Hybrid Model using Subspace Distribution Clustering Hidden Markov Models and Semi-Continuous Hidden Markov Models for Embedded Speech Recognizers

Youngkyu Cho, Sung-a Kim, and Dongsuk Yook

Department of Computer Science and Engineering
Korea University, Seoul, Korea
{ccameo,skim,yook}@voice.korea.ac.kr

Abstract

Today’s state-of-the-art speech recognition systems typically use continuous density hidden Markov models with mixture of Gaussian distributions. Such speech recognition systems have problems; they require too much memory to run, and are too slow for large vocabulary applications. Two approaches are proposed for the design of compact acoustic models, namely, subspace distribution clustering hidden Markov models and semi-continuous hidden Markov models. However, these models require also large memory to acquire high recognition accuracy. In this paper, we propose a new hybrid model using subspace distribution clustering hidden Markov model and semi-continuous hidden Markov model with the aim of achieving much more compact acoustic models.

1. Introduction

The hidden Markov models (HMMs) has been widely used for automatic speech recognizers [1]. Each state of HMMs is modeled as a mixture of elementary probability density functions. To obtain higher recognition accuracy, HMMs typically require huge amounts of Gaussian distributions. Those models have a major impediment to their deployment in mass applications because they require too large memory, and are too slow to run. A significant challenge is to design these recognizers so that they may be run on more affordable machines of lower processing power and smaller memory size without losing accuracy [2]. One commonly used approach, which reduces both the computational cost and the size of models, is the use of parameter tying; for example, semi-continuous HMM (SCHMM) [4,5] and subspace distributions clustering HMM (SDCHMM) [2,3]. Both methods similarly divide the feature space into streams, and tie subspace (or stream) distributions across all states of all HMMs. Also, they can be derived from already existing continuous density hidden Markov model (CDHMM).

SCHMMs resemble the M-mixture continuous HMMs with all the continuous output probability density functions shared among all Markov states. Compared with the continuous mixture HMMs, the SCHMMs can maintain the modeling ability of large mixture probability density functions. In addition, the number of free parameters and the computational complexity can be reduced, because all the probability density functions are tied together, providing a good compromise between detailed acoustic modeling and trainability [6].

SDCHMM can represent original full-space distributions as some combinations of a small number of subspace distribution prototypes. Combinational effects of SDCHMM can be very powerful. For instance, a 3-subspace SDCHMM system with 128 prototypes can represent $128^3 = 2097152$ different full space distributions. SDCHMMs are also computationally efficient because if a small number of subspace Gaussians are shared by a large number of full space Gaussian components, log likelihood of these subspace Gaussians can be precomputed only once at the beginning of every frame, and their values can be stored in lookup tables [2].

In this paper, we propose hybrid models using subspace distribution hidden Markov models with powerful combinational effect and semi-continuous hidden Markov models with low computational complexity with the aim of achieving much more efficient acoustic models.

This paper is organized as follows. Section 2 describes the theory of the SDCHMM and SCHMMs, analyzes of SDCHMMs and SCHMMs the memory requirements. Section 3 introduces a new hybrid acoustic model using SDCHMM and SCHMM. Finally, we draw our conclusion in section 4.

2. Review of SDCHMMs and SCHMMs

In this section, we review the theory of subspace distribution clustering hidden Markov modeling and semi-continuous hidden Markov modeling. And we compare the memory requirements of SDCHMMs with those of SCHMMs.

2.1. Theory of SDCHMMs

The theory of SDCHMM is derived from already existing CDHMM. The observation probability of the i\(^{th}\) state of a CDHMM is given by

$$P_i^{CDHMM}(O) = \sum_{m=1}^{M} c_{im}N(O; \mu_{im}, \sigma_{im}^2);$$  

(1)
where $P(O_i)$ is output probability of observation $O$ at state $i$, $c_m$ is the weight of the $m$th component of state $i$, $\mu_m$ and $\sigma^2_m$ are the mean and variance of the $m$th component of state $i$ [7]. To derive $K$-stream SDCHMMs from a set of CDHMMs, each Gaussian component is represented as a vector of $K$ indices that indicate the particular set of subspace Gaussian components, which are to be used to calculate the likelihood. Thus,

$$P^{\text{SDCHMM}}(O) = \sum_{m=1}^{K} c_m \left( \prod_{k=1}^{K} N_{\text{end}}(O_k; \mu_{mk}, \sigma^2_{mk}) \right),$$  

where $O_k$ are the $k$th subspace of observation $O$, and $\mu_{mk}$ and $\sigma^2_{mk}$ are mean and variance vectors of the $m$th mixture of $i$th state in the $k$th subspace, respectively. To derive $K$-stream SDCHMMs from a set of CDHMMs, the subspace Gaussians in each stream are clustered into a small set of $L$ codewords. Each original subspace Gaussian is then approximated by its nearest subspace Gaussian codeword. For each Gaussian component in the system, therefore, it is only necessary to store the set of indices.

2.2. Theory of SCHMM

SCHMM may appear similar to SDCHMM. SCHMM also divides the feature space into streams, and ties stream distributions across all states of all HMMs. The notations used in this paper are summarized as follows:

$P(O)$ state output probability given observation $O$;

$O_k$ $k$th stream of observation $O$;

$c_{mk}$ weight of the $m$th Gaussian of the $i$th state in the $k$th stream;

$\mu_{mk}$ mean vector of the $m$th Gaussian in the $k$th stream;

$\sigma^2_{mk}$ variance vector of the $m$th Gaussian in the $k$th stream;

$\gamma_k$ stream weight of the $k$th stream;

$N(\cdot)$ Gaussian pdf.

The observation probability of the $i$th state of a SCHMM is given by

$$P^{\text{SCHMM}}(O) = \prod_{k=1}^{K} \sum_{m=1}^{M} c_{mk} N_{\text{end}}(O_k; \mu_{mk}, \sigma^2_{mk})^\gamma_k.$$  

Each state output distribution of SCHMM is defined by $M$ mixture component weight $c_{mk}$ and the stream weight $\gamma_k$ from each of streams. SCHMMs estimate a mixture of Gaussian densities for each of the streams independently, and then combine them with stream weights, while SDCHMMs do not. Each Gaussians of a stream is associated with its own mixture component weight, whereas a mixture component weight is shared among all the $K$-subspace Gaussians of a SDCHMM state [2].

2.3. Memory requirements of SDCHMM and SCHMM

Memory requirements of SDCHMMs and SCHMMs are compared in Table 1. We assume that there are 2000 state pool, 20 Gaussians for each state, 512 codewords, 39 subspaces for SDCHMMs and 4 streams for SCHMMs.
An implication of the difference in the scope of the assumptions is the number of streams required: The SCHMM favors fewer streams of higher dimensions, so that correlation among more features can be modeled and there will be fewer mixture component weights. Conversely, SDCHMM favors more streams of lower dimensions so that quantization of the subspace Gaussians of CDHMMs will give smaller quantization errors [2]. As can be seen in Table 1, memory requirements of SDCHMMs and SCHMMs are reduced by 86% and by 34%, respectively.

Memory requirements for codewords, indices, weights, and stream weights are calculated as follows:

- **codewords** = \#codewords × 39(dimensions) × 2(mean, variance) × 4bytes
- **indices** = \#state pool × \#Gaussians × \#subspace × 1byte
- **weights** =
  - for SDCHMM = \#state pool × \#Gaussians × 4bytes
  - for SCHMM = \#state pool × \#codewords × \#streams × 4bytes
- **streamweights** = \#state pool × \#streams × 4bytes.

For each Gaussian component in the SDCHMM, it is necessary to store the index vectors. Each of stream Gaussians of SCHMMs is associated with its own mixture weight independently. Then, streams are combined with stream weights.

### 3. A Hybrid SDCHMMs and SCHMMs

In this section, we propose a new hybrid model using subspace distribution hidden Markov models with powerful combinational effect and semi-continuous hidden Markov models with low computational complexity.

#### 3.1. Theory of hybrid model using SDCHMM and SCHMM

The theory of hybrid model is also derived from that of the CDHMMs, using the following notations:

- \( P(O) \) state output probability given observation \( O \);
- \( O_{sk} \) \( s \)th subspace for the \( k \)th stream of observation \( O \);
- \( c_{imkc} \) weight in the \( i \)th state of the \( m \)th mixture in the \( k \)th stream;
- \( \mu_{skm} \) mean vector of the \( s \)th subspace of the \( m \)th mixture in the \( k \)th stream;
- \( \sigma_{skm}^2 \) variance vector of the \( s \)th subspace of the \( m \)th mixture in the \( k \)th stream;
- \( \gamma_k \) stream weight of the \( k \)th stream;
- \( N() \) Gaussian pdf.

The observation probability density of state \( i \) is given by

\[
\Pr^{\text{hybrid}}(O) = \prod_{k=1}^{K} \prod_{c_{imk}} \prod_{s=1}^{S} N(O_{sk} | \mu_{skm}, \sigma_{skm}^2)^{\gamma_k}.
\]  

(4)

Formally, let us denote the full vector space of dimension \( D \) by \( R^D \) with an orthogonal basis. \( R^D \) is decomposed into \( K \) orthogonal streams \( R^{d_k} \) of dimension \( d_k \). \( R^{d_k} \) is also decomposed into \( S \) orthogonal subspaces \( R^{d_{sk}} \) of dimension \( d_{sk} \). Each of the original full space Gaussians is projected onto each of the \( S \) subspace Gaussians of dimension \( d_{sk} \).

*Fig. 1.* A hybrid model with three streams and three subspaces per each stream

![Fig. 1](image-url)
The equation of (4) may be implemented in three steps:

Step 1) First, train a one-stream Gaussian mixture model with $L$ components onto the $K$ streams and the $S$ subspaces.
Step 2) Project all Gaussians in the original CDHMM onto those streams with subspaces.
Step 3) Tie the subspace Gaussians from all states and all phone models (CDHMMs) in each stream. This is done by clustering the subspace Gaussians into a small number of Gaussian prototypes in each subspace for each stream.

3.2. Comparison of memory requirements with other models

We compare our hybrid models with three other hidden Markov modeling methodologies: CDHMMs and SDCHMMs and SCHMMs. Memory requirements of hybrid model using SDCHMM and SCHMM are described in Table 2. We assume that there are 2000 state pool and 20 Gaussians for each state. For tied models, we assume that there are codebooks with 256 codewords and 4 streams with 3 subspaces per each stream.

Memory requirement for codewords, indices, weights, and stream weights are calculated as follows:

\[
\begin{align*}
codewords &= \text{#codewords} \times 39(\text{dimensions}) \times 2(\text{mean, variance}) \times 4\text{bytes} \\
indices &= \text{#state pool} \times \text{#Gaussians} \times \text{#subspace} \\
weights &= \text{#state pool} \times \text{#Gaussians} \times \text{#streams} \\
\text{streamweights} &= \text{#state pool} \times \text{#streams} \times 4\text{bytes}
\end{align*}
\]

<table>
<thead>
<tr>
<th>Condition &amp; memory requirement</th>
<th>A hybrid model</th>
</tr>
</thead>
<tbody>
<tr>
<td>#States pool</td>
<td>2000</td>
</tr>
<tr>
<td>#Gaussians (39-dim)</td>
<td>20</td>
</tr>
<tr>
<td>#codewords</td>
<td>256</td>
</tr>
<tr>
<td>#subspaces (streams)</td>
<td>4 (3)</td>
</tr>
<tr>
<td>Memory for codewords</td>
<td>79,872 byte</td>
</tr>
<tr>
<td>Memory for indices</td>
<td>480,000 byte</td>
</tr>
<tr>
<td>Memory for Gaussians</td>
<td>640,000 byte</td>
</tr>
<tr>
<td>Memory for stream weights</td>
<td>160,000 byte</td>
</tr>
<tr>
<td>Total memory requirement</td>
<td>1,359,872 byte</td>
</tr>
</tbody>
</table>

Table 2. Condition and memory requirement of Hybrid model.

As can be seen in Table 2, Memory requirement of hybrid model using SDCHMM and SCHMM are about 89% less than that of CDHMMM. Compared to SDCHMM and SCHMM, our hybrid models yield a more compact model set. Hybrid models using SDCHMM and SCHMM reduced memory requirements over 8%, compared with SDCHMM and SCHMM.

4. Conclusion

Compared with the continuous mixture HMMs, the SCHMMs and SDCHMMs can maintain the modeling ability of large mixture probability density functions. In addition, the computational complexity can be reduced, because all the probability density functions are tied together. Also SDCHMMs can simulate all original full-space distribution by some combinations of a small number of subspace distribution prototypes. SDCHMMs are also computationally efficient because log likelihoods of these subspace Gaussians can be precomputed only once at the beginning of every frame.

Therefore, our hybrid model using subspace distribution hidden Markov models and semi-continuous hidden Markov models may make acoustic models more compact, while they have powerful representational power.

5. References