Accelerating Segment Model Decoding for LVCSR by Parallel Processing of Neighboring Segments

Shouye Peng, Wen-Ju Liu, and Hua Zhang

National Laboratory of Pattern Recognition, Institute of Automation, China Academy of Sciences, Beijing 100190
{sypeng, lwj, hzhang}@nlpr.ia.ac.cn

Abstract. In human speech, most boundaries between phones/words are fuzzy. If a time slice which only includes a single boundary is given, it is possible that the boundary may locate at any frame within the slice. Different boundary locations form several potential observation segments, which should have similar acoustic spaces because of their neighboring trait in time domain. We call them neighboring segments. In this paper, a fast algorithm of parallel processing of neighboring segments is proposed for decoding. Since the decoder can search a bigger pruning threshold in parallel processing, the proposed algorithm is faster than decoding a single segment. This algorithm is successfully integrated into a Segment Model (SM) based Mandarin Large Vocabulary Continuous Speech Recognition (LVCSR) system, and saves approximately 50% decoding time without obvious influence on the recognition accuracy.

Keywords: Speech Recognition, LVCSR, Segment Model.

1 Introduction

Segment Model (SM) is a family of methods that adopt segmental distribution rather than frame-based features to represent the underlying trajectory of the observation sequence. Because SMs can resolve some limitations of Hidden Markov Model (HMM) partially [1], they are good alternatives to HMM in speech recognition systems [2]. Experiments also proved that SMs have performed better than HMM [1,3,4]. However, it is hard for SMs to be employed into practice due to their high complexities.

Stochastic Segment Model (SSM) belongs to SM family, and is mainly discussed in this paper. SSM is a sequence of regions simulated by Gauss Mixture Models (GMM), and the score of SSM is the sum of region probabilities. In LVCSR system, many segment models are usually involved into the likelihood estimation of the observation sequence, but only a few of them with high scores would be reserved at last. So the region probability calculations of those eliminated models are unnecessary during decoding. V. Digalakis et al [3] tried to prune the unnecessary calculations by estimating model scores from part of segments: The calculations of region probabilities were divided into multi-stages. In each stage, the score of SSM was evaluated by the probability sum of part regions, and then these SSMs with low scores were pruned. The decoder need not continue to calculate the region probabilities of pruned models any more. This pruning could save much decoding time, and is called multistage pruning algorithm in paper [4].
Due to soft vocal organs, human pronunciation changes slowly. That indicates the neighboring speech segments should have similar boundaries and acoustic spaces, and should also obey similar pruning strategies when decoding. In this paper, a parallel decoding algorithm is proposed for processing neighboring segments. The algorithm combines multistage pruning into parallel decoding, which provides a wider time domain for searching a bigger pruning threshold. Experiment shows this parallel algorithm performs better for pruning than only decoding a single segment.

The paper is organized as follows. The detail of SSM decoding is addressed in the next section. In section 3, the multistage pruning method is discussed, and then we would represent the fast algorithm this paper proposes. Next, in section 4, the experiment and analysis are described. Finally, conclusions are given in section 5.

2 Decoding Framework of SSM

2.1 SSM and Decoding

SSM represents the observation sequence with variable length by a fixed-length region sequence. Given an observation segment \( x_1^N = \{x_1, x_2, \ldots, x_N \} \), it would be mapped into a fixed-length frame sequence \( y_1^L \) using the resample function [4,5]. That is:

\[
y_i = x_{\left\lfloor \frac{L}{N} \right\rfloor}^i, \quad 0 < i \leq L.
\]

where \( \left\lfloor z \right\rfloor \) is the maximum integer no larger than \( z \). As shown in Fig. 1, if the observation segment is \( (\tau, m) \), the resample function will map \( (\tau, m) \) into \( L \) frames.

Two sets are involved in SSM decoding mainly [4]. One is candidate set, and the other is expanding set. When the decoding of all the observation segments which take \( m \) as the ending frame is finished, the reserved models would make up the candidate set of frame \( m \) presented by \( \text{CandSet}(m) \). Then the expanding set \( \text{ExpSet}(m) \) can be generated by expanding the models in \( \text{CandSet}(m) \) through lexicon.

SSM decoding is as follows [4]:

1. The expanding set of first frame is only initialized to include the silence model.
2. Beginning with 2 as the frame \( m \), the scores of all models in \( \text{ExpSet}(\tau) \) are calculated for estimating \( (\tau, m) \). Given by:

![Fig. 1. Decoding framework of SSM](image-url)
\[ D_m(\tau, \alpha) = \left\{ \ln(\frac{p(x_m^\tau | \alpha)}{m - \tau}) + \ln[p(\alpha)] + \ln[p_s(x_m^\tau | \alpha)] \right\}. \]  

(2)

where \( 0 \leq \tau < m < T \), \( D_m(\tau, \alpha) \) is the total score for estimating \((\tau, m)\), \( p(\alpha) \) is the language score, \( p(x_m^\tau | \alpha) \) is the SSM score and \( p_s(x_m^\tau | \alpha) \) is the duration score.

3. For all the segments ending in frame \( m \) with allowable duration \( L_{\max} \) in Fig. 1, the total scores of global paths are calculated respectively as follows:

\[ J^*_m(\tau, \alpha) = \{ J^*_m(\alpha) + D_m(\tau, \alpha) + C \} \]
\[ J^*_m(\alpha) = \max_{\beta, \tau} \{ J^*_m(\tau, \beta) \}, \]
\[ \beta \rightarrow \alpha, \tau < m, \beta \in \text{ExpSet}(\tau), \]
\[ J^*_0(\text{silence}) = 0 \]

(3)

where \( J^*_m(\tau, \alpha) \) is the total likelihood score of global path, one of whose boundaries locates at \( \tau \). \( C \) is the penalty factor, \( J^*_m(\alpha) \) is the maximum likelihood score of global path whose expanding model is \( \alpha \) at \( m \); \( \beta \rightarrow \alpha \) means that \( \beta \) expands into \( \alpha \) in frame \( m \).

4. \( \text{CandSet}(m) \) is formed from the above two steps, and then \( \text{ExpSet}(m) \) would be acquired from \( \text{CandSet}(m) \). Set \( m = m + 1 \), and go back to Step 2 until \( m = T \).

5. The one-best path could be obtained by backtracking \( \max_{\alpha} \{ J^*_T(\alpha) \} \).

### 2.2 Computational Complexity of SSM Decoding

From (2) and (3), many models are involved into the likelihood estimations for all of observation segments ending at frame \( m \). Supposing that in Fig. 1, the allowable duration of each segment is \( L_{\max} \), the total length of observation sequence is \( T \), the number of models is \( |\Omega| \) and the time for estimating each segment is \( C_{\text{seg}} \) approximately. The complexity of SSM decoding is \( O(T |\Omega| L_{\max} C_{\text{seg}}) \). In our test speech corpus, \( T \) is about 450 on the average, and the decoding system refers to more than 20,000 models, the amount of computation is very huge.

It would be useful for speeding up decoding to reduce the number of involved models, which is the basic concept of multistage pruning. The detail of this algorithm is presented in the following section.

### 3 Fast Decoding Algorithm

#### 3.1 Multistage Pruning

From Step 2 of SSM decoding, the involved models consume a large amount of calculations, but only a small part of them are selected to be expanded in Step 4, indicating most of the calculations are unnecessary in Step 2. In HMM, these calculations were pruned well [6], and the similar method was also introduced into
SSM decoding [3]. The decoding for an observation segment is divided into multi-stages. Then in each stage, for each SSM, we calculate probabilities of part regions, which are used to evaluate the model score. Those models with low scores would be pruned. When the last stage is approached, most of non-matched models are pruned. Given by:

\[
\ln[p(x^N_i | \alpha)] = \sum_{k=0}^{K-1} \sum_{i=0}^{L} \ln[p(y_i | \alpha, r_i)] \delta_k(i) .
\]  

(4)

where \( \delta_k(i) = 1 \) if \( i = k \), otherwise \( \delta_k(i) = 0 \), \( K \) is the total stages. \( r_i \) denotes \( i \)-th region and \( p(y_i | \alpha, r_i) \) is the \( i \)-th region probability for model \( \alpha \). \( p(x^N_i | \alpha) \) is the model score for estimating \( x^N_i \). Due to pruning, the score calculations for most models in (4) are not finished. The pruning threshold \( \beta(n) \) is calculated dynamically in stage \( n \). Given by:

\[
\beta(n) = \max_{\alpha} \left[ \sum_{k=1}^{n} \sum_{i=0}^{L} \ln[p(y_i | \alpha, r_i)] \delta_k(i) \right] - \lambda \cdot n \cdot N / K .
\]

(5)

where \( N \) is the duration of \( x^N_i \) and \( \lambda \) is the pruning control parameter. Formula (5) indicates that the models would be pruned, if their scores are lower than the highest score beyond a certain range in stage \( n \).

The threshold could be adjusted by changing parameter \( \lambda \) in (5). If \( \lambda \) is too small, these right models may be pruned. Previous experiment showed that 400 for \( \lambda \) is acceptable [4]. When using this algorithm in Mandarin LVCSR system, a lot of decoding time is saved [5].

3.2 Parallel Processing of Neighboring Segments

In mandarin speech, the boundary of two phones is fuzzy and may locate at any point of neighboring frames. In Fig. 2, each of 26, 27 and 28 is possibly true boundary between

---

Fig. 2. Decoding based on single segment and multi-segments
ing 4 and zh. If the decoder decides that the frame of 27 is a possible boundary, it may do the same decision to 26 and 28, because these frames have similar features. Similarly, there also exist neighboring segments sharing similar acoustic spaces. Like (10,18) and (11,19), they have identical features from 11 to 18. Intuitively, the neighboring segments should have a consistent pruning rule for decoding. This viewpoint represents the core-point of parallel decoding algorithm. If neighboring segments are decoded in parallel, the decoder is able to make the best of model scores in a wide scope, and generate an optimal pruning threshold for the neighboring segments. When multistage pruning is combined into this algorithm, a lot of decoding time should be saved.

Now the definition of neighboring segments is introduced. Given a hypothetical segment, floating its starting frame and ending frame at limit ranges, it would form multi-segments which are called neighboring segments represented as follows:

\[ \text{Neb}((st_1, ed_1)) \]

where \( st_1 \) and \( ed_1 \) represent the ranges of starting frames and \( ed_1 \) for ending frames.

As shown in Fig. 2, \( \text{Neb}((10,14) \rightarrow (19,20)) \) describes the neighboring segments of \{ (10,19), (10,20), (11,19),..., (14,19), (14,20) \} which are decoded in parallel. Given by:

\[
Pr(b(st, ed, K, \alpha) + \sum_{l=0}^{K-1} \ln[p(y_i | \alpha, r)] \delta_i(i)).
\]

where \( st_1 \leq st \leq st_2, ed_1 \leq ed \leq ed_2 \). \( \text{Pr}(b(st, ed, n, \alpha) \) is the model score for estimating \((st, ed)\) in the stage \( n \). The threshold \( \beta(n) \) is replaced by \( \beta'(n) \):

\[
\gamma(n) = \max_{st_1 \leq st \leq st_2, ed_1 \leq ed \leq ed_2} \max_{\alpha} \left\{ \left( \text{Pr}(b(st, ed, n, \alpha) \right) \right\}.
\]

\[
\beta'(n) = \gamma(n) - \lambda n (ed - st + 1) / K.
\]

\( \gamma(n) \) is the highest score for estimating neighboring segments in stage \( n \), and it is usually bigger than the values given by the first part of formula (5). As already explained, \( \gamma(n) \) is shared by neighboring segments in (9), so \( \beta'(n) \) can provide a bigger pruning threshold than \( \beta(n) \) for the likelihood estimation of \((st, ed)\). When this fast algorithm is integrated into an SM-based Mandarin LVCSR system, about 50% time is saved.

Essentially, the parallel decoding algorithm is helpful to find the local highest score in a wide time domain, and to select a best boundary among fuzzy boundaries.

4 Experiment and Analysis

4.1 Experimental Setup

The testing data in our experiment are 863 continuous speech corpora which include 240 sentences of six males and last for 1027 seconds. MFCC feature of 39 dimensions is used, and the window length is 25.6ms with 10ms frame shifting.
We have three systems in our experiment. The first one is HMM system built by HTK [6]. The second is SSM base-line system, and the third is SSM-n. The HMMs and SSMs are built for context dependent syllables initial/final. In the HMM system, each model with left to right topology has 5 states including the first state, 3 emission states and the final state. Every emission state is simulated by 16-mixture GMM. The model in both SSM systems consists of 15 regions simulated by GMMs of 12 mixtures. The SSM base-line system only uses multistage pruning algorithm and the SSM-n is integrated with the fast algorithm by parallel processing of neighboring segments. There are no differences among SSM-1, SSM-2, SSM-3, except for the floating frame ranges. 2-gram language models are also introduced in the three systems. Because HTK with language models is much slower than the real system, this experiment does not compare the decoding time of HMM with SSM.

4.2 Results and Analysis

In our experiment, the value of $\lambda$ in (5) and (9) is set to 400. The results of syllable recognition for five groups are given in Table 1. In this table, $st2-st1$ and $ed2-ed1$ represent the ranges of the starting and ending frames in SSM-n system. The value of 0 means no floating. Sub, Del, and Ins represent the substitution error, the deletion error and the insertion error respectively.

Both of SSM systems have lower error rates than HMM system from Table 1, indicating SSM has better modeling performance than HMM. From Fig. 3, the SSM base-line system can not achieve the real-time requirements well, which explains SSM’s weakness when applied into LVCSR system. The SSM-n system saves about 50% time

**Table 1. The recognition results**

<table>
<thead>
<tr>
<th>Systems</th>
<th>$st2-st1$</th>
<th>$ed2-ed1$</th>
<th>Sub%</th>
<th>Del%</th>
<th>Ins%</th>
<th>Err%</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>/</td>
<td>/</td>
<td>15.64</td>
<td>1.34</td>
<td>0.10</td>
<td>17.08</td>
</tr>
<tr>
<td>SSM</td>
<td>/</td>
<td>/</td>
<td>13.29</td>
<td>0.25</td>
<td>0.10</td>
<td>13.64</td>
</tr>
<tr>
<td>SSM-1</td>
<td>7</td>
<td>0</td>
<td>13.13</td>
<td>0.26</td>
<td>0.00</td>
<td>13.39</td>
</tr>
<tr>
<td>SSM-2</td>
<td>5</td>
<td>1</td>
<td>13.90</td>
<td>0.19</td>
<td>0.03</td>
<td>14.12</td>
</tr>
<tr>
<td>SSM-3</td>
<td>1</td>
<td>5</td>
<td>14.44</td>
<td>0.32</td>
<td>0.03</td>
<td>14.79</td>
</tr>
</tbody>
</table>

**Fig. 3. The comparison of runtimes**
compared to its base-line system without losing recognition accuracy. A conclusion is 
drawn that the fast algorithm by parallel processing of neighboring segments could make 
a great improvement to accelerate decoding for SSM-based LVCSR system.

For the three conditions of SSM-n system, the SSM-1 has the best recognition 
results, it has not only decreased the error rate, but also saved a half of decoding time. 
The SSM-1 only floats the starting frames of neighboring segments. We know both of 
candidate set and expanding set are involved into decoding for every frame from 
Section 2. If the decoder floats the ending frames, due to characteristics of forward 
decoding, the system has to allocate more memory for storing models from the both 
sets. So it is better not to float the ending frame. The SSM-2 and SSM-3 have accuracy 
loss compared to the base-line system. Maybe it is not suitable to float all neighboring 
segments in a fixed-range. From Table 1, it is obvious that the floating ranges of 
neighboring segments have great influence on the recognition results.

5 Conclusions

In this paper, observation segments with similar acoustic spaces were decoded in 
parallel. This parallel algorithm generated an optimal threshold for pruning 
unnecessary calculations during decoding, and saved a half of decoding time when it 
was integrated into SSM-based Mandarin LVCSR system. The experiment not only 
compared the performance of SSM with HMM system, but also shown the good 
performance of the fast algorithm which processed neighboring segments in parallel.

Acknowledgments. This work was supported in part by the China National Nature 
Science Foundation (No. 60675026, No. 60121302, No. 9082001), the 863 China 
National High Technology Development Projects (No. 20060101Z4073, No. 
2006AA01Z194) and the National Grand Fundamental Research 973 Program of China 
(No. 2004CB318105).

References

1. Ostendorf, M., Roukos, S.: A Stochastic Segment Model for Phoneme Based Continuous 
(1989)
2. Ostendorf, M., Digalakis, V., Kimball, O.: From HMM’s to Segment Models: A Unifled 
View of Stochastic Modeling for Speech Recognition. IEEE Transactions on Speech Audio 
3. Digalakis, V., Ostendorf, M., Rohlicek, J.: Fast Algorithms for Phone Classification and 
Recognition Using Segment Based Models. IEEE Transactions on Signal Processing 40(12), 
2885–2896 (1992)
Submitted to Institute of Automation. China Academy of Sciences for the Degree of Doctor 
of Engineering (2006)
Trajectory Segment Model by Avoidance of Redundant Computation on Segment. 
International Journal of Computational Linguistics and Chinese Language Processing 11(1), 
73–86 (2006)