International Symposium on Chinese Spoken Language Processing (ISCSLP 2004)
Hong Kong
December 15-18, 2004

# A FRAMEWORK FOR FAST SEGMENT MODEL BY AVOIDANCE OF REDUNDANT COMPUTATION ON SEGMENT*

*Yun Tang, Wenju Liu, Yiyan Zhang and Bo Xu*

National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences
{tangyun, lwj}@nlpr.ia.ac.cn, xubo@hitic.ia.ac.cn

## ABSTRACT

Segment model (SM) is a family of methods by using segmental distribution rather than frame-based features (e.g. HMM) to represent the underlying characteristics of observation sequence. It has been proved to be more precise than that of HMM. However, the high complexity prevents these models from practical system. In this paper we present a framework to reduce the computational complexity of segment model by fixing the number of the basic unit in the segment to share the intermediate computation results. Our work is twofold. First, we compared the complexity of SM with HMM and proposed a fast SM framework based on the comparison. Second we use two examples to illustrate this framework. The fast SMs have better performance than the system based on HMM, and at the mean time, we successfully keep the computation complexity of SM at the same level of HMM.

## 1. INTRODUCTION

HMM has been used successfully for acoustic modeling in many speech recognition systems. Assuming that feature vectors are conditionally independent given the state sequence, the task of extracting the trajectory can be elegantly achieved by Viterbi algorithm frame by frame. However, this assumption is far from reality, which limits the ability of HMM to grasp the relations within a segment. The state in HMM is a stationary process and the output of these states are independent. It is a not enough to represent a non-stationary observation sequence by a piecewise state sequence [6]. In order to handle these problems, a lot of methods have been proposed. SM [1] is a family of methods among them.

The well acoustic modeling of SM is at the cost of much higher computation than that of HMM, which prevents SM from being applied in practical system. The high complexity of SM is due to the evaluation on segment score which cannot be decomposed and the intermediate information of score evaluation is not shareable between different segments even for the case that two segments only differ by one frame. Most of works accelerating the SM are focused on efficient pruning algorithm [8] [9]. These algorithms do speed up SM greatly but they are still far slower than HMM, since the computation of these algorithms is based on segment while HMM is based on frame. In our study, we propose a framework to reduce the complexity of the SM, which divide the computation of segment scores into frame-based scores and sharing the intermediate computation results. Guided by this frame-work, we proved the complexity of Stochastic Segment model (SSM) [2], one kind of SM, is not proportional to the product of the number of the models and the maximum number of allowable durations but is only related to the number of the models, or more exactly, the number of regions (states) in model, which keeps the same level of complexity as the HMM's. We have also greatly enhanced the speed of the Parametric Trajectory model (PTM) [5] [6], another kind of SM, after some minor modifications to the original algorithm.

This paper is organized as follows. A comparison of computation complexity between the SM and the HMM is addressed in the next section. Then we propose the fast SM framework and use two examples to illustrate it, the fast SSM and the fixed PTM in section 3. Section 4 shows the experimental results of the fast SM. Finally, conclusions are given in section 5.

## 2. COMPARISON BETWEEN HMM AND SM

Given sequence for a sentence, $x_1^T = \{x_1, x_2, ..., x_T\}$, the decoding process for HMM is (Viterbi algorithm):

$$J_m^*(a,i) = p(x_m \mid a,i) + \max_{i-2 \le j \le i} (J_{m-1}^*(a,j)). \quad (1)$$

$$1 \le m \le T, \quad 1 \le a \le |\Omega|, \quad 1 \le i, j \le L_a$$

where $J_m^*(a,i)$ is the maximum accumulated state sequence score in time point $m$, given the state $i$ and reference $a$; $p(x_m \mid a,i)$ is the state score of the frame $x_m$ at the state $i$ of

* This work was supported in part by the China National Nature Science Foundation (No. 60172055) and the Beijing Nature Science Foundation (No.4042025).

the reference $a$ ; $|\Omega|$ is the number of references, $L_a$ is the number of states for reference $a$ .

The formula is applied to all internal states of each segment, (i.e., $i \geq 2$ ). At the boundary of the reference pattern, i.e., $i = 1$ , the formula is reduced to

$$J_m^*(a,1) = p(x_m \mid a,1) + \max\left[\max_{1 \leq b \leq |\Omega|}[J_{m-1}^*(b, L_b), J_{m-1}^*(a,1)]\right]. \quad (2)$$

The final solution for the best path is

$$J^* = \max_{1 \leq a \leq |\Omega|}[J_T^*(a, L_a)] \quad (3)$$

and the best path can be gotten by backtracking the best final score.

According to (1) and (2), the total cost of the Viterbi algorithm is essentially the cost of computing the state scores. The computation of the state scores is proportional to the number of states in each reference model and the observation sequence length. For a searching algorithm without pruning, the approximate time complexity is $O(T \cdot |\Omega| \cdot \bar{L} \cdot C_s)$ and the space complexity is $O(|\Omega| \cdot \bar{L} \cdot T)$ , where $C_S$ is the time cost of computing $p(x_m \mid a, n)$ and $\bar{L}$ is the average number of states in each model.

The decoding process for SM is:

$$J_m^* = \max_{t,a}\{J_t^* + \ln[p(x_t^m \mid a)](m-t) + l \ln(P_S(x_t^m \mid a)) + \ln[P(a)] + C\} \quad (4)$$

where $J_m^*$ is the accumulated score of the best reference sequence ending at time point $m$ ; $P_m(x_t^m \mid a)$ is the segment score for segment $x_t^m$ ; $P_S(x_t^m \mid a)$ is segmental level information such as duration distribution and $C$ is the penalty factor for each segment.

The best segment sequence can be obtained by back-tracking from the best final score $J_T^*$ too.

At each time point, number of $L_{\max}$ hypothesis segments need to be examined for each reference model and this is the main computation in the SSM. So the time complexity of the SM is $O(C_{Seg} \cdot T \cdot |\Omega| \cdot L_{\max})$ , where $L_{\max}$ is the maximum number of allowable model durations and $C_{Seg}$ is the time cost of a segment. $C_{Seg}$ is comparable to $C_S \cdot \bar{L}$ in the HMM or even more complex. The space complexity is $O(T)$ . The SM is more costly than the HMM.

## 3. FAST FRAMEWORK OF SM

The efficiency of Dynamic programming lies in the reduction of computation on overlapping sub-problems, which includes two aspects: one is the overlapping computation on the basic unit and the other is the overlapping operations to recombine these units into different final results at some definite constraints. A sub-problem is called overlapping if it has to be solved over and over again. As far as SM and HMM concerned, the overlapping unit is the segment in SM and the frame in HMM. The ratio of time cost for the overlapping unit in SM is enormously higher than that in HMM. Computation of the basic unit scores is an extremely time consuming process in traditional SSM. In our test, 97.6% of the time has been spent to obtain the overlapping unit scores in SSM while the percentage in HMM is 51.4%.The main computation of the SM is due to the computation on the overlapping unit.

Most SMs can be categorized to the Constrained Mean Trajectory SM (CMTSM) [1], such as SSM and PTM. CMTSM assumes that the region is independent from other regions given the segment; the mapping of feature vector to region is related to the duration of segment and its position in the segment. That means the computation of frame score in specific region is irrelevant to other frames in the same segment or other regions in the same reference and the segment score can be calculated by summing up these frame scores in the segment in a linear way without complex operations, e.g., dynamic time warping.

$$\ln p(x_1^N \mid a) = \prod_{i=1}^{N} \ln p(x_i \mid a, r_i) \quad (5)$$

This assumption guarantees to dissolve the overlapping unit from segment into frames and recombine these frame scores to obtain the segment score in an efficient way. The overlapping property of the frame-based unit is another question we need to consider since not all SM algorithms that belong to the CMTSM will meet this property. Usually, the overlapping property for frame-based unit in SM is hard to meet. In this case, we use a linear time re-sampling to deal with the variable length of segment and region scores can be shared by other segment. Given speech sequence $x_1^N$ , a fixed length sequence $y_1^L$ can be obtained by [2]:

$$y_i = x_{\left\lfloor \frac{i}{L} N \right\rfloor}, \quad 0 \leq i < L \quad (6)$$

where $\lfloor x \rfloor$ is the largest integer $n \leq x$ .

In short, to speed up a specific CMTSM, we will check the overlapping property of the frame-based unit, e.g., region. If the condition doesn't meet, we fix the number of regions in model and use the linear time re-sample to make the model uniform. In implementation, when moving to a new time point in the observation sequence, we first compute all the region models in active reference patterns and store them in a huge table for later retrieval; then we obtain the "state" score of dynamic programming at current time point by summing up the region scores at the current segment. The computation for each time point is the incremental part for the segment and $R-1$ addition operations to obtain the segment score for each reference pattern instead of the computation on the whole segment. The fast SM algorithm is described as follows:

1. Make the segment model uniform;

2. Initialize: $m=0$, $J_0^*=0$ $p(x_0 \mid a, r_i)=0$, $a \in \Omega$, $r_i \in [1,R]$;
3. $m=m+1$, $t=m$, if $m>T$ goto 6; else compute $p(x_t \mid a, r_i)$, $a \in \Omega$, $r_i \in [1,R]$;
4. $t=t-1$, if $t=m-L_{\max}$ or $t=0$, $\ln P(x_t^m \mid a)=C_{Min\,\Pr}$, else $\ln P(x_t^m \mid a)=\ln P(y_1^R \mid a)=\sum_{i=1}^{R} \ln p(y_i \mid a, r_i)$;
5. $J_m^*=\max_{t,a}\{J_t^*+\ln[P(x_t^m \mid a)](m-t)+l\ln(P_S(x_t^m \mid a))+\ln[P(a)]+C\}$ $f_m=\{t,a\}$ goto 2;
6. Trace back $J_T^*$ to get the optimal path;
7. Terminate.

where $y_1^R$ is linear time re-sampling sequence of $x_t^m$.

This is the framework we proposed to reduce the complexity of SM. We will use two more examples to illustrate the framework.

### 3.1. Complexity of SSM

SSM represents observation sequence by a fixed length time alignment region sequence. The linear time re-sample described above is used to map the variable length segment $x_1^N$ to the fixed length model region sequence $y_1^L$. The log conditional probability of a segment $x_1^N$ given model $a$ is:

$$\log[p(x_1^N \mid a)]=\sum_{i=1}^{L}\log[p(y_i \mid a, r_i)]+l\log[P(N \mid a)] \quad (7)$$

where $P(N \mid a)$ is the duration distribution of segment given $a$. According to (7), $C_{Seg}$ is proportional to the number of regions in the model and can be represented as $C_R \cdot \bar{L}$, where $C_R$ is the time cost of unit model $p(y_i \mid a, r_i)$ and $\bar{L}$ is the average number of regions model. And the complexity of the SSM is $O(C_R \cdot T \cdot |\Omega| \cdot \bar{L} \cdot L_{\max})$.

From the analysis above, SSM meets the conditions of the fast SM framework. The computation of segment score in (7) can be distributed to each region of the model. The region is the overlapping unit we referred above and the overlapping property is guaranteed by the linear time re-sampling. The total cost of the SSM algorithm is essentially the cost of computing the region scores. So the time complexity of the computation on region is $O(C_R \cdot T \cdot |\Omega| \cdot \bar{L})$ and the space complexity is $O(|\Omega| \cdot L_{\max} \cdot \bar{L})$. The complexity of the SSM is at the same level as the HMM's and is not proportional to the number of allowable durations.

### 3.2. Fast PTM

In PTM, the features in a segment are modeled by parameterization through constant, linear, or higher order polynomial regression instead of using a sequence of regions to represent the curve of trajectory. A speech segment $x_1^N$ given model $a$ can be modeled as:

$$x_i=\sum_{p=0}^{P} B_a(p)\left(\frac{i-1}{N-1}\right)^p+E_i(\Sigma_a) \quad (8)$$

where $B_a(p)$ is the polynomial regression coefficient of order $p$ and $E_i$ is a residual error with covariance matrix $\Sigma_a$ after data fitting by the first term in (8). The frame score $p(x_i \mid a, r_i)$ in (5) given duration $N$ is:

$$p(x_i \mid a, r_i)=\frac{1}{(2p)^{d/2}\mid \Sigma_a \mid^{1/2}}\exp\{-\frac{1}{2}(x_i-\sum_{p=0}^{P} B_a(p)\left(\frac{i-1}{N-1}\right)^p)^T \Sigma_a^{-1}(x_i-\sum_{p=0}^{P} B_a(p)\left(\frac{i-1}{N-1}\right)^p)\} \quad (9)$$

In the original method, the number of regions is variable to accommodate the variable duration of segment. Though PTM meets the function (5), the basic units in PTM are changed as the duration $N$ changed and no longer meet the requirement of overlapping. For example, two segments for the same reference both begin at time point 1, the first one ends at 10 and the other ends at 15. The durations of both segments are different, so the $P$-order polynomials in (8) are different too, which leads that the frame scores in (9) can't be shared in different segments.

In fast PTM, we fix the number of region sequence in the reference and use the linear time re-sample to map the variable length segment to the region sequence with fixed duration. The overlapping unit is the frame like structure $p(x_i \mid a, r_i)$ in (9) after fixing the number of duration in the model. The following steps are just similar to the SSM. In this way, it has greatly accelerated the PTM by slightly downgrading the performance of the system.

## 4. EXPERIMENTS AND RESULTS

### 4.1. Experiment Corpus

Speech Materials: the Mandarin digit string database includes 55 males' speech and each one has 80 utterances. The length of each utterance varies from 1 to 7 digits and the average length is 4. The vocabulary is "0" to "9"and "yiao1". Statistical results show that all digits have the same probability to be uttered, and the connections of digits are considered and balanced. At the mean time, positions (start/middle/end) of the digits in the string are balanced too [4]. We take the first 40 speakers as the training set and the remaining 15 speakers as the test set.

Baseline System: There are three kinds of system, HMM, SSM and PTM. In all experiments diagonal covariance matrix is assumed.

HMMI : 8 Sts, 16 MCs, Uni
HMMII: 8Sts, 16MCs, Tri
SMMI : 25Res, 5MCs, Uni
SMMII: 40Res, 10MCs, Uni

PTM : 15MCs, Uni
where Sts: states, MCs: mixture components, Res: regions, Uni: Uniphone, Tri: Triphone
HMMII integrates duration distribution by A* algorithm.
The PTM used as baseline is consisted of three sub-segments [6].

### 4.2. Experiment results

Table 1 compares the modeling ability of HMM and SSM. From the experiments we can see that SSM can model more precisely than HMM. SSM not only performs better than the HMMI, whose total number of multidimensional pdfs is approximately equal to SSMI, but also excels the HMMII. As the number of regions and mixture components increase, the SM has great potential to achieve higher performance. SSMII has achieved 95% recognition results for mandarin digit strings.

*Table 1:* Comparison of Digit string recognition between SSM and HMM

| | IW Cor. | S Corr. | Ins err | Del err | Sub err |
|---|---|---|---|---|---|
| HMMI | 96.12% | 87.10% | 0.64% | 2.14% | 1.10% |
| HMMII | 97.47% | 91.80% | 0.19% | 0.87% | 1.47% |
| SSMI | 98.50% | 92.52% | 0.23% | 0.72% | 1.63% |
| SSMII | 99.25% | 95.00% | 0.33% | 0.31% | 1.04% |

IW Cor: isolate word correction, S Cor: string correction, Ins err: insertion error, Del err: deletion error, Sub err: substitution error
The fast PTM realized for comparison fixed the number of regions to 60 and each sub-segment in the model has 20 regions. The feature frames in segment for the PTM are divided to 3 sub-segments according to the time evenly and map to 3 sub-segments separately. Other parameters are the same with the baseline system. Table 2 presents the recognition results of the fixed PTM and the original PTM. We can conclude from the experiment that the performance of PTM system has been slightly downgraded after these modifications, but it is acceptable.

*Table 2*: Recognition results of PTM and Fixed PTM

| Methods | IW Corr. | S Corr |
|---|---|---|
| PTM | 99.38% | 95.10% |
| Fixed PTM | 98.24% | 94.5% |

The efficiency of different recognition systems are compared in the table 3, including the classic SSM, fast SSM, PTM, fixed PTM and HMM based system. We use the utterances of one person (80 strings) for test. As is shown by table 3, the fast algorithm does boosting the SM and reduce the complexity of SM to the same level of HMM. The most noticeable achievement is obtained in fixed PTM, which is 90 times faster than original PTM.

*Table 3:* Time Comparison of SM, Fast SM, and HMM

| | T (s) | | T(s) |
|---|---|---|---|
| HMMI | 35 | HMMII | 87 |
| Classic SSMI | 1816 | PTM | 23854 |
| Fast SSMI | 101 | Fast SSMII | 162 |
| Fixed PTM | 271 | | |

## 5. CONCLUSIONS

Though the similar skill has been applied in HMM for a long time, it is first time to be proposed in SM (CMTSM). The main reason is that SM models the observation sequence in segment rather than in frame and it is natural to compute the segment score integrally. The other is the variable length of segment, which results in the basic units are not sharable in different segments. Fast SM framework boosts the SM greatly by make uniform the basic unit in SM and sharing the intermediate results of segmental computation. We also use two examples to illustrate this framework: SSM and PTM, which are far more effective than the original model with comparable performance. By this framework, we see the potential to implement the SM to LVCSR in current computation condition and it will be our focus of future work.

## 6. REFERENCES


[1] M. Ostendorf et al, "From HMM's to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition." *IEEE Trans. Speech Audio Processing*, 1996.
[2] M. Ostendorf et al "A stochastic segment model for phoneme based continuous speech recognition," *IEEE Trans. Acoust. Speech. Signal Processing*, vol. 37, 1989.
[3] L. R. Rabiner et al "High Performance Connected Digit Recognition, Using Hidden Markov Models**,"** *IEEE Trans. Acoust. Speech. Signal Processing*, 1989,
[4] Yonggang Deng et al "Towards high performance continuous Mandarin digit string recognition," in *Proc. Int. Conf. on Spoken Language Processing*, vol3, 2000.
[5] H. Gish et al, "Secondary Processing using Speech Segments for an HMM Word Spotting System," in *Proc. Int. Conf. on Spoken Language Processing* , 1992
[6] Li Deng et al, "Speech Recognition Using Hidden Markov Models with Polynomial Regression Functions as Non-stationary States," *IEEE Trans. Speech Audio Processing*, vol. 2, 1994
[7] L. Rabiner and B. H. Juang, "Fundamentals of speech recognition," *Prentice Hall PTR*, 1993.
[8] V.V. Dugakakis, et al, "Fast Algorithms for phone classification and recognition using Segment-based Models," *IEEE Trans. Speech Audio Processing*, 1992
[9] Steven C. Lee and James R. Glass, "Real-Time Probabilistic Segmentation for Segment-Based Speech Recognition," in *Proc. Int. Conf. on Spoken Language Processing*, 1998