

Efficient Subsequence Search on Streaming Data Based on Time Warping Distance

Sura Rodpongpun¹,
Vit Niennattrakul², and Chotirat Ann Ratanamahatana³, Non-members

ABSTRACT

Many algorithms have been proposed to deal with subsequence similarity search problem in time series data stream. Dynamic Time Warping (DTW), which has been accepted as the best distance measure in time series similarity search, has been used in many research works. SPRING and its variance were proposed to solve such problem by mitigating the complexity of DTW. Unfortunately, these algorithms produce meaningless result since no normalization is taken into account before the distance calculation. Recently, GPUs and FPGAs were used in similarity search supporting subsequence normalization to reduce the computation complexity, but it is still far from practical use. In this work, we propose a novel Meaningful Subsequence Matching (MSM) algorithm which produces meaningful result in subsequence matching by considering global constraint, uniform scaling, and normalization. Our method significantly outperforms the existing algorithms in terms of both computational cost and accuracy.

Keywords: Subsequence Matching, Dynamic Time Warping Distance, Data Stream, Normalization

1. INTRODUCTION

Due to the age of data explosion, analysis of data stream in real time is crucial in many data mining tasks including classification, clustering, anomaly detection, and pattern discovery. Commonly, these tasks require a subsequence matching algorithm as an important subroutine. Recently, SPRING [10], a breakthrough subsequence matching algorithm for data stream under Dynamic Time Warping (DTW) distance [9] has been proposed. SPRING can report an optimal subsequence in linear time. More specifically, it incrementally updates DTW distance, for each new streaming data point, only in time complexity of the query sequence's length. After the proposal of SPRING, many authors [1][7][13] have introduced fast algorithms to improve performance of subsequence matching. In this work, we claim that all

of those past research works [1][10][7][13] are meaningless because the query sequence and candidate sequences from the data stream were not normalized. Normalization [3] is essential to achieve accurate and meaningful distance calculation, as it normalizes the data to have similar offset and distribution, regardless of the distance measure used, especially for DTW distance measure. Unfortunately, as we have mentioned above, current subsequence matching algorithms concern mostly about speed enhancement, but neither on accuracy nor meaningfulness. Fig.1 illustrates subsequence searching in ECG data [3]. Many subsequences with similar shape to the query are missed by the search without normalization.

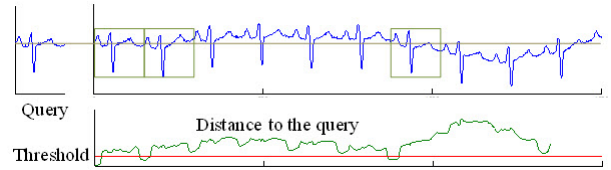


Fig.1: Subsequence searching without normalization in ECG data. Many subsequences with similar shape to the query are left undetected.

However, there is an effort to resolve this problem by trying other approaches; the latest one devises some hardware [11] to accelerate the computation time. The authors propose two techniques, i.e. GPUs and FPGAs, to speed up subsequence matching using DTW with normalization. They have shown that GPUs and FPGAs can help speed up the search significantly. However, it is not practical in real world problems; implementation is hardware dependent, and some systems are not flexibly adjusted to the problem.

We introduce a novel subsequence matching algorithm called MSM (Meaningful Subsequence Matching) for data stream under DTW distance. MSM consists of two new ideas. First, we introduce a multi-resolution lower bound, LB_GUN (Lower-Bounding distance function under Global constraint, Uniform scaling, and Normalization) combining with the well-known LB_Keogh [5] lower-bounding function. LB_GUN is a new lower-bounding distance function extended from LB_Keogh. Second, SSM (Scaling Subsequence Matrix) is used for lower-bounding dis-

Manuscript received on April 1, 2011.

^{1,2,3} The authors are with Department of Computer Engineering, Chulalongkorn University Phayathai Rd., Pathumwan, Bangkok 10330 Thailand, E-mail: g53srd@cp.eng.chula.ac.th, g49vnn@cp.eng.chula.ac.th and ann@cp.eng.chula.ac.th

tance estimation of LB_GUN by incrementally estimating value of normalized data point while guaranteeing no false dismissals. The distances for every scaled query sequence are stored in SSM, and then MSM algorithm monitors SSM to report the optimal range query or the optimal top-k query when a new streaming data point is received. From these two ideas, MSM can monitor data stream nearly in linear time, and it also achieves much higher accuracy than existing algorithms as we expected.

The remainder of this paper is organized as follows. We provide some essential background in Section 2, and state the problem definitions in Section 3. MSM, our proposed method, is described in Section 4. Experimental results are reported in Section 5, and our work is concluded in Section 6.

2. BACKGROUND

In this section, we provide essential background knowledge of Dynamic Time Warping distance measure, global constraint, lower-bounding function for DTW distance, uniform scaling, and normalization.

2.1 Dynamic Time Warping Distance Measure

Dynamic Time Warping (DTW) distance measure [9] is a well-known shape-based similarity measure for time series data. It uses a dynamic programming technique to find an optimal warping path between two time series. Suppose we have two time series sequences, a sequence X of length n and a sequence Y of length m . The distance is calculated by the following equations.

$$D(X_{1..n}, Y_{1..m}) = d(x_n, y_m) + \min \begin{cases} D(X_{1..n-1}, Y_{1..m-1}) \\ D(X_{1..n}, Y_{1..m-1}) \\ D(X_{1..n-1}, Y_{1..m}) \end{cases} \quad (1)$$

where $D(X_{1..n}, \emptyset) = D(\emptyset, Y_{1..m}) = \infty$, $D(\emptyset, \emptyset)$ and \emptyset is an empty sequence. Any distance metric can be used for $d(x_i, y_j)$, including L_1 -norm, i.e., $d(x_i, y_j) = |x_i - y_j|$.

2.2 Global Constraint

Global constraint efficiently limits the optimal path to give a more suitable alignment. Recently, an R-K band [8], a general model of global constraints, has been proposed. R-K band represents a global constraint by a one-dimensional array R , i.e., $R = \langle r_1, r_2, \dots, r_i, r_n \rangle$, where n is the length of time series, and r_i is the height above the diagonal in y -axis and the width to the right of the diagonal in x -axis. Each r_i value is arbitrary, making the R-K band an arbitrary-shaped global constraint.

2.3 Lower-bounding Function for DTW Distance

Although DTW outperforms many other distance measures, it is known to require huge computational complexity. Therefore, LB_Keogh has been proposed to speed up similarity search. $LB_{Keogh}(Q, C)$ between the query sequence $Q = \langle q_1, q_2, \dots, q_i, \dots, q_n \rangle$ and a candidate sequence $C = \langle c_1, c_2, \dots, c_i, \dots, c_n \rangle$ can be computed as follows

$$LB_{Keogh}(Q, C) = \sum_{i=1}^n \begin{cases} |c_i - u_i| & ; \text{if } c_i > u_i \\ |l_i - c_i| & ; \text{if } c_i < l_i \\ 0 & ; \text{otherwise} \end{cases} \quad (2)$$

where $u_i = \max\{q_{i-r_i}, \dots, q_{i+r_i}\}$ and $l_i = \min\{q_{i-r_i}, \dots, q_{i+r_i}\}$ are envelope elements calculated from a global constraint $R = \langle r_1, r_2, \dots, r_i, r_n \rangle$.

2.4 Uniform Scaling

Many research works [2][12] have been shown that when the uniform scaling technique is applied, performance, especially the accuracy, significantly increases. More specifically, uniform scaling technique shrinks/stretches a time series sequence $X = \langle x_1, x_2, \dots, x_i, x_n \rangle$ to a new time series sequence $Y = \langle y_1, y_2, \dots, y_i, y_n \rangle$, where $y_j = x_{[j \cdot n/m]}$. We also define a scaling factor f as a ratio between length m of new time series Y and length n of original time series X or $f = m/n$, and define a scaling range $[f_{min}, f_{max}]$, where f_{min} and f_{max} are minimum and maximum scaling factors which give lengths n_{min} and n_{max} , respectively.

2.5 Normalization

The two time series sequences are compared using any similarity measure; all the data should first be normalized. Z-normalization [3] has been proposed and widely used in time series data mining community, making mean and standard deviation values of the new time series sequence to be zero and one, respectively. Suppose we normalize time series sequence $X = \langle x_1, \dots, x_i, \dots, x_n \rangle$ to sequence $Y = \langle y_1, \dots, y_i, \dots, y_n \rangle$, we can simply formulate transformation function as $y_i = (x_i - \mu_X) / \sigma_X$, where μ_X and σ_X are the mean and standard deviation of time series sequence X , respectively.

3. PROBLEM DEFINITION

In this paper, we focus on two main query problems on streaming time series data, i.e., optimal range query and optimal top- k query. The objective of the optimal range query is to find non-overlapping normalized subsequences from a data stream, whose distance between a candidate sequence and a query sequence must be less than a threshold, where the query sequence is scaled and normalized under uniform scaling between scaling range $[f_{min}, f_{max}]$. On

the other hand, optimal top- k query reports top- k non-overlapping normalized subsequences. Nevertheless, the scaled query sequences and all candidate subsequences in the data stream must be normalized in order to return meaningful results. A naïve method to monitor incoming data stream first initializes a set of normalized scaled query sequences, and then candidate sequences are extracted from the data stream using sliding-window model. After normalization, distance calculation is performed on the extracted subsequences and non-overlapping optimal results are reported (if any). However, this naïve method requires as high as $O(n^3)$ time complexity for each new incoming streaming data point.

4. PROPOSED METHOD

Since the naïve method consumes too high time complexity, we propose a novel approach for subsequence matching which gives meaningful result. We call our proposed method as an MSM algorithm (Meaningful Subsequence Matching), which contains two new ideas, i.e., a multi-resolution lower-bounding function LB_GUN (Lower-Bounding function under Global constraint, Uniform scaling, and Normalization), and SSM (Scaling Subsequence Matrix) which incrementally estimates value of LB_GUN under global constraint, uniform scaling, and normalization in linear time while guaranteeing no false dismissals. Three following subsections of LB_GUN, SSM, and MSM algorithm are precisely described.

4.1 Lower-Bounding Distance under Global Constraint, Uniform Scaling, and Normalization (LB_GUN)

LB_GUN is a lower-bounding function of DTW distance extended from LB_Keogh [5] whose distance calculation can be done in linear time. Before calculation, LB_GUN first creates an envelope E' from scaled and normalized envelopes. More specifically, three sequence sets are generated, i.e., sets of Q^\sim , R^\sim , and E^\sim . The scaled query set $Q^\sim = \{Q'_{n_{min}}, \dots, Q'_k, \dots, Q'_{n_{max}}\}$ is first generated by scaling and normalizing a query sequence Q to every normalized scaled query sequence, and the scaled global constraint $R^\sim = \{R'_{n_{min}}, \dots, R'_k, \dots, R'_{n_{max}}\}$ set is derived from scaling a specific global constraint set R^\sim with all possible scaling lengths from n_{min} to n_{max} . An envelope E'_k of a normalized scaled query sequence Q'_k and a scaled global constraint R'_k for sequence length k is created as in LB_Keogh, and is stored in the envelope set $E^\sim = \{E'_{n_{min}}, \dots, E'_k, \dots, E'_{n_{max}}\}$. Then, E' is generated by merging all envelopes in the set E^\sim together, where $E' = \{\langle u'_1, l'_1 \rangle, \dots, \langle u'_i, l'_i \rangle, \dots, \langle u'_{n_{max}}, l'_{n_{max}} \rangle\}$. To find lower-bounding distance between a query sequence Q and a candidate sequence C under global constraint, uniform scaling, and normalization, an envelope E'

of a query sequence Q is generated as mention above. $LB_{GUN}(Q, C, n)$ is shown in Equation (3).

$$LB_{GUN}(Q, C, n) = \frac{1}{\sigma_{C_{1\dots n}}} \left(\sum_{i=1}^n \alpha_i + \mu_{C_{1\dots n}} \sum_{i=1}^n \beta_i \right) + \sum_{i=1}^n \gamma_i \quad (3)$$

$$\alpha_i = \begin{cases} c_i & ; \text{if } c'_i \geq u'_i \\ -c_i & ; \text{if } c'_i \leq l'_i \\ 0 & ; \text{otherwise} \end{cases} \quad (4)$$

$$\beta_i = \begin{cases} -1 & ; \text{if } c'_i \geq u'_i \\ 1 & ; \text{if } c'_i \leq l'_i \\ 0 & ; \text{otherwise} \end{cases} \quad (5)$$

$$\gamma_i = \begin{cases} -u'_i & ; \text{if } c'_i \geq u'_i \\ l'_i & ; \text{if } c'_i \leq l'_i \\ 0 & ; \text{otherwise} \end{cases} \quad (6)$$

where $\mu_{C_{1\dots n}}$ and $\sigma_{C_{1\dots n}}$ are arithmetic mean and standard deviation of data points 1 to n of a candidate sequence C , $c'_i = (c_i - \mu_{C_{1\dots n}}) / \sigma_{C_{1\dots n}}$, n_{min} and n_{max} are desired scaling lengths, and $n_{nim} \leq n \leq n_{max}$.

4.2 Scaling Subsequence Matrix

SSM (Scaling Subsequence Matrix) is another important component in MSM algorithm. It stores lower-bounding distances determined by LB_GUN for each new incoming streaming data point s_t at time t from data stream S . Suppose we have a query sequence Q ; each element $\langle t, j \rangle$ of the matrix contains five values, i.e., $v_{t,j}$, $w_{t,j}$, $x_{t,j}$, $y_{t,j}$, and $z_{t,j}$, calculated from time $t-j$ to time t . Therefore, values in matrix element $\langle t, j \rangle$ can be incrementally updated from the matrix element $\langle t-1, j-1 \rangle$ according to the following equations.

$$v_{i,j} = v_{t-1,j-1} + \begin{cases} s_t & ; \text{if } s'_i \geq u'_i \\ -s_t & ; \text{if } s'_i \leq l'_i \\ 0 & ; \text{otherwise} \end{cases} \quad (7)$$

$$w_{i,j} = w_{t-1,j-1} + \begin{cases} -1 & ; \text{if } s'_i \geq u'_i \\ 1 & ; \text{if } s'_i \leq l'_i \\ 0 & ; \text{otherwise} \end{cases} \quad (8)$$

$$x_{i,j} = x_{t-1,j-1} + \begin{cases} -u'_i & ; \text{if } s'_i \geq u'_i \\ l'_i & ; \text{if } s'_i \leq l'_i \\ 0 & ; \text{otherwise} \end{cases} \quad (9)$$

$$y_{t-j} = y_{t-1,j-1} + s_t \quad (10)$$

$$Z_{t,j} = z_{t-1,j-1} + (s_t)^2 \quad (11)$$

$$lb_{t,j} = \frac{1}{\sigma_{t,j}}(v_{t,j} + \mu_{t,j} \cdot w_{t,j}) + x_{t,j} \quad (12)$$

where $s'_t = \frac{s_t - \mu_{t,j}}{\sigma_{t,j}}$, $\mu_{t,j} = \frac{y_{t,j}}{j}$, $\sigma_{t,j} = \sqrt{\frac{z_{t,j}}{j} - (\mu_{t,j})^2}$, u_j and l_j are from an enveloped E' generated from a query sequence Q , $1 \leq j \leq n_{max}$, $n_{min} \leq j \leq n_{max}$, and $lb_{t,j}$ is a lower-bounding distance LB_GUN for an element $\langle t, j \rangle$.

4.3 Meaningful Subsequence Matching

Since SSM is updated at every arrival of new streaming data point s_t , our MSM algorithm can monitor both optimal range query and optimal top- k query. More specifically, for optimal range query, MSM first calculates and updates values including lower-bounding distances in SSM, which is an estimation of LB_GUN and then checks whether best-so-far distance d_{best} is smaller than threshold ε . If so, MSM reports an optimal subsequence when there is no overlapping subsequence, and MSM resets d_{best} and values in SSM. For all $lb_{t,j}$ which are smaller than d_{best} in range from n_{min} to n_{max} , LB_GUN and LB_Keogh are calculated and compared to d_{best} respectively to prune off the DTW distance calculation. If they are not pruned by any lower-bounding distances, DTW distance is computed to update d_{best} and the optimal subsequence's position. Additionally, MSM uses only two columns of SSM that are values in time t and values in time $t-1$. All lower-bounding distances and DTW distance are normalized by dividing by i . The MSM algorithm for optimal range query is described in Table 1.

Table 1: MSM Algorithm for optimal range query

MSMOPTIMALRANGEQUERY ALGORITHM	
1	Input: a new streaming data point s_t
2	Output: an optimal subsequence (if any)
3	update v_i, w_i, x_i, y_i and z_i for all i , $1 \leq i \leq n_{max}$ and lb_i for all i , $n_{min} \leq i \leq n_{max}$
4	if ($d_{best} < \varepsilon$ and $\forall i, t_{best}^{end} \leq t - i$)
5	REPORT($d_{best}, S[t_{best}^{end}, t_{best}^{start}]$)
6	$d_{best} = \infty$
7	reset v_i, w_i, x_i, y_i and $z_i = \infty$ for all i , $t_{best}^{end} > t - i$
8	for ($i = n_{min}$ to n_{max})
9	if ($lb_i \leq d_{best}$)
	if ($LB_GUN(Q'_i, NORMALIZE(S[t - i : t])) < d_{best}$)
	if ($LB_KEOGH(Q'_i, NORMALIZE(S[t - i : t])) < d_{best}$)
10	distance = DTW($Q'_i, NORMALIZE(S[t - i : t])$)
11	if ($distance \leq d_{best}$)
12	$d_{best} = distance$; $t_{best}^{end} = t - i$; $t_{best}^{start} = t$
13	substitute v_i', w_i', x_i', y_i' and z_i' for v_i, w_i, x_i, y_i and z_i

Table 2: MSM Algorithm for optimal top- k query

MSMOPTIMALTOPKQUERY ALGORITHM	
1	Input: a new streaming data point s_t
2	Output: update set P of top- k subsequence
3	$[C, d_C] = \text{MSMOPTIMALRANGEQUERY}(s_t, \varepsilon)$
4	If($C \neq \text{NULL}$)
5	$P.push(C, d_C)$
6	if ($size(P) > k$)
7	$P.pop()$
8	$\varepsilon = P.peek().d_C$

MSM algorithm for optimal top- k query is implemented based on the optimal range query. With a priority queue, MSM stores the k -best non-overlapping subsequence with DTW distance from the result of MSMOPTIMALRANGEQUERY. First, we initialize a threshold ε to positive infinity. Then, for every new streaming data point s_t , the queue is updated, and the threshold ε is set to the largest DTW distance in the queue. The MSM algorithm for optimal top- k query is described in Table 2.

5. EXPERIMENTAL EVALUATION

Since none of the current subsequence matching algorithms under DTW distance can handle the changes of data distribution, offset, and scaling, we compare our proposed method with naïve approach in terms of computational time only since our proposed method and the naïve method will both achieve the same accuracy. On the other hand, we compare our accuracy with SPRING, the best existing subsequence matching under DTW distance. Note that we do not compare our running time with that of SPRING; while SPRING will have smaller running time, its results are inaccurate due to lack of normalization, therefore is not a reasonable comparison.

Streaming datasets are generated by combining training data sequences from the UCR classification/clustering datasets [6] and synthesized random walk sequences. A stream is initialized with a random walk sequence, and then a training data sequence is appended to the stream. To smooth the stream, before concatenation, each sequence is offset by the last value of the stream. The dataset we used in the experiments are Aidac, Beef, CBF, Coffee, ECG200, Gun Point, Lighting7, Olive Oil, Trace and Synthetic Control which are represented by Data 1, Data 2, Data 3, Data 4, Data 5, Data 6, Data 7, Data 8, Data 9 and Data 10, respectively.

In the first experiment, we compare our MSM algorithm with naïve method in terms of computational cost by measuring the number of distance calculations. Fig.2 shows the numbers of all distance calculations by varying global constraint to 2, 4, 6, 8 and 10 respectively, and in Fig.3, scaling range $[f_{min}, f_{max}]$ are varied from $[0.8, 1.2]$, $[0.85, 1.15]$, $[0.9, 1.1]$ and $[0.95, 1.05]$ respectively. The numbers of all distance

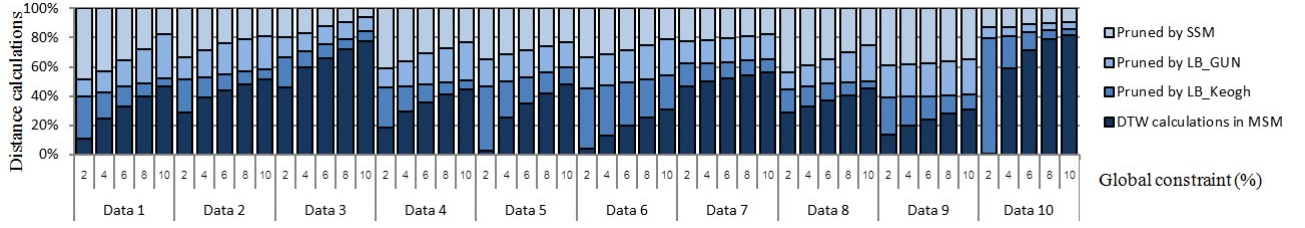


Fig.2: DTW distance calculations filtered out by MSM with varying global constraints

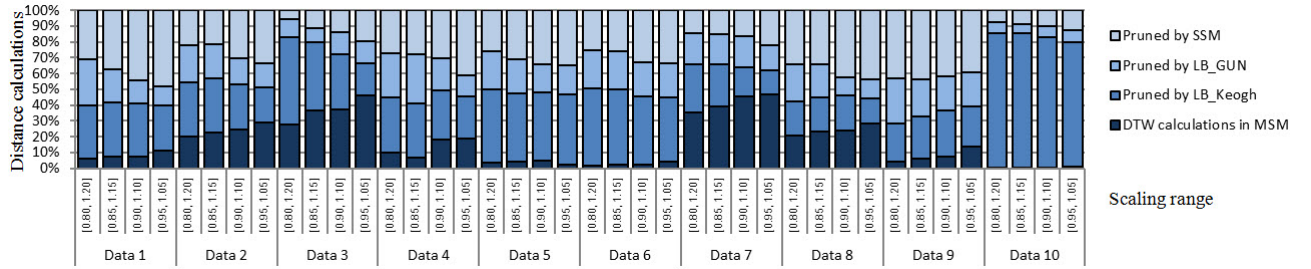


Fig.3: DTW distance calculations filtered out by MSM with varying scaling ranges

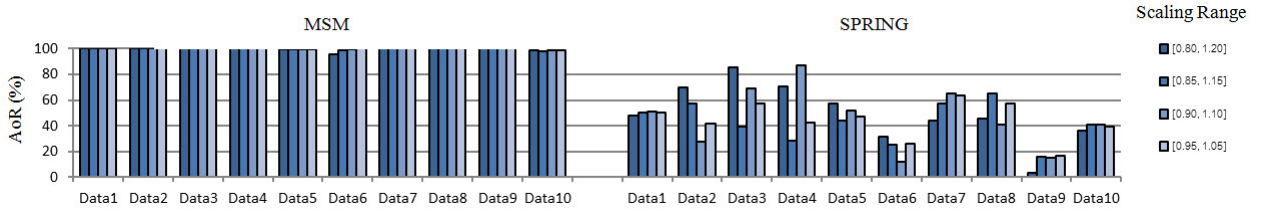


Fig.4: MSM outperforms SPRING every scaling range in terms of AoR

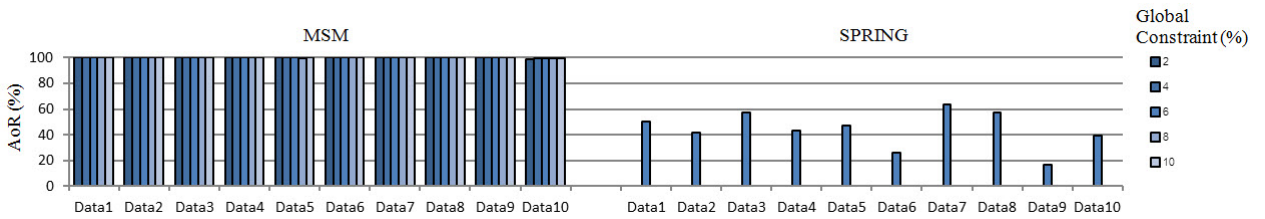


Fig.5: MSM outperforms SPRING every global constraint value in terms of AoR

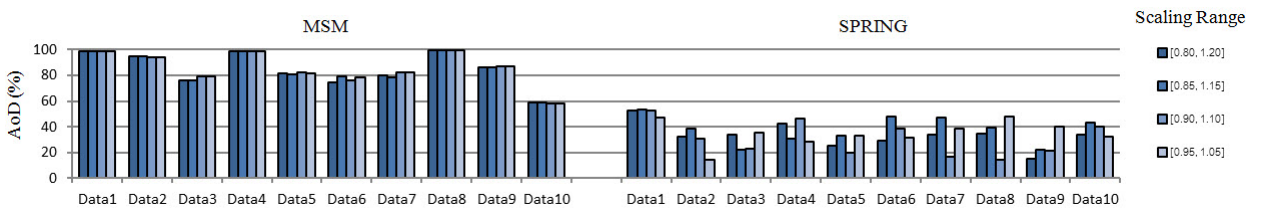


Fig.6: MSM outperforms SPRING every scaling range in terms of AoD

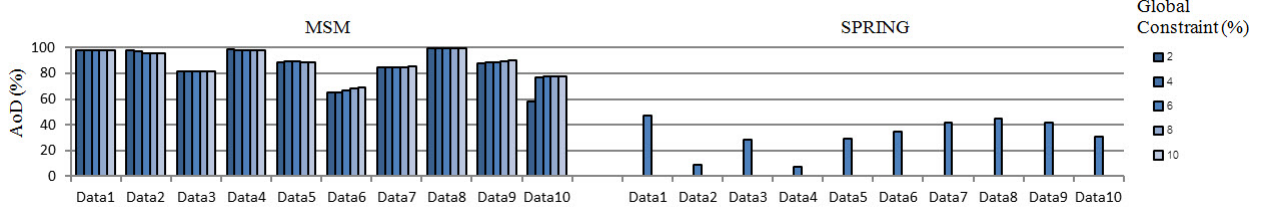


Fig. 7: MSM outperforms SPRING every global constraint value in terms of AoD

calculations are normalized to 100% which represent numbers of DTW calculations used in the Naïve method. As expected, MSM is much faster than the naïve method by a large margin. Additionally, in MSM, our multi-resolution lower-bounding function is efficiently used to filter out several candidate sequences in linear time while guaranteeing no false dismissals; therefore, MSM algorithm requires only a small number of DTW distance calculations comparing with the naïve method.

Then, we compare our MSM algorithm with SPRING to measure performance in terms of accuracy, both Accuracy-on-Retrieval (AoR) and Accuracy-on-Detection (AoD). AoR reflects quality of an algorithm that is able to find the patterns in data stream; on the other hand, AoD reflects quality of the returned results. Suppose we have data stream S , a set of expected pattern sequences E , and a set of retrieved sequences R . We first define an overlapping subsequence. Let $S[ts:te]$ be the subsequence starting at t_s and ending at t_e . Overlapping subsequence $O_{X,Y}$, where $X = S[a : b]$ and $Y = S[c : d]$, and overlap percentage $P_{X,Y}$ are defined as $O_{X,Y} = S[\min\{a, c\} : \min\{b, d\}]$ and $P_{X,Y} = \frac{|O_{X,Y}|}{\max\{b, d\} - \min\{a, c\} + 1}$, respectively. Both AoR and AoD can be defined over overlapping subsequence $O_{X,Y}$ and overlapping percentage $P_{X,Y}$ as $AoR = \frac{|\{O_{X,Y} | P_{X,Y} > p, X \in R, Y \in E\}|}{|E|}$ and $AoD = \frac{\sum \{P_{X,Y} | P_{X,Y} > p, X \in R, Y \in E\}}{|\{O_{X,Y} | P_{X,Y} > p, X \in R, Y \in E\}|}$, respectively,

where p is a threshold of $P_{X,Y}$ that defines a sequence in R as a discovered sequence. Fig.4 and Fig.5 compare AoRs of MSM and SPRING under various scaling ranges and global constraints, respectively. Fig.6 and Fig.7 illustrate AoDs on every scaling range and global constraint, respectively. The results show that MSM produces more meaningful result since SPRING does not support global constraint (illustrated as one single column of 100% global constraint in Fig.5 and Fig.7), uniform scaling, nor normalization.

6. CONCLUSION

This paper proposes a novel and meaningful subsequence matching algorithm, so called MSM (Mean-

ingful Subsequence Matching), under global constraint, uniform scaling, and normalization. Two ideas have been introduced in MSM algorithm, i.e., a multi-resolution lower-bounding function LB.GUN (Lower-Bounding distance function under Global constraint, Uniform scaling, and Normalization, and a Scaling Subsequence Matrix (SSM) which estimates value of LB.GUN for each candidate subsequence. Our algorithm can update lower-bounding distance incrementally under normalization, while guaranteeing no false dismissals in linear time. With these two ideas, MSM algorithm can efficiently monitor data stream and can answer both optimal range query and optimal top- k query problems. Since none of the current algorithm produces meaningful result, we evaluate our proposed method comparing with the naïve method in terms of time consumption and SPRING, the best existing subsequence matching under DTW distance, in terms of accuracies. As expected, our MSM algorithm is much faster and more accurate by a very large margin.

7. ACKNOWLEDGEMENT

This research is partially supported by the Thailand Research Fund (Grant No. MRG5380130), the Thailand Research Fund given through the Royal Golden Jubilee Ph.D. Program (PHD/0319/2551 to S. Rodpongpun and PHD/0141/2549 to V. Niennatrakul).

References

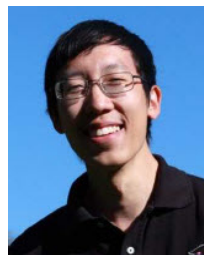
- [1] V. Athitsos, P. Papapetrou, M. Potamias, G. Kollios, and D. Gunopulos, "Approximate Embedding-based Subsequence Matching of Time Series," *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD'08)*, pp. 365-378, 2008.
- [2] A. W. C. Fu, E. J. Keogh, L. Y. H. Lau, C. A. Ratanamahatana, and R. C. W. Wong, "Scaling and Time Warping in Time Series Querying," *The VLDB Journal*, vol. 17, pp. 899-921, 2008.
- [3] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley, *PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Re-*

source for *Complex Physiologic Signals*, Circulation 101(23):e215-e220, 2000.

- [4] J. Han and M. Kamber, *Data Mining: Concepts and Techniques: Morgan Kaufmann*, 2001.
- [5] E. J. Keogh and C. A. Ratanamahatana, "Exact Indexing of Dynamic Time Warping," *Knowledge and Information Systems*, vol. 7, pp. 358-386, 2005.
- [6] E. J. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana, "UCR Time Series Classification/Clustering Page," http://www.cs.ucr.edu/~eamonn/time_series_data.
- [7] V. Niennattrakul, D. Wanichsan, and C. A. Ratanamahatana, "Accurate Subsequence Matching on Data Stream under Time Warping," *PAKDD 2009 International Workshops (PAKDD'09)*, pp. 156-167, 2010.
- [8] C. A. Ratanamahatana and E. J. Keogh, "Making Time-Series Classification More Accurate Using Learned Constraints," *Proceedings of the 4th SIAM International Conference on Data Mining (SDM'04)*, pp. 11-22, 2004.
- [9] C. A. Ratanamahatana and E. J. Keogh, "Three Myths about Dynamic Time Warping Data Mining," *Proceedings of the 5th SIAM International Conference on Data Mining (SDM'05)*, pp. 506-510, 2005.
- [10] Y. Sakurai, C. Faloutsos, and M. Yamamura, "Stream Monitoring under the Time Warping Distance," *Proceedings of the 23th International Conference on Data Engineering (ICDE'07)*, pp. 1046-1055, 2007.
- [11] D. Sart, A. Mueen, W. Najjar, E. Keogh, and V. Niennattrakul, "Accelerating Dynamic Time Warping Subsequence Search with GPUs and FPGAs," *IEEE International Conference on Data Mining (ICDM'10)*, pp. 1001-1006, 2010.
- [12] D. Yankov, E. J. Keogh, J. Medina, B. Chiu, and V. B. Zordan, "Detecting Time Series Motifs under Uniform Scaling," *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07)*, pp. 844-853, 2007.
- [13] P. Zou, L. Su, Y. Jia, W. Han, and S. Yang, "Fast Similarity Matching on Data Stream with Noise," *Proceedings of the 24th International Conference on Data Engineering Workshops (ICDEW'08)*, pp. 194-199, 2008.



Sura Rodpongpan is a Ph.D. student in Computer Engineering at Chulalongkorn University. He received his B.Eng. in Computer Engineering from Mahidol University. He was granted scholarships from the Thailand Research Fund through the Royal Golden Jubilee Ph.D. Program. His research interests are time series data mining, machine learning, and artificial intelligence.



Vit Niennattrakul received his B.Eng. and Ph.D. in Computer Engineering from Chulalongkorn University in 2006 and 2011, respectively. He was granted scholarships from the Thailand Research Fund through the Royal Golden Jubilee Ph.D. Program from 2007 to 2011, Chulalongkorn University Graduate Scholarship to Commemorate the 72nd Anniversary of His Majesty King Bhumibol Adulyadej, and the 90th Anniversary of Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund). His research interests include time series data mining, machine learning, and natural language processing.



Chotirat Ann Ratanamahatana is an assistant professor in Computer Engineering Department, Chulalongkorn University, Bangkok, Thailand. She received her undergraduate and graduate studies from Carnegie Mellon University and Harvard University, respectively, and received her Ph.D. from the University of California, Riverside. Her research interests include data mining, information retrieval, machine learning, and human-computer interaction.