# Segmental Semi-Markov Model Based Online Series Pattern Detection Under Arbitrary Time Scaling*

Guangjie Ling[1,2], Yuntao Qian[1,2], and Sen Jia[1,2]

[1] College of Computer Science, Zhejiang University, Hangzhou, 310027, P.R. China
[2] State key Laboratory of Information Security,
Institute of Software of Chinese Academy of Sciences, Beijing, 100049, P.R. China
balley_ling@163.com, ytqian@zju.edu.cn, zjujiasen@hotmail.com

**Abstract.** Efficient online detection of similar patterns under arbitrary time scaling of a given time sequence is a challenging problem in the real-time application field of time series data mining. Some methods based on sliding window have been proposed. Although their ideas are simple and easy to realize, their computational loads are very expensive. Therefore, model based methods are proposed. Recently, the segmental semi-Markov model is introduced into the field of online series pattern detection. However, it can only detect the matching sequences with approximately equal length to that of the query pattern. In this paper, an improved segmental semi-Markov model, which can solve this challenging problem, is proposed. And it is successfully demonstrated on real data sets.

## 1 Introduction

In recent years, online series pattern detection technique has attracted increasing interest in time series data mining communities, as it plays an important role in many applications such as endpoint detection in plasma etch processes and pattern detection in medical data. Efficient online detection of similar patterns under arbitrary time scaling of a given time sequence (see Fig. 1) is a challenging problem in the real-time application field of time series data mining. For example, persons reproduce the same tune or motions at different speeds [1, 2], and many financial time series also contain such similar patterns [3]. Readers are referred to [4] for details.

Some methods based on sliding window have been proposed to solve this problem. Although their ideas are simple and easy to realize, their computational loads are very expensive. So model based methods are proposed. Recently, the segmental semi-Markov model [5, 6] is introduced into the field of online series pattern detection. However, it can only detect the matching sequences with approximately equal length to that of the query pattern [7-9]. In this paper, an improved segmental semi-Markov model, which can solve this challenging problem, is proposed. And it is successfully demonstrated on real data sets.

* This research is supported partly by Science and Technology Project of Zhejiang (2006C21001).

X. Li, O.R. Zaiane, and Z. Li (Eds.): ADMA 2006, LNAI 4093, pp. 731 – 740, 2006.

First, some symbols to be used throughout this paper are summarized in Table 1. Then the online pattern detection can be described as follows. Given a real-time time series $D$ and a query time series $Q$ which is acquired by prior knowledge (where $|D| >> |Q|$), and a scaling factor $l$, $l \geq 1$, which represents the maximum allowable stretching and shrinking of $Q$ by $l$ and $1/l$ respectively, the matching sequences of $Q$ in $D$ for any scaling range specified by $l$ are located.

**Table 1.** Summary of symbols

| Symbols | Definitions |
|---|---|
| $D$ | Real-time time series |
| $Q$ | Query pattern |
| $\|X\|$ | Length of sequence $X$ |
| $X[i]$ | The $i$-th entry of sequence $X(1 \leq i \leq \|X\|)$ |
| $X[i...j]$ | Subsequence of $X$ ,including entries from the $i$-th to the $j$-th |
| $l$ | Scaling factor, $l \geq 1$ |

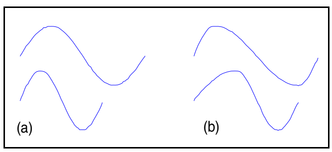


**Fig. 1.** Similar patterns under different time scaling. (a)Uniform scaling. (b)Arbitrary scaling.

## 2 Online Pattern Detection Methods Based on Sliding Window

Sliding window is a typical approach for online detection of similar patterns. The approach begins at the initial position of $D$, gets a window of minimum size $W_{min}$ (where $W_{min}$ is the lower scaling bound specified by $l$, $W_{\min} = \lfloor |Q| / l \rfloor$), then checks whether $D[1...W_{min}]$ matches $Q$ under some similarity measure. With the left side of the window anchored at $D[1]$, each subsequence $D[1...k]$ is scanned orderly in a similar manner to check if it matches $Q$ for all $W_{\min} \leq k \leq W_{\max}$ (where $W_{max}$ is the upper scaling bound specified by $l$, $W_{\max} = \lceil |Q| \cdot l \rceil$). Repeat the same procedure with the window anchored at position $D[2]$, then $D[3]$ etc., until end of $D$.

There are many methods proposed to match similar patterns under time scaling. Keogh *et al* [2] use uniform scaling and Euclidean distance to match similar patterns. This method can only deal with the similar patterns under uniform scaling, as shown in Fig. 1(a). Similarly, the limitation also holds for the "CD-Criterion" technique [10].

Dynamic time warping (DTW) [11] distance compares sequences of different lengths by stretching them, so it can be used to measure the similar patterns under arbitrary time scaling. But some disadvantages have been found in practice, e.g., it may introduce fault matching patterns due to local over-scaling, and its time complexity is $O\left(|D|\cdot|Q|^3\cdot(l^2-1/l^2)\right)$, which is unsuitable for the real-time application.

Fu *et al* [4] utilize scaled and warped matching (SWM) and its corresponding lower bounding technique to look for similar patterns under arbitrary time scaling. Considering the left side of the window anchored at $D[i]$, the lower bounding technique starts by calculating the lower bounding distance between $Q$ and all subsequence beginning with $D[i]$ in the range specified by $l$. If the distance exceeds the user-specified tolerance, we can be sure that there are no matching patterns of $Q$ starting at $D[i]$, and the left side of the window can slide to $D[i+1]$; otherwise, using SWM to check whether there exists subsequence similar to $Q$. We call this method SWM_LB for short. The pruning power $P$ describes the effectiveness of lower bounding technique, which is defined as follows [4]:

$$P=\frac{\texttt{Number of objects that do not require full SWM}}{|D|}$$

And the time complexity of SWM_LB is $O\left(|D|\cdot\left(((l-1/l)+\rho)\cdot|Q|^2+(1-P)\cdot\left(\rho|Q|^3\cdot(l-1/l)\right)\right)\right)$, where $\rho$ is the fraction of $|Q|$ (the time warping constraint $r=|Q|\cdot\rho$). Note that the larger $P$ becomes, the more efficient the algorithm would be. As far as we know, SWM_LB is best for online series pattern detection in all sliding window based methods, so we empirically compare it to our approach in Section 5.

## 3 Segmental Semi-Markov Model

The basic theory of Hidden Markov Model (HMM) was proposed by Baum and his colleagues in the late 1960s and early 1970s [12]. For a HMM with the transition probability $P(s_t=j\mid s_{t-1}=i)=A_{ij}$, once in state $i$, the system will stay in it for $d$ time units, where $d$ has an implicit geometric distribution: $P(d)=A_{ii}^{d-1}(1-A_{ii})$. During the stay in state $i$, the system generates $d$ observations, which are conditionally independent and identically distributed.

The segmental semi-Markov model is an extension of the standard HMM. It was originally proposed in the speech recognition literature [5, 6], then Ge *et al* [8] introduced it into the field of online series pattern detection. The segmental semi-Markov model improves the standard HMM by introducing explicit state duration distributions [5] and segment observation models [6]:

a. The duration $d$ can have an explicit distribution that may be non-geometric, e.g., Gauss distribution, Poisson distribution.
b. The observations of every state can have an explicit distribution to model the dependence among them.

# 4 Improved Segmental Semi-Markov Model

## 4.1 Model Construction

The segmental semi-Markov model is a good solution to detect the matching sequences with approximately equal length to that of the query pattern. However, the corresponding segmental observations of similar patterns under arbitrary time scaling may differ considerably, as shown in Fig. 2 (left), so it can not solve the problem mentioned in Section 1. In this section, we propose an improved segmental semi-Markov model which modifies the existing model from the following three aspects:

*1. Introducing the offset distribution to replace the observation distributions.*
Assume the subsequence of the $i$th segment is $Y = y_1, y_2, \ldots, y_n$, and its corresponding sequence generated by linear regression function is $Y' = y_1', y_2', \ldots, y_n'$. The offset $R_i$ of the $i$th segment is defined as the root mean squared errors between $Y$ and $Y'$:

$$R_i(y_1 \ldots y_n, y_1' \ldots y_n') = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2} \quad . \tag{1}$$

The offset $R_i$ describes the fitting degree between the subsequences of the $i$th segment and its regression line. Form Equation (1) we know that, for any similar patterns, the offset $R_i$ is independent of other segmental observations. Hence, the difference between the observations of corresponding segments is allowable (see Fig. 2).

$R_i$ is governed by $P(R_i \mid \theta_{R_i})$, where $\theta_{R_i}$ is the set of parameters for the distribution. We use Gaussian distribution to describe the form of $P(R_i \mid \theta_{R_i})$:

$$P(R_i \mid \theta_{R_i}) \propto \begin{cases} N(\mu_{R_i}, \sigma_{R_i}^2), & R_i \geq \mu_{R_i} \\ N(R_i, \sigma_{R_i}^2), & R_i < \mu_{R_i} \end{cases} \quad . \tag{2}$$

with parameters $\theta_{R_i} = \{\mu_{R_i}, \sigma_{R_i}^2\}$.

*2. Introducing the amplitude ( $Y$ coordinate) difference distribution.*
Without the segmental observation distributions, the shape of a segment can not be modeled, so we introduce the amplitude difference distribution to perform this task.

Assume the time ( $X$ coordinate) of the endpoint of the $i$th segment is $t$, and the number of data points in the segment is $d_i$. Let us call the amplitude difference of the $i$th segment $\Delta Y_i$, and we define it as $\Delta Y_i = y_t - y_{t-d_i+1}$, as shown in Fig. 2(right).

$\Delta Y_i$ is governed by $P(y_t - y_{t-d_i+1} \mid \theta_{y_i})$, where $\theta_{y_i}$ is the set of parameters for the distribution. The actual form of $P(y_t - y_{t-d_i+1} \mid \theta_{y_i})$ depends on the specific application. Usually it would be the following Gaussian distribution:

$$P(y_t - y_{t-d_i+1} \mid \theta_{y_i}) \propto N(\mu_{y_i}, \sigma_{y_i}^2) \quad . \tag{3}$$

with parameters $\theta_{y_i} = \{\mu_{y_i}, \sigma_{y_i}^2\}$.

*3. Introducing an extra state—pre-pattern state, and proposing a method to compute its classification and transition probabilities.*
The pre-pattern state (state 0) models the data before the query pattern. Introducing this state is mainly to meet the needs of the online pattern detection, and it is first mentioned by Ge and Smyth [7]. However, it is difficult to determine the classification probability that any data belongs to the pre-pattern state ([7] does not specify it). Here we propose a method to solve this problem. We first specify the probabilities of the data belonging to states $1 \ldots K$ (assume the model has $K$ states), and then recalculate their values according to Equation (4), where $p_i^{query}$ is the probability of the data belonging to state $i$, which is the endpoint of the query pattern's $i$th segment:

$$p_i = \frac{p_i}{p_i^{query}} \quad i = 1 \ldots K \ . \tag{4}$$

Then compute the probability of the data belonging to the pre-pattern state:

$$p_0 = 1 - \sum_{i=1}^{K} p_i \ . \tag{5}$$

Finally we normalize the probabilities:

$$p_i = \frac{p_i}{\sum_{j=0}^{K} p_j} \quad i = 0, 1, \ldots K \ . \tag{6}$$

For transition probability of pre-pattern state, we set $A_{0,0} = 0$ and $A_{0,1} = 1$.

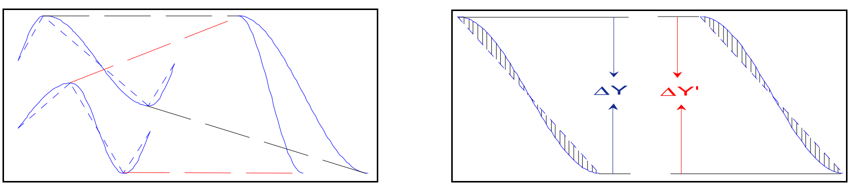


**Fig. 2.** (left) The second segmental observations of the two similar patterns differ considerably. (right) Two curves from the two similar patterns' second segments have equal amplitude difference, that means $\Delta Y = \Delta Y'$, and both fit their regression lines well.

## 4.2 Modeling the Query Pattern

First dividing $Q$ into $K$ segments, then estimating the parameters as follows. For transition matrix $A$, we set $A_{i,i+1} = 1$, and $A_{i,j} = 0$ if $j \neq i+1$ except $A_{K,0} = 1$. Given scaling factor $l$, the state duration distribution $P(d_i)$ is defined as following distribution:

$$P(d_i) \propto \begin{cases} \dfrac{1}{\left(\left\lceil d_i^{query} \cdot l \right\rceil - \left\lfloor d_i^{query} / l \right\rfloor\right)}, & \left\lfloor d_i^{query} / l \right\rfloor \leq d_i \leq \left\lceil d_i^{query} \cdot l \right\rceil \\ 0 & , \quad \text{otherwise} \end{cases} \ . \tag{7}$$

where $d_i^{query}$ is the length of $Q$'s $i$th segment. The offset distribution is set to be the form as Equation (2), and its parameters $\mu_{R_i}$ and $\sigma_{R_i}^2$ are set to be $R_i^{query}$ and $0.2R_i^{query}$ respectively, where $R_i^{query}$ is the offset of the $i$th segment of $Q$. For the amplitude difference distribution, we usually use Equation (3) to model the pattern with $\mu_{y_i}$ being $\Delta Y_i^{query}$ and $\sigma_{y_i}^2$ being $0.2\Delta Y_i^{query}$, where $\Delta Y_i^{query}$ is the amplitude difference of the $i$th segment of the query pattern.

### 4.3 Online Pattern Detention

Now we can apply the improved segment semi-Markov model to detect arbitrary scaling similar patterns. Let us call the detection algorithm ISSMM for short. According to the model, for $D$ at each time $t$, the algorithm first calculates the quantity $\hat{p}_i^{(t)}$ that represents the probability of the data belonging to each state $i$, $1 \le i \le K$. The recursive function for calculating $\hat{p}_i^{(t)}$ is

$$\hat{p}_i^{(t)} = \max_j \left( \max_{d_i} \left[ \hat{p}_j^{t-d_i} A_{ji} \right] P(d_i \mid \theta_{d_i}) P(y_t - y_{t-d_i+1} \mid \theta_{y_i}) P(R_i \mid \theta_{R_i}) \right) . \tag{8}$$

Then computes the probability of pre-pattern state and finally normalize the results. The state $i$ and the time $t-d_i$ for the maximum value $\hat{p}_i^{(t)}$ are recorded in $PREV(i,t)$, then we can trace back from $PREV(i,t)$ through the table $PREV$ to get the most likely state sequence. Fig. 3(left) summarizes this procedure in pseudo-code. The algorithm chooses the state with maximum value as the state of the data. If the state is $K$, we declare that one similar pattern has been found, as shown in Fig. 3(right).

Note that it takes constant time to calculate $P(d_i \mid \theta_{d_i})$ and $P(y_t - y_{t-d_i+1} \mid \theta_{y_i})$; and in order to calculate $P(R_i \mid \theta_{R_i})$, it must take $O(d_i)$ time to calculate the offset $R_i$ first. So according to Equation (8) we can deduce that the time complexity of ISSMM is $O\left(|D| \cdot |K| \cdot |Q|^2 \cdot (l^2 - 1/l^2)\right)$, which is lower than other methods mentioned in Section 2 when $|Q| >> |K|$.

## 5 Experiment Results

In this section, we perform our experiments on two real data sets (available from http://www.cs.ucr.edu/~eamonn), which are normalized with mean being 0. Both ISSMM and SWM_LB are used to detect the similar patterns in the same time series.

*1. Results on Motion Capture data set*

The Motion Capture data set was distilled from several hours of recording with Vicon (an optical motion capture system), using 124 sensors [2]. We randomly select a sequence from the data set to use as the query pattern, and then randomly choose 10 other similar sequences and 10 dissimilar sequences to form a time series acting as $D$, see Fig. 4. The scaling factor $l$ is set to 1.2. Table 2 shows the comparative results.

function $s_1 s_2 \dots s_t = AMLSS(y_1 y_2 \dots y_t)$
1. for each state $i$ $(1 \le i \le K)$
2. Compute $\hat{p}_i^{(t)}$, $PREV(i,t)$;
3. $\hat{p}_i^{(t)} = \frac{\hat{p}_i^{(t)}}{p_i^{query}}$
4. end for
5. $\hat{p}_0^{(t)} = 1 - \sum_1^K \hat{p}_i^{(t)}$;
6. normalize $\hat{p}_i^{(t)}$
7. $j = \arg\max_i(\hat{p}_i^{(t)})$;
8. return;

procedure $DETECT(y_1 y_2 \dots y_t \dots)$
1. 1. $t = 1$;
2. $s_1 s_2 \dots s_t = AMLSS(y_1 y_2 \dots y_t)$;
3. if $(s_t == K)$
4. declare 'found';
5. else
6. $t = t + 1$;
7. goto 2;
8. end if

**Fig. 3.** Pseudo-code for ISSMM algorithm. (left) Pseudo-code for AMLSS(finding the most likely state sequence $s_1 s_2 \dots s_t$ for data sequence $y_1 y_2 \dots y_t$). (right) Pseudo-code for DETECT.

Because the similar patterns differ from the other patterns considerably, the lower bounding technique helps a lot to save running time for SWM_LB, and the pruning power $P$ reaches to 0.827. Nevertheless, ISSMM still beats SWM_LB in terms of running time, though the precision of SWM_LB is as good as that of ISSMM.

*2. Results on Gun Problem data set*

The Gun Problem data set comes from the video surveillance domain. The data set has two classes, and all instances were created using one female actor and one male actor in a single session. The two classes are Gun-Draw and Point, as shown in Fig. 5.

We conduct our first experiment on this data set as follows. We randomly select a sequence from the Gun-Draw class to use as the query pattern, see Fig. 6(top). From Fig. 5, we see the amplitude difference of different actors may differ a lot in the 2nd and 8th segments. So we use the following uniform distribution instead of the Gaussian distribution to model the query pattern for these two segments (see Fig. 6(top)):

$$P(\Delta Y_i) \propto \begin{cases} \frac{1}{2\Delta Y_i^{query} - 0.5\Delta Y_i^{query}}, & 0.5\Delta Y_i^{query} \le \Delta Y_i \le 2\Delta Y_i^{query} \\ 0, & \text{otherwise} \end{cases} . \tag{9}$$

Then we randomly choose 10 other Gun-Draw and 10 Point sequences, performed by the same actor performing the query pattern, to form a long time series acting as $D$, see Fig. 6(middle). The scaling factor $l$ is set to 2.5. Table 3 shows the results.

The overall motions of both classes differ subtly, so the lower bounding technique is less efficient, and the pruning power $P$ is only 0.085. ISSMM beats SWM_LB in both precision and speed.

We conduct our second experiment on the data set as follows. We use the same query pattern as the one used in the first experiment. However, we randomly pick out 10 other Gun-Draw sequences performed by both actors, where half by each, and

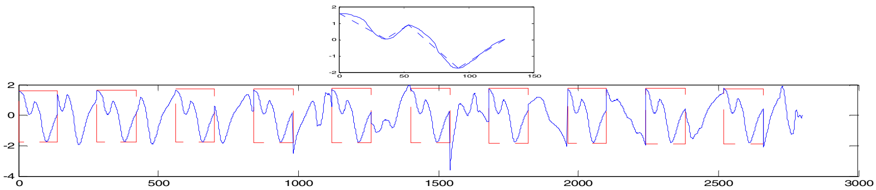

**Fig. 4.** The experiment data on Motion Capture data set. (top) The query pattern represented by the solid curve is divided into 4 linear segments. (bottom) The time series to be detected, and the occurrences of the similar patterns are tagged by the dashed rectangles.

**Table 2.** The experiment results on Motion Capture data set

| | Fault detection rate | Missing detection rate | Running time (second) |
|---|---|---|---|
| ISSMM | 0 | 0 | 343.33 |
| SWM_LB | 0 | 0 | 428.94 |

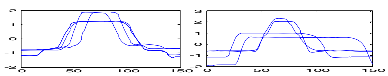

**Fig. 5.** (left)Some examples from Gun-Draw data. (right)Some examples from Point data.

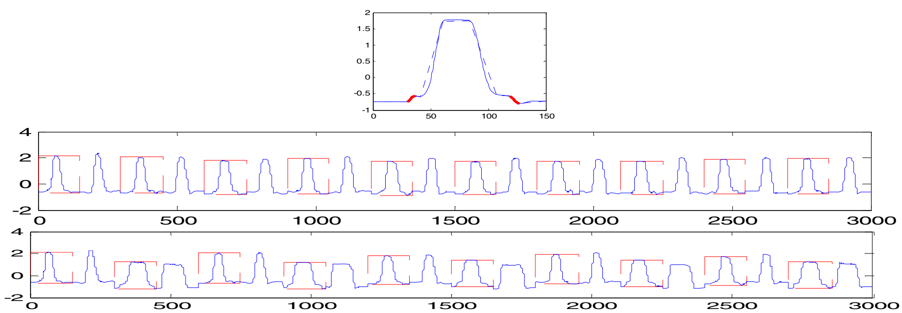

**Fig. 6.** The experiment data on Gun Problem data set. (top) The query pattern represented by the solid curve is divided into 9 linear segments. Especially the 2nd and 8th segments are marked in bold line. (middle) The time series to be detected in the first experiment, and the occurrences of the similar patterns are tagged by the dashed rectangles. (bottom) The time series to be detected in the second experiment, and the occurrences of the similar patterns are tagged by the dashed rectangles.

**Table 3.** The first experiment results on Gun Problem data set

| | Fault detection rate | Missing detection rate | Running time (second) |
|---|---|---|---|
| ISSMM | 0 | 20% | 2417.8 |
| SWM_LB | 33.3% | 20% | 27992 |

**Table 4.** The second experiment results on Gun Problem data set

| | Fault detection rate | Missing detection rate | Running time (second) |
|---|---|---|---|
| ISSMM | 9% | 0 | 2042.4 |
| SWM_LB | 25% | 70% | 16461 |

similarly we pick out 10 Point sequences. Then we concatenate these 20 sequences to form a long time series acting as $D$, see Fig. 6(bottom). The scaling factor $l$ is also set to 2.5. Table 4 shows the comparative results.

Owing to arbitrary time scaling, the amplitudes of Gun-Draw patterns performed by different actors differ sharply after normalization, but their shapes are similar. SWM_LB miss all the Gun-Draw patterns performed by the other actor who is different from the one performing the query pattern, while ISSMM can detect all of them.

## 6 Conclusion and Future Work

In this paper, based on the existing segmental semi-Markov model, we modify it in several aspects. The improved model is applied to online detect arbitrary scaling similar patterns. And it is successfully demonstrated on real data sets.

In future work, we will consider using the model in the noisier environment to widen the application scope of the model.

## References

1. Meek, C., Birmingham, W.: The Dangers of Parsimony in Query-By-Humming Applications. In Proc. of International Symposium on Music Information Retrieval. Baltimore, USA (2003) 51–56
2. Keogh, E., Palpanas, T., Zordan, V.B., Gunopulos, D., Cardle, M.: Indexing Large Human-Motion Databases. In Proc. of 30th International Conference on Very Large Data Bases. Toronto, Canada (2004) 780–791
3. Mandelbrot, B.: Fractals and Scaling in Finance: Discontinuity, Concentration, Risk. New York :Springer-Verlag, (2000)
4. Fu, A.W., Keogh, E., Lau, L.Y.H., Ratanamahatana, C.A.: Scaling and Time Warping in Time Series Querying. In Proc. of the 31st VLDB. Trondheim, (2005) 649-660
5. Ferguson, J.D.: Variable Duration Models for Speech. In Proc. Symposium on the Application of Hidden Markov Models to Text and Speech. (1980) 143-179
6. Ostendorf, M., Digalakis, V.V., Kimball, O.A.: From HMM's to Segment Models: a Unified View of Stochastic Modeling for Speech Recognition. IEEE Transactions on Speech and Audio Processing. (1996) 4(5):360-378
7. Ge, X.P., Smyth, P.: Deformable Markov Model Templates for Time-Series Pattern Matching. In Proc. of the 6th ACM SIGKDD International Conference on Knowledge Discover and Data Mining. Boston, USA (2000) 81-90
8. Ge, X.P., Smyth, P.: Hidden Markov Models for Endpooint Detection in Plasma Etch Processes. Technical Report, Dep. Of ICS, UCI. Available from http://citeser.nj.nec.com/
9. Jia, S. Qian, Y.T., Dai, G.: An Advance Segmental Semi-markov Model Based Online Series Pattern Detection. In Proc. of the 17th International Conference on Pattern Recognition. Vol.3. Cambridge, UK. (2004) 634–637

10. Argyros, T., Ermopoulos, C.: Efficient Subsequence Matching in Time Series Databases under Time and Amplitude Transformations. In Proc. Of 3rd ICDM. USA (2003) 481-484
11. Rabiner, L., Rosenberg, A., Levinson, S.: Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition. IEEE Transactions on Acoustics, Speech and Signal Processing. (1978) 26(6):575-582
12. Baum, L.E., Petrie, T.: Statistical Inference for Probabilistic Functions of Finite State Markov Chains. Annals of Mathematical Statistics. (1966) 37(6):1554-1563