf nodes for each root nodes is set to zero, the HGMM becomes a GMM. In Section 2, we describe the UBM-MAP baseline system. Our model is discussed in Section 3. In Section 4, we present evaluation experiments that compare the performance of GMM with that of HGMM. It is shown that HGMM outperforms GMM by more than 18% relatively. The experimental results will be shown in Section 4. Finally, we provide our conclusion in Section 5.

2. THE BASELINE SYSTEM
For a UBM-MAP baseline system, two gender-dependent 1024-mixture UBMs are trained on a large number of speakers (6 hours of speech from all the 230 male and 309 female speakers of the 1999 NIST Speaker Recognition evaluation corpus) with Expectation-Maximization (EM) algorithm. For each target speaker, a 1024-mixture GMM is trained with MAP adaptation from the GD-UBM model. The score (or average log-likelihood ratio) of a given test segment \( O = \{ O_1, ..., O_T \} \) is computed as:

\[
\Lambda(O) = \frac{1}{T} \sum_{i=1}^{T} \left( \log p(O_t | \lambda_{\text{UAS}}) - \log p(O_t | \lambda_{\text{UBM}}) \right)
\]

where \( T \) is the length of the test segment, \( O_t \) is the feature vector at time \( t \) and \( \lambda_{\text{UAS}} \) and \( \lambda_{\text{UBM}} \) are parameters of the target model and UBM respectively.

3. HIERARCHICAL GAUSSIAN MIXTURE MODEL
The HGMM can be considered as a general GMM. In GMM, the data distribution is presented as a mixture of a number of Gaussian components as follows:

1This work was carried during the first author’s internship at Microsoft Research Asia.
\[ p(x_i | \lambda) = \sum_{i=1}^{M} w_i p_i(x_i) \]  

(2)

where \( p_i(x_i) \) is a Gaussian component, which is parameterized by a mean vector \( \mu_i \) and a covariance matrix \( \Sigma_i \). \( M \) is the number of mixtures. \( w_i \) is the mixture weights which satisfy the constraint \( \sum_{i=1}^{M} w_i = 1 \).

\[ p(x_i) = \sum_{i=1}^{M} w_i \frac{1}{(2\pi)^{D/2} | \Sigma |^{1/2}} \exp \left\{ -\frac{1}{2} (x_i - \mu_i)^\top \Sigma^{-1} (x_i - \mu_i) \right\} \]  

(3)

and in HGMM each \( p_i(x_i) \) is not a Gaussian density but a local Gaussian Mixture density. So, in HGMM formula (3) becomes as follows:

\[ p(x_i) = \sum_{j=1}^{K} w_{ij} \frac{1}{(2\pi)^{D/2} | \Sigma_j |^{1/2}} \exp \left\{ -\frac{1}{2} (x_i - \mu_j)^\top \Sigma_j^{-1} (x_i - \mu_j) \right\} \]  

(4)

\( K \) is the number of local mixtures. \( w_{ij} \) is the local mixture weights which satisfy the constraint \( \sum_{j=1}^{K} w_{ij} = 1 \).

All Gaussian components in local GMM belong to a root node that is also represented by a Gaussian component. During MAP training, the observation vectors of each local Gaussian component are pooled to estimate the new parameter of the root nodes, and then these new estimates also help to adjust the parameters of components that have no sufficient observations.

Figure 1 shows the overview of the structure of GMM. All the Gaussian p.d.f.s are combined with different weights to form a global distribution function. The local p.d.f. of HGMM, as shown in Figure 2, is not a Gaussian p.d.f, but a local GMM. From these local GMM p.d.f.s we combine them with different weights to form the global distribution function. In other words, with the local GMM density, the fine structure of local area can be described more precisely. Thus, the HGMM can describe the distribution with higher precision.

### 3.1 Training of the HGMM

When training the HGMM, we still use the EM algorithm to perform maximum likelihood training. The specifics of the training are as follows. Given the training vectors \( X = \{x_1, \ldots, x_T\} \), we first determine the probabilistic alignment of the training vectors into each component of HGMM. That is, for the \( i^{th} \) component of first hierarchy, we compute

\[ \gamma(i | x_i) = \frac{w_i p_i(x_i)}{\sum_{l=1}^{M} w_l p_l(x_i)} \]  

(5)

and for each Gaussian mixture of \( i^{th} \) component

\[ \gamma_i(j | x_i) = \frac{w_{ij} p_{ij}(x_i)}{\sum_{k=1}^{K} w_{ik} p_{ik}(x_i)} \]  

(6)
cascade these two formulas, we get
\[
\gamma(i, j | x_i) = \gamma(i | x_i) \gamma(j | x_i)
\] (7)
And then we can compute the statistics for the weight, mean, and variance parameters for each component.
\[
n^j = \sum_{i=1}^T \gamma(i, j | x_i)
\] (8)
\[
n^i = \sum_{i=1}^T \gamma(i | x_i)
\] (9)
\[
\gamma_j(x) = \frac{1}{n_j} \sum_{i=1}^T \gamma(i, j | x_i)x_i 
\] (10)
\[
\gamma_i(x^2) = \frac{1}{n_i} \sum_{i=1}^T \gamma(i | x_i)x_i^2
\] (11)
\[
\gamma_i(x^2) = \frac{1}{n_i} \sum_{i=1}^T \gamma(i | x_i)x_i^2
\] (12)
\[
\gamma_i(x^2) = \frac{1}{n_i} \sum_{i=1}^T \gamma(i | x_i)x_i^2
\] (13)

3.2. MAP Adaptation on the HGMM
The MAP adaptation on HGMM is different from that of GMM. Because of the hierarchical structure, we can adjust the parameters of local GMM at the same time, and compensate those Gaussian components without sufficient observations. The first step is just like the EM training of the HGMM: we compute the sufficient statistics for each component. And then re-estimate the parameters of HGMM as follows:
\[
\hat{\mu}_i = \alpha^w_i E_i(x) + (1 - \alpha^w_i)\mu_i
\] (14)
\[
\hat{\mu}_j = \alpha^w_j n_j / n_i + (1 - \alpha^w_j)\mu_j
\] (15)
\[
\hat{\mu}_i - \hat{\mu}_j = \alpha^w_j (E_j(x) - E_i(x)) + (1 - \alpha^w_j)(\mu_j - \mu_i)
\] (16)
\[
\alpha^i_j = \frac{n^j}{n^i + \rho^j}
\] (17)
\[
\alpha^i_j = \frac{n^j}{n^i + \rho^j}
\] (18)
where, \( \rho^j \) is the fixed relevance factor (we set \( \rho^j = 16 \)). In current implementation, we just adjust the mean. According to Reynolds[3], adapting only the mean gave the best performance.

3.3. Testing with the HGMM
The two hierarchical structures provide a different measurement from the conventional GMM models. We compute the occupation probability based on the first layer’s Gaussian component. After that, these occupation probability are used as the weight of each component in HGMM to form the likelihood of each vector as the formula
\[
\gamma(i | x_i) = \frac{w_i p_i(x_i)}{\sum_{i=1}^M w_i p_i(x_i)}
\] (20.a)
\[
p(x_i | \lambda) = \sum_{i=1}^M r(i | x_i) p_i(x_i)
\] (20.b)
In (20.a), the first hierarchy Gaussian component was used to calculate the sufficient statistics for each component, and then in (20.b), the local GMM was used to compute the refined likelihood for each vector. In this method, each vector’s sufficient statistic is normalized to 1, in other words, each testing vector’s contribution is equalized.

4. EXPERIMENTS
4.1. Evaluation Corpus
Speaker verification experiments are carried out on the 1999 NIST Speaker Recognition Evaluation corpus [7]. In NIST 99 corpus, the available telephone numbers per conversation was exploited to create matched and mismatched telephone handset (or number) test conditions. This corpus consists of 230 male target speakers and 309 female speakers. All speakers serve as both target and impostor speaker. The training data for each speaker consist of two minutes of speech taken from a single conversation. There are 3157 target trials and 34463 impostor trials.

4.2. Front-end Processing
First, the speech is pre-emphasized with a factor of 0.97 and segmented into frames by a 20-ms Hamming window progressing at a 10-ms frame rate. Then 12 MFCCs except for the 0th component and the first differentials are extracted from the speech[8]. Cepstral analysis is performed only on the passband (300-3400Hz) of the telephone speech. Finally, both RASTA[9] filtering and Cepstral Mean Subtraction (CMS) are used to remove linear channel convolutional effects on the cepstral features.


4.3. Evaluation Results on NIST99

As shown in Table 1, the performance of HGMM models is better than that of GMM models. The EER is 18.2% better than the EER of the GMM models. The DET curves of these two models are shown in Figure 3.

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Model Size</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>HGMM</td>
<td>256 x 4 mixture</td>
<td>12.00%</td>
</tr>
<tr>
<td>GMM</td>
<td>1024 mixture</td>
<td>14.67%</td>
</tr>
</tbody>
</table>

Table 1: The Speaker verification result of HGMM and GMM on NIST99.

![Figure 3: DET curves for HGMM models and GMM models. The EER is better for the HGMM models on NIST99.]

4.4. Evaluation Results on NIST02

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Model Size</th>
<th>SRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>HGMM</td>
<td>128 x 4 mixture</td>
<td>96.3%</td>
</tr>
<tr>
<td>GMM</td>
<td>512 mixture</td>
<td>95.2%</td>
</tr>
</tbody>
</table>

Table 2: The Speaker verification result of HGMM and GMM on NIST02.

![Figure 3: DET curves for HGMM models and GMM models. The EER is better for the HGMM models on NIST99.]

4.4. Evaluation Results on NIST02

We also performed experiment on the 2002 NIST Multimodal Speaker Recognition Development corpus [11]. In NIST 2002 MM corpus, there are speech waveforms from 10 speakers collected over three different channels. The speaker verification result is shown in Table 2. The speaker recognition result is shown in Table 3. In both experiments, the HGMM outperforms the GMM.

5. CONCLUSION AND FUTURE WORK

This paper introduces hierarchical Gaussian Mixture Model and compared the new model with GMM for a text-independent speaker verification task. The 1999 NIST Speaker Recognition Evaluation corpus is used to conduct the experiments. It is shown in the experiments that the performance of HGMM was better than that of GMM. This hierarchical structure can cluster a larger number of mixtures into fewer general clusters. In our two layers implementation, the cluster number equals the number of nodes in the first hierarchy. Then the adaptation process is conducted within each cluster, so the training efficiency is better than conventional GMM.

Although our experiment shows that the HGMM outperforms the GMM, there are more extension to be done, such as adding Handset Dependent UBM to normalize the scores under mismatched condition. In front-end processing, better silence removing method will further improve the performance.

6. ACKNOWLEDGEMENT

The authors thank Chao Huang, Chengyuan Ma, Tao Chen and other members of the speech group at Microsoft Research Asia for many fruitful discussions.

7. REFERENCES