

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/262312434>

Shape-Based Clustering for Time Series Data

Conference Paper · May 2012

DOI: 10.1007/978-3-642-30217-6_44

CITATIONS

55

READS

7,034

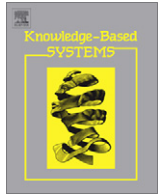
3 authors, including:



[Chotirat Ann Ratanamahatana](#)
Chulalongkorn University

89 PUBLICATIONS 7,171 CITATIONS

SEE PROFILE



Shape-based template matching for time series data

Vit Niennattrakul, Dararat Srisai, Chotirat Ann Ratanamahatana*

Department of Computer Engineering, Chulalongkorn University, 254 Phayathai Road, Pathumwan, Bangkok 10330, Thailand

ARTICLE INFO

Article history:

Received 7 December 2010

Received in revised form 21 April 2011

Accepted 23 April 2011

Available online 29 April 2011

Keywords:

Time series

Template matching

Shape averaging

Dynamic time warping

Classification

ABSTRACT

Dynamic time warping (DTW) distance has been proven to be one of the most accurate distance measures for time series classification. However, its calculation complexity is its own major drawback, especially when a massive training database has to be searched. Although many techniques have been proposed to speed up the search including indexing structures and lower bounding functions, for large databases, it is still untenable to embed the algorithm and search through the entire database of a system with limited resources, e.g., tiny sensors, within a given time. Therefore, a template matching is a solution to efficiently reduce storage and computation requirements; in other words, only a few time series sequences have to be retrieved and compared with an incoming query data. In this work, we propose a novel template matching framework with the use of DTW distance, where a shape-based averaging algorithm is utilized to construct meaningful templates. Our proposed framework demonstrates its utilities, where classification time speedup is in orders of magnitude, while maintaining good accuracy to rival methods.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Time series classification [28,27,18,21] is one of the major problems in time series data mining community [3,12,13,26], where its applications contribute in several domains, e.g., speech recognition [6,22], biometrics [14,11,17], multimedia [29,9], etc. Nearest neighbor classifier with dynamic time warping (DTW) distance measure [18,20] has shown in many recent works to outperform dozen other distance measures by using only a few parameters [4]. This leads to active research in DTW distance measure. However, this DTW distance measure has a well-known drawback, i.e., its computational complexity is quadratic. Since the nearest neighbor classifier requires to search through every data sequence in a database, it is definitely impractical to implement on a system with limited resources in either memory storage or computational power, e.g., a tiny sensor or an embedded system.

Recently, many techniques to speed up nearest neighbor search for DTW distance have been widely proposed including lower bounding distance functions [10,30,7,31,23,16] and index structures [7,23,31,16]. A lower bounding distance function, a much faster calculation, is used to estimate DTW distance between two time series sequences with one simple condition: the lower bounding distance must be smaller than or equal to its actual DTW distance. Unlike the lower bounding distance function, an index

structure has been proposed to guide the search by accessing only portions of the database instead of searching through the entire database. Although these well-known techniques can reduce some computational time, a large number of sequences are still required to be retrieved for the nearest neighbor calculation. In addition, more storage space is required for storing an index structure, while its main objective is aiming to reduce storage space for a system with limited resources.

Template matching is a solution. Instead of searching for a nearest neighbor from an entire database, only a few templates have to be retrieved, and the class label of the best-matched template is returned as an answer for the issued query. Typically, one template for each class is constructed, so the number of data needed to be stored is merely equal to the number of classes. With a template matching framework, a system with limited resources is now practical; in other words, the system can significantly reduce both storage and computation requirements for classification problems.

Generally, to construct a template, all data sequences of the same class are averaged. Unlike other typical data types, time series data need a shape-based averaging algorithm instead of a typical amplitude averaging approach since correlation among adjacent dimensions of time series exists [15,25]. Additionally, amplitude averaging produces an undesired mean, which leads to an inaccurate classification. Fig. 1(b) shows an undesired averaged result containing two events, whereas both original sequences, A and B, consist of only one event. In fact, a good template should preserve characteristics of these two data sequences, i.e., only one event should appear, as shown in Fig. 1(c). This characteristic-preserving template can be achieved by a shape-based averaging method.

* Corresponding author. Tel.: +66 8 9499 9400; fax: +66 2 218 6955.

E-mail addresses: g49vnn@cp.eng.chula.ac.th (V. Niennattrakul), g51dsr@cp.eng.chula.ac.th (D. Srisai), ann@cp.eng.chula.ac.th (C.A. Ratanamahatana).

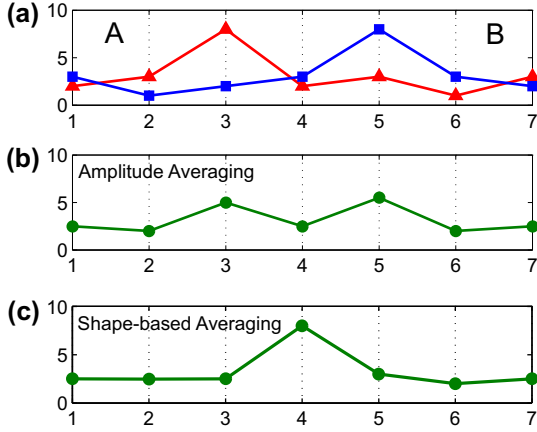


Fig. 1. (a) Two time series sequences *A* and *B* are averaged using (b) an amplitude averaging and (c) a shape-based averaging.

However, finding a shape-based mean is still controversial because data sequences are averaged in a DTW distance space, not in a Euclidean space. Unfortunately, to the best of our knowledge, no optimal solution has yet been proposed. Over a decade ago, Gupta et al. proposed a heuristic solution called NLAFF [5], but only a handful of work has adapted it to a time series data mining domain [19,24]. Particularly, NLAFF does not produce good averaged results, and consequently this leads to poor classification accuracy (this will be demonstrated and compared with our proposed method in Section 5).

In this work, we propose a novel shape-based template matching framework (STMF) for time series data with deterministic heuristic averaging algorithms. STMF consists of two phases, i.e., a training phase, where templates are constructed, and a test phase, where a query sequence is classified with the constructed templates. To construct a template, a new averaging scheme with two averaging functions, cubic-spline dynamic time warping (CDTW) averaging and iterative cubic-spline dynamic time warping (ICDTW) averaging, is introduced in this paper. With these algorithms, very well-formed templates are stored in the database. In the test phase, templates are retrieved and compared with the query sequence, and a class label of the nearest template will be the answer to the query. It is worth to note that classification with templates typically achieves lower accuracy than classification with an entire database. In experimental evaluation section, we will show that our STMF achieves comparable accuracies, while being able to speed up the classification in orders of magnitude.

The rest of the paper is organized as follows. Sections 2 and 3 provide essential background and related work, respectively. Our framework, STMF (shape-based template matching framework), will be introduced along with two averaging algorithms in Section 4. In Section 5, extensive experimental evaluation will be demonstrated. Finally, we offer conclusions and directions for future work in Section 6.

2. Background

This section provides essential background knowledge to understand our proposed methods in this paper.

2.1. Dynamic time warping (DTW) distance

DTW distance [1,20,18] is a well-known shape-based similarity measure. It uses a dynamic programming technique to find an optimal warping path between two time series sequences. To calculate

the distance, it first creates a distance matrix, where each element in the matrix is a cumulative distance of a minimum of three surrounding neighbors. Suppose we have two time series, a sequence $A = \langle a_1, \dots, a_i, \dots, a_n \rangle$ and a sequence $B = \langle b_1, \dots, b_j, \dots, b_m \rangle$. First, we create an n -by- m matrix, and then each (i, j) element, γ_{ij} , of the matrix is defined as:

$$\gamma_{ij} = |a_i - b_j|^p + \min\{\gamma_{i-1,j-1}, \gamma_{i-1,j}, \gamma_{i,j-1}\} \quad (1)$$

where γ_{ij} is the summation of $|a_i - b_j|^p$ and a minimum cumulative distance of three elements surrounding the (i, j) element, and p is the dimension of L_p -norms. When all elements in the matrix are filled, the DTW distance is determined from the last element $\gamma_{n,m}$ of the matrix. For time series domain, $p = 2$, equipping to a Euclidean distance, is typically used. Since the DTW distance is important background knowledge for this paper, we provide more concrete pseudo code in Table 1 and an illustrative example in Fig. 2.

2.2. Dynamic time warping (DTW) averaging

DTW averaging was first introduced by Gupta et al. [5] to find an averaged signal between two time series sequences. Unlike the DTW distance, DTW averaging uses another matrix to store an index of the adjacent element that has a minimum cumulative distance. After elements in the path matrix are filled up, the path is traced back from the last element to the first element. An averaged result is then calculated along the path. Suppose the path $W = \langle w_1, \dots, w_k, \dots, w_N \rangle$, where w_k is the k th coordinate (i_k, j_k) in the optimal path of sequences *A* and *B*, where i_k and j_k are indices of data points in sequences *A* and *B*, respectively. Therefore, a new sequence *C* is derived from elements $c_k = \frac{a_{i_k} \cdot \omega_A + b_{j_k} \cdot \omega_B}{\omega_A + \omega_B}$, where ω_A and ω_B are the weights of the sequences *A* and *B*, respectively. We also

Table 1

Dynamic time warping distance measure.

FUNCTION [dist] = DTW-DISTANCE(<i>A</i> , <i>B</i>)	
1.	Let <i>n</i> be the length of time series <i>A</i>
2.	Let <i>m</i> be the length of time series <i>B</i>
3.	Initialize $D = \text{ARRAY}[n][m]$
4.	For ($i = 1$ to n)
5.	For ($j = 1$ to m)
6.	If ($i = 1$ and $j \neq 1$)
7.	$\min = D_{i,j-1}$
8.	Else if ($i \neq 1$ and $j = 1$)
9.	$\min = D_{i-1,j}$
10.	Else
11.	$\min = \text{MIN}(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1})$
12.	End if
13.	$D_{i,j} = \min + a_i - b_j ^p$
14.	End for
15.	End for
16.	Return $\text{dist} = \sqrt[p]{D_{n,m}}$

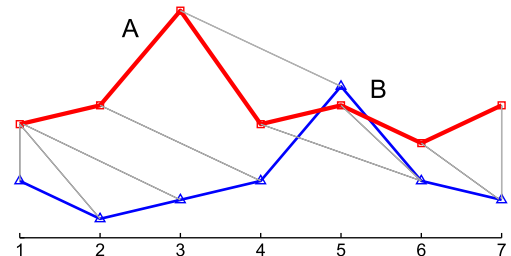


Fig. 2. Mapping between two time series sequences $A = \langle 2, 3, 8, 2, 3, 1, 3 \rangle$ and $B = \langle 3, 1, 2, 3, 8, 3, 2 \rangle$ in DTW distance calculation.

provide a concrete pseudo code and an example of DTW averaging in Table 2 and Fig. 3, respectively.

It is important to note that DTW averaging function is an operation which has only commutative property with no associative property [15]. In the other words, if there are three sequences A, B, and C, a result of averaging A and B, then C is not necessarily equal to a result of averaging B and C, then A. A sequence ordering can largely affect the averaged result. In addition, an averaging sequence will always be longer or equal to the original sequences. If a large dataset is to be averaged, averaging sequences will be very long which will definitely decrease a system's performance. Therefore, in this work, we propose two new DTW averaging functions to resolve this problem and a new averaging scheme to efficiently order averaging sequences.

3. Related work

Over a decade ago, Gupta et al. [5] proposed a heuristic shape-averaging scheme called NLAFF, which was first introduced in signal processing community, and later has been utilized in data mining tasks [19,24]. Specifically, NLAFF uses a DTW averaging to produce a mean between a pair of time series sequences. NLAFF consists of two averaging schemes, i.e., NLAFF₁ and NLAFF₂. NLAFF₁ averages sequences in hierarchical manner. Suppose there are eight sequences, i.e., A₁ to A₈. A₁ and A₂ are averaged to produce A_{1,2}, and A₃ and A₄ are averaged to produce A_{3,4}, and so on. Then, in the next level, A_{1,2} and A_{3,4} are averaged to produce A_{(1,2),(3,4)}, and so on. Limitation of NLAFF₁ is that it requires the number of sequences to be a power of two. Unlike NLAFF₁, NLAFF₂ averages sequences in sequential manner. A₁ and A₂ are first averaged to produce A_{1,2}, and then A_{1,2} and A₃ are averaged to produce A_{(1,2),3}, and so on.

Since NLAFF₁ has a limitation that it requires the number of sequences to be a power of two, Gupta et al. recommend to use a combination of both NLAFF₁ and NLAFF₂. For example, to average 100 sequences, 4 sequences will be discarded, and the rest of the

sequences will be separated into three groups of 32 sequences, each of which will be averaged using NLAFF₁. Therefore, three averaged sequences produced from NLAFF₁ will then be averaged using NLAFF₂. Obviously, NLAFF is nondeterministic. Since a DTW averaging function does not have associative property, different orderings of sequences in both NLAFF₁ and NLAFF₂ will lead to different averaged results. Additionally, an averaged sequence produced by NLAFF will be very long since DTW averaging function will always produce a longer sequence than its original sequences. In this work, we propose two new DTW averaging functions and an averaging scheme which will produce a more accurate averaged result, and when this result is used as a template, it will produce a more accurate classification accuracy.

4. Shape-based template matching framework

Shape-based template matching framework (STMF) utilizes shape-based averaging in creating characteristic-preserving templates using the Dynamic time warping (DTW) distance as a similarity measurement. STMF consists of two phases, i.e., a training phase and a test phase. In a training phase, one template for each class is constructed from an entire raw database, and then templates are stored with their class labels. As the best case, only one template for each class is required; however, the number of templates can be more than one. Note that the overall system's performance including a storage requirement and a computational time will improve as the number of templates increases. Table 3 shows a simple idea of a training phase of STMF, where an entire database is an input, and an output is a set of templates.

In a test phase, only a set of templates is retrieved and compared with a query for a closest match, where a set of templates is very small comparing to the original database. Therefore, classification time of template matching will be much faster than the typical one-nearest-neighbor classifier in many orders of magnitude. However, this classification with template matching has a trade-off that its classification accuracy may decrease since some characteristics of data objects in the database could be lost in the averaging process where some details are dominated by a majority of the data. To be more illustrative, Table 4 shows how to classify an incoming query with the stored templates.

To average a set of sequences, we propose a scheme to compute an averaged result since the shape-based averaging does not have a commutative property [17]. Instead of averaging sequences in a random order as done in NLAFF, we propose a heuristic solution to return a good averaged result by averaging a most similar pair of sequences first. After the averaged result is generated, a pair of sequences from the remaining data including the previous averaged result is determined for the next iteration. We keep on going until only one sequence is left. We provide a pseudo code in Table 5.

In this work, we propose two novel averaging functions, i.e., cubic-spline dynamic time warping (CDTW) in Section 4.1 and Iterative cubic-spline dynamic time warping (ICDTW) in Section 4.2.

Table 2

Function modified from DTW distance to find an optimal warping path.

FUNCTION $[W] = \text{DTW-AVERAGING}(A, B, \omega_A, \omega_B)$	
1.	$W = \text{WARPINGPATH}(A, B)$
2.	Let N be a length of the path W
3.	Let C be a time series sequence of length N
4.	For ($k = 1$ to N)
5.	$[i, j] = W_k$
6.	$c_k = \frac{\omega_A \cdot \omega_B + \omega_B \cdot \omega_A}{\omega_A + \omega_B}$
7.	Add c_k to C
8.	End for
9.	Return C

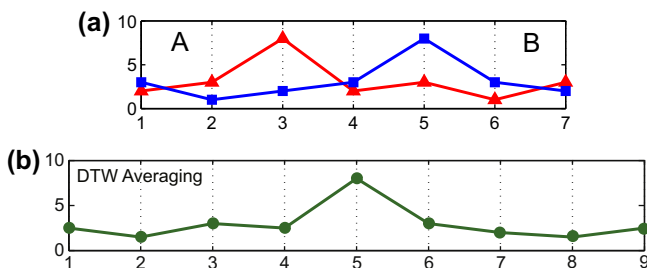


Fig. 3. (a) Two sequences A = (2, 3, 8, 2, 3, 1, 3) and B = (3, 1, 2, 3, 8, 3, 2) are averaged by the DTW averaging algorithm to generate (b) an averaged sequence C = (2.5, 1.5, 2, 3, 8, 2.5, 3, 1.5, 2.5). Note that the length of the result will increase in each averaging.

Table 3

Training/template construction phase.

FUNCTION $[T] = \text{SMTF-TRAININGPHASE}(\mathbb{D})$	
1.	Let \mathbb{L} be a set of labels in \mathbb{D}
2.	For each label L in \mathbb{L}
3.	$\mathbb{D}_L =$ data objects with the same label L
4.	$T = \text{AVERAGINGSCHEME}(\mathbb{D}_L)$
5.	Add $[T, L]$ in T
6.	End for
7.	Return T

Table 4

Test/classification phase.

FUNCTION $[label_{Best}] = \text{STMF-TESTPHASE}(\mathbb{T}, Q)$	
1.	$dist_{Best} = \text{INFINITE}$
2.	$label_{Best} = \text{NULL}$
3.	For each template T in \mathbb{T}
4.	$dist = \text{DTW-AVERAGING}(Q, T)$
5.	If $(dist < dist_{Best})$
6.	$dist_{Best} = dist$
7.	$label_{Best} = T \cdot label$
8.	End if
9.	End for
10.	Return $label_{Best}$

Either one of these two averaging functions can be used as the AVERAGINGFUNCTION in Line 4 of Table 5.

4.1. Cubic-spline dynamic time warping (CDTW) averaging function

CDTW averaging function produces a more accurate averaged result by considering both position and amplitude of each data point in a new averaged sequence, while the DTW averaging function (Table 2) considers only the amplitude. In other words, the DTW averaging function equally treats every new data point in a new sequence, while the CDTW averaging function additionally determines where a new data point should be placed. Specifically, a position and an amplitude of a data point in the sequence can be observed as an x - and y -coordinate in time series. To be more illustrative, Fig. 4 shows a comparison between two new sequences generated by CDTW and DTW. From the figure, the sequence generated from the CDTW algorithm is more useful since it preserves both position and amplitude from the warping path.

Suppose the path $W = \langle w_1, \dots, w_k, \dots, w_N \rangle$, where w_k is the k th coordinate (i_k, j_k) in the optimal path of sequences A and B , where i_k and j_k are indices of data points in sequences A and B , respectively. Therefore, a position c_{k_x} of a data point in a new sequence C is determined by $c_{k_x} = \frac{i_k \cdot \omega_A + j_k \cdot \omega_B}{\omega_A + \omega_B}$, and an amplitude c_{k_y} of a data point in a new sequence C is determined by $c_{k_y} = \frac{a_{i_k} \cdot \omega_A + b_{j_k} \cdot \omega_B}{\omega_A + \omega_B}$, where ω_A and ω_B are the weights of the sequences A and B , respectively.

However, the length of the sequence C is always equal to or longer than the two original sequences; therefore, re-sampling is required. In this work, CDTW uses a cubic-spline interpolation [2] since it requires no parameter and outperforms other

interpolation techniques in re-sampling of natural sequences. Additionally, CDTW re-samples positions of the averaged result to integer values. As illustrated in Fig. 5, the sequence C of 9 data points is re-sampled to the sequence C' of 7 data points. We provide a concrete pseudo code of CDTW in Table 6.

4.2. Iterative cubic-spline dynamic time warping (ICDTW) averaging function

Although CDTW produces a good averaged result since it considers both a position and an amplitude, another essential but not necessary condition for averaging is that the averaged result should be in the middle of two original sequences. In other words, DTW distances between the sequences and the result should be equal. Therefore, we propose an iterative approach for the CDTW averaging function called iterative cubic-spline dynamic time warping (ICDTW) averaging function that can truly represent characteristics of a set of subsequences. A good real averaged result can be determined from the result that gives minimum summation distances between the result itself and every data sequence. Specifically, if two data sequences are considered, the averaged result is the sequence which not only has minimum summation distance but also gives an equal distance between itself to these two data sequences.

We would like to emphasize that the distances between the generated result from the CDTW function and the two original time series are *not* always equal; therefore, the averaged result needs to be slightly adjusted. Obviously, since all elements in the sequence are real numbers, it is very difficult to obtain the sequence that satisfies this condition. We therefore propose a heuristic and deterministic solution, i.e., ICDTW averaging function mentioned above. To average two time series sequences A and B , the ICDTW function will find new weights β_A and β_B that make the averaged result C be the center between the sequences A and B . Obviously, finding both weights β_A and β_B is not very practical since the weights β_A and β_B are real numbers. We instead heuristically use a binary search to find only the weight β_A , when the weight β_B is

Table 5

STMF averaging scheme.

FUNCTION $[C] = \text{STMF-AVERAGINGSCHEME}(\mathbb{S})$	
1.	Initialize a weight $\omega = 1$ for each sequence S in \mathbb{S}
2.	While $(\text{Size}(\mathbb{S}) > 1)$
3.	$[A, B] = \text{Most similar sequences in } \mathbb{S}$
4.	$C = \text{AVERAGINGFUNCTION}(A, B, \omega_A, \omega_B)$
5.	Remove A and B from \mathbb{S}
6.	$\omega_C = \omega_A + \omega_B$
7.	Add C to \mathbb{S}
8.	End while
9.	Return C

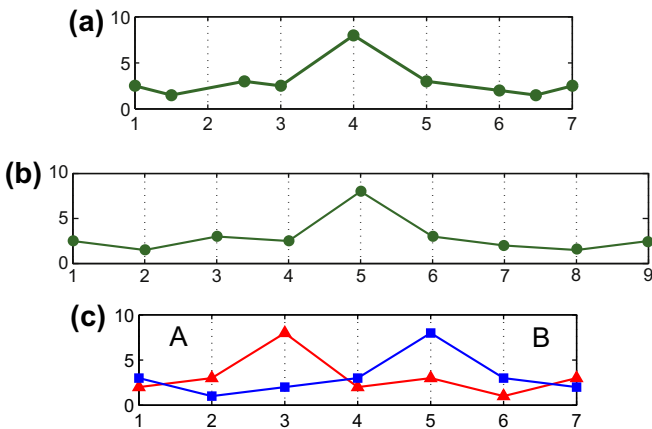


Fig. 4. Comparison between averaged results generated from (a) CDTW and (b) DTW averaging functions, where (c) two inputs are $A = (2, 3, 8, 2, 3, 1, 3)$ and $B = (3, 1, 2, 3, 8, 3, 2)$.

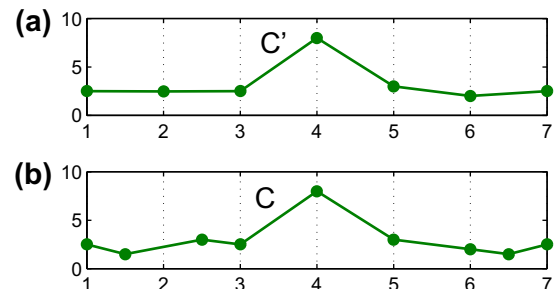


Fig. 5. (a) A new sequence C' which is re-sampled from (b) 9 data points in the sequence C in Fig. 4(a).

Table 6

Cubic-spline dynamic time warping averaging function.

FUNCTION [C] = CDTW-AVERAGING(A, B, ω_A, ω_B)	
1.	$W = \text{WARPINGPATH}(A, B)$
2.	Let N be the length of the path W
3.	Let N' be the length of time series A and B
4.	Let C be a time series sequence of size N
5.	Let C' be a time series sequence of size N'
6.	For ($k = 1$ to N)
7.	$[i, j] = w_k$
8.	$x = \frac{i \cdot \omega_A + j \cdot \omega_B}{\omega_A + \omega_B}$
9.	$y = \frac{a_i \cdot \omega_A + b_j \cdot \omega_B}{\omega_A + \omega_B}$
10.	Add $[x, y]$ to C
11.	End for
12.	$C' = \text{CUBICSPLINE}(C)$
13.	Return C'

Table 7

Iterative cubic-spline dynamic time warping averaging function.

FUNCTION [C] = ICDTW-AVERAGING(A, B, ω_A, ω_B)	
1.	Initialize weights $\beta_{A_1} = 10^{-5}$, $\beta_{A_2} = 10^5$, and $\beta_B = 1$
2.	Initialize weight $\beta_{A_3} = \frac{(\beta_{A_1} + \beta_{A_2})}{2}$
3.	$C = \text{CDTW-AVERAGING}(A, B, \beta_{A_3}, \beta_B)$
4.	$d_{CA} = \text{DTWDISTANCE}(C, A) \cdot \omega_A$
5.	$d_{CB} = \text{DTWDISTANCE}(C, B) \cdot \omega_B$
6.	$\beta_{A_3} = d_{CA} < d_{CB} ? \beta_{A_1} : \beta_{A_2}$
7.	While ($(d_{CA} - d_{CB}) > 0$)
8.	$\beta_{A_3} = \frac{(\beta_{A_1} + \beta_{A_2})}{2}$
9.	$C = \text{CDTW-AVERAGING}(A, B, \beta_{A_3}, \beta_B)$
10.	$d_{CA} = \text{DTWDISTANCE}(C, A) \cdot \omega_A$
11.	$d_{CB} = \text{DTWDISTANCE}(C, B) \cdot \omega_B$
12.	If ($d_{CA} < d_{CB}$)
13.	$\beta_{A_1} = \beta_{A_3}$
14.	Else
15.	$\beta_{A_2} = \beta_{A_3}$
16.	End if
17.	End while
18.	Return C

Table 8

Details of datasets.

Dataset	Number of classes	Length	Size of training set	Size of test set
Synthetic Control	6	60	300	300
CBF	3	128	30	900
Face All	14	131	560	1690
OSU Leaf	6	427	200	242
50 Words	50	270	450	455
Trace	4	275	100	100
Two Patterns	4	128	1000	4000
Wafer	2	152	1000	6174
Face Four	4	350	24	88
Lightning-2	2	637	60	61
Lightning-7	7	319	70	73
ECG	2	96	100	100
Adiac	37	176	390	391
Yoga	2	426	300	3000
Fish	7	463	175	175

fixed. Specifically, for each iteration, a new weight β_A is considered whether or not the generated averaged result C has an equal DTW distance to the sequences A and B . If the distance is equal, ICDTW terminates. In other words, we only need to determine the weight β_A and hold the weight β_B constant because two sets of weights are equivalent. For example, for $\{\beta_A, \beta_B\} = \{4, 5\}$, it can be reduced to

$\{0.8, 1\}$ when the weight β_B is fixed to 1; therefore, searching for β_B is enough to find any pair of weights $\{\beta_A, \beta_B\}$. Pseudo code of the ICDTW averaging function is provided in Table 7.

Note that both CDTW and ICDTW averaging functions can be used in STMF under users' preference. For CDTW, the averaged result preserves the shapes of both original sequences by considering both the position and the amplitude of the warping alignment, while ICDTW averaging returns more accurate characteristics of the averaged result by calibrating the averaged sequence to have the same distance between itself and the two original sequences. Performances of CDTW and ICDTW will be demonstrated in the next section.

5. Experimental evaluation

Three following experiments will demonstrate the superiority of our proposed method over the current existing approaches. The first experiment shows accuracies of our shape-based averaging method, i.e., a new averaging scheme with two proposed CDTW and ICDTW algorithms, comparing with NLAFF. In the second experiment, we show that our STMF with both CDTW and ICDTW outperforms NLAFF and traditional nearest neighbor classification in terms of accuracy, storage requirement, and time usage for classification problems. Our extension of STMF is also evaluated to show that STMF can support multiple templates within each class, and to show that traditional nearest neighbor classification is simply a special case. All codes are implemented in C++ and run on an Intel Core i7 desktop computer. We evaluate our proposed method with 15 datasets from the publically available UCR classification/clustering archive [8]. Table 8 shows the number of classes, length of each time series data, and size of training/test sets.

5.1. First experiment

In this experiment, we will demonstrate that our proposed averaging methods, which utilize a new averaging scheme with CDTW and ICDTW, well represent sequences in the datasets. For each dataset, its training data and test data are all combined, and then all sequences are averaged. In real-world applications, sequences should be separated by its own class to achieve maximum utilities. The averaged results are evaluated using SumDist function defined as a summation of the distance between the averaged result and all the original sequences in the dataset. If a value from SumDist is small, it means that this method generates a good averaged result. SumDist function is provided as follows.

$$\text{SumDist}(\hat{D}, \mathbb{D}) = \sum_{i=1}^{|\mathbb{D}|} \text{DTWDISTANCE}(\hat{D}, D_i) \quad (2)$$

where \mathbb{D} is a dataset, \hat{D} is the averaged result, and D_i is a data sequence in the dataset \mathbb{D} .

Table 9 shows the comparison between SumDist of NLAFF and our proposed methods, CDTW and ICDTW. The lowest SumDist for each dataset is emphasized in bold. Both proposed methods achieve lower SumDist values since all sequences are averaged using a new averaging schemes, while NLAFF averages sequences in random manner. In addition, no re-sampling method is adopted in NLAFF to scale the averaged sequence down to the same length. This means that at each step, NLAFF will produce a longer averaged sequence. It is apparent from the experiment results that CDTW and ICDTW generate more accurate averages.

We can see from the results that ICDTW usually yields smaller SumDist than CDTW in almost all of the datasets, and smaller SumDist than NLAFF in every dataset. However, in some cases such as 50 Words and Yoga datasets, where sequences within the class are very diverse, ICDTW may give slightly larger SumDist than CDTW (but still performs much better than NLAFF); in such cases, CDTW performs better because ICDTW tries its best to equalize the

Table 9
SUMDIST of averaging methods.

Dataset	NLAAF	STMF	
		CDTW	ICDTW
Synthetic Control	8962	3545	3538
CBF	17,827	5119	4856
Face All	43,314	12,820	12,597
OSU Leaf	12,179	3313	3276
50 Words	28,810	5034	5215
Trace	9417	2054	2046
Two Patterns	105,517	34,376	33,268
Wafer	693,016	54,115	54,115
Face Four	1735	757	729
Lightning-2	2877	1223	1196
Lightning-7	2961	1126	1113
ECG	1504	545	544
Adiac	1884	512	495
Yoga	163,207	16,826	17,193
Fish	944	371	364

Table 10
Classification accuracy and time usage (seconds) of STMF compared with NLAAF.

Dataset	NLAAF	STMF	
		CDTW	ICDTW
Synthetic Control	0.80 (1)	0.97 (0)	0.92 (0)
CBF	0.94 (4)	0.96 (1)	0.95 (1)
Face All	0.57 (46)	0.83 (11)	0.81 (12)
OSU Leaf	0.35 (67)	0.41(13)	0.42 (7)
50 Words	0.42 (75)	0.60 (44)	0.58 (43)
Trace	0.92 (6)	0.98 (1)	1.00 (2)
Two Patterns	0.92 (122)	0.97 (7)	0.95 (13)
Wafer	0.10 (514)	0.64 (14)	0.63 (8)
Face Four	0.56 (2)	0.83 (1)	0.81 (1)
Lightning-2	0.56 (16)	0.56 (1)	0.54 (2)
Lightning-7	0.59 (7)	0.66 (2)	0.70 (1)
ECG	0.65 (1)	0.70 (0)	0.71 (0)
Adiac	0.48 (17)	0.49 (11)	0.47 (12)
Yoga	0.48 (684)	0.48 (27)	0.48 (27)
Fish	0.57 (25)	0.58 (7)	0.59 (8)

Table 11
Storage requirement in KB for storing templates.

Dataset	NLAAF	STMF	Original dataset
Synthetic Control	12.40	1.44	72.00
CBF	43.91	1.54	15.36
Face All	68.35	7.36	293.44
OSU Leaf	172.15	10.25	341.60
50 Words	303.87	54.00	486.00
Trace	68.87	4.40	110.00
Two Patterns	117.81	2.05	512.00
Wafer	476.97	1.22	608.00
Face Four	24.21	5.60	33.60
Lightning-2	120.37	5.10	152.88
Lightning-7	64.37	8.93	89.32
ECG	13.30	0.77	38.40
Adiac	44.14	26.05	274.56
Yoga	477.70	3.41	511.20
Fish	82.44	12.96	324.10

distances, but this in turn could increase the summarized distance (SUMDIST). ICDTW guarantees that distances between the averaged result and the two original sequences are identical, though not guarantee to be minimal. In other words, there is a tradeoff between having an exact average with equal distances and having an approximate average with minimal distances.

5.2. Second experiment

This experiment demonstrates the utility of our proposed shape-based template matching framework (STMF) over a

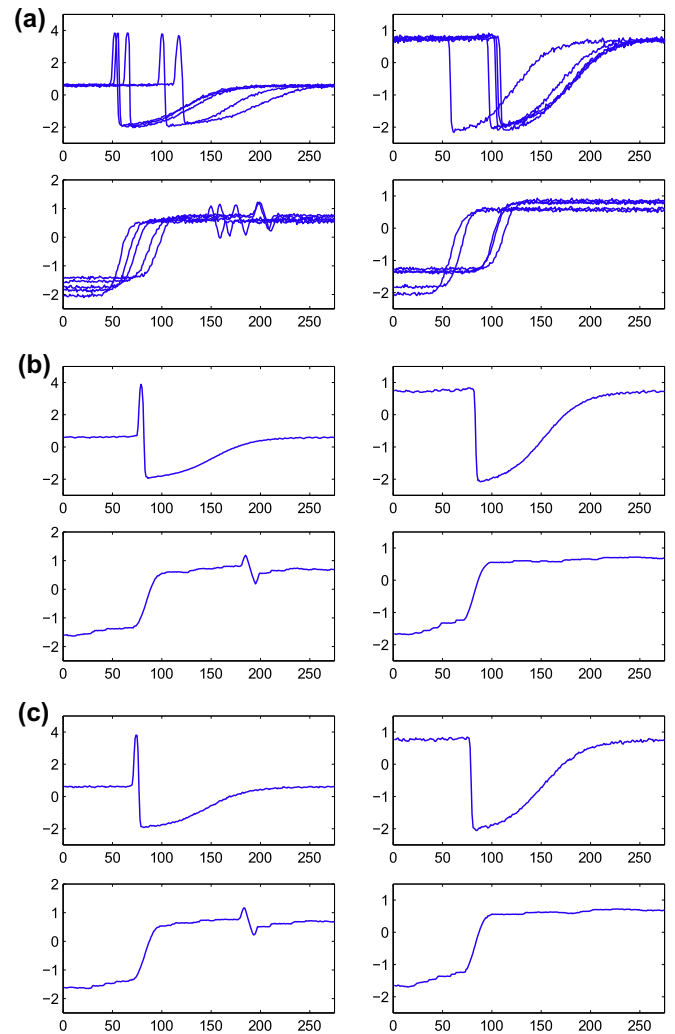


Fig. 6. (a) Example sequences of four classes of the Trace dataset and templates of each class generated from STMF with (b) CDTW and (c) ICDTW.

template matching with NLAAF in terms of three metrics, i.e., classification accuracy, storage requirements, and time usage. Classification accuracy is determined by a classification on a set of templates, and storage requirement is measured from the amount of memory needed to store a set of templates. Time needed to classify the test dataset is also reported. Comparison of accuracy, storage, and time usage are shown in Tables 10 and 11, where our method outperforms NLAAF in every dataset. Templates of Trace dataset generated with CDTW and ICDTW are illustrated in Fig. 6. The best result is emphasized in bold for each dataset.

The main reason is the fact that data sequences in most datasets distribute unequally. Although ICDTW generates a more accurate averaged result, this averaged result may not be the best sequence to represent the whole unequal distribution of the dataset; each class may best be represented by two or more templates since multiple sub-classes may exist. Therefore, by using only one template to represent every sequence of the same class, ICDTW could return averaged characteristics of those multiple sub-classes which tends to increase classification error.

5.3. Third experiment

Our proposed averaging scheme allows STMF to produce multiple templates for each class in classification. In this experiment, we show the classification accuracy of STMF when the number of

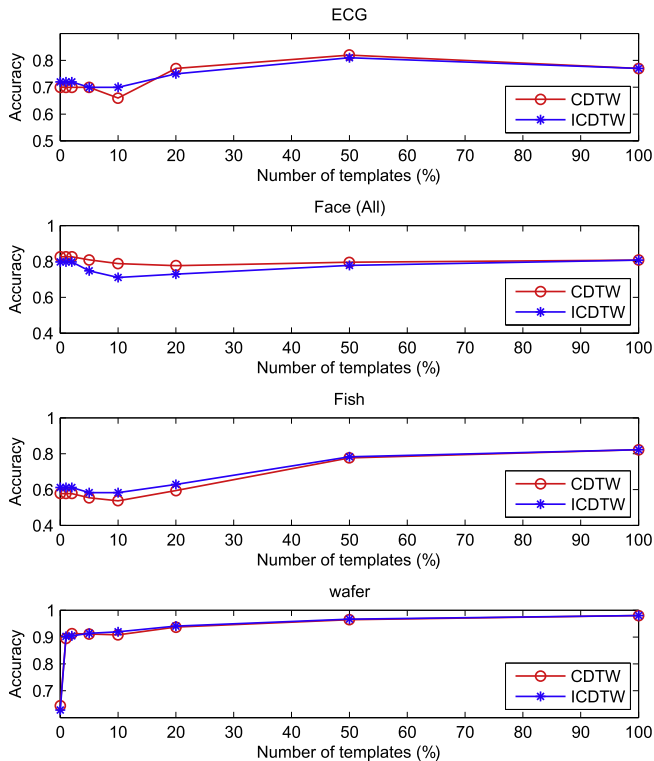


Fig. 7. Classification accuracy of ECG, Face All, Fish, and Wafer datasets when the number of templates are varied.

templates are varied. The number of templates used in real-world applications depends on resources of the system, i.e., memory storage and computational power. Specifically, the system with small memory storage and limited computational power is suggested to use smallest number of templates possible. In Fig. 7, we shows classification accuracy when the size of templates are varied in four datasets, i.e., ECG, Face All, Fish, and Wafer. The classic nearest neighbor classification is considered a special case when the number of templates is set to the number of instances in the class. In other words, every training sequence is used in template matching and no sequence is discarded.

6. Conclusion and future work

In this work, we propose a novel shape-based template matching framework which utilizes a new averaging scheme and averaging functions to generate an accurate set of templates. This set of templates is used as a dataset for query classification. Compared with the existing method, our proposed method outperforms in every case in terms of classification accuracy, time usage, and storage requirement. In addition, our method can also be extended to generate two or more templates for each class when users have more resources in real-world applications. This research can be applied to diverse domains where time series classification is needed and it will be more useful when the system has limited resource in terms of storage and computational power.

Acknowledgement

This research was supported in part by the Thailand Research Fund given through the Royal Golden Jubilee Ph.D. Program (PHD/0141/2549 to V. Niennattrakul and C.A. Ratanamahatana), in part by the Chulalongkorn University Graduate Scholarship to Commemorate the 72nd Anniversary of His Majesty King Bhumibol

Adulyadej, and in part by the 90th Anniversary of Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund).

References

- [1] D.J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, in: KDD Workshop, 1994, pp. 359–370.
- [2] R.L. Burden, J.D. Faires, A.C. Reynolds, Numerical Analysis, 1997.
- [3] R. de A. Araújo, A class of hybrid morphological perceptrons with application in time series forecasting, Knowledge-based Systems 24 (4) (2011) 513–529.
- [4] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, E.J. Keogh, Querying and mining of time series data: experimental comparison of representations and distance measures, Proceedings of the VLDB Endowment 1 (2) (2008) 1542–1552.
- [5] L. Gupta, D. Molfese, R. Tammanna, P. Simos, Nonlinear alignment and averaging for estimating the evoked potential, IEEE Transactions on Biomedical Engineering 43 (4) (1996) 348–356.
- [6] F. Itakura, Minimum prediction residual principle applied to speech recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 23 (1) (1975) 67–72.
- [7] E. Keogh, C.A. Ratanamahatana, Exact indexing of dynamic time warping, Knowledge and Information Systems 7 (3) (2005) 358–386.
- [8] E. Keogh, X. Xi, L. Wei, C.A. Ratanamahatana, UCR time series classification/clustering page. <http://www.cs.ucr.edu/~eamonn/time_series_data>, 2010.
- [9] E.J. Keogh, L. Wei, X. Xi, M. Vlachos, S.-H. Lee, P. Protopapas, Supporting exact indexing of arbitrarily rotated shapes and periodic time series under euclidean and warping distance measures, The VLDB Journal 18 (3) (2009) 611–630.
- [10] S.-W. Kim, S. Park, W.W. Chu, An index-based approach for similarity search supporting time warping in large sequence databases, in: Proceedings of the 17th International Conference on Data Engineering (ICDE'01), Heidelberg, Germany, 2001, pp. 607–614.
- [11] Z. Kovacs-Vajna, A fingerprint verification system based on triangular matching and dynamic time warping, IEEE Transactions of Pattern Analysis and Machine Intelligence 22 (11) (2000) 1266–1276.
- [12] Y.-S. Lee, L.-I. Tong, Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming, Knowledge-based Systems 24 (1) (2011) 66–72.
- [13] H. Li, C. Guo, Piecewise cloud approximation for time series mining, Knowledge-based Systems 24 (4) (2011) 492–500.
- [14] M.E. Munich, P. Perona, Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification, in: Proceedings of the 7th International Conference on Computer Vision (ICCV'99), Corfu, Greece, 1999, p. 108.
- [15] V. Niennattrakul, C.A. Ratanamahatana, Inaccuracies of shape averaging method using dynamic time warping for time series data, in: Proceedings of the 7th International Conference on Computational Science (ICCS'07), Beijing, China, 2007, pp. 513–520.
- [16] V. Niennattrakul, P. Ruengronghirunya, C. Ratanamahatana, Exact indexing for massive time series databases under time warping distance, Data Mining and Knowledge Discovery 21 (3) (2010) 509–541.
- [17] V. Niennattrakul, D. Wanichsan, C.A. Ratanamahatana, Hand geometry verification using time series representation, in: Proceedings of 11th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES'07), Vietri sul Mare, Italy, 2007, pp. 824–831.
- [18] C.A. Ratanamahatana, E. Keogh, Making time-series classification more accurate using learned constraints, in: Proceedings of the Fourth SIAM International Conference on Data Mining (SDM'04), Lake Buena Vista, FL, 2004, pp. 11–22.
- [19] C.A. Ratanamahatana, E.J. Keogh, Multimedia retrieval using time series representation and relevance feedback, in: Proceedings of 8th International Conference on Asian Digital Libraries (ICADL2005), Bangkok, Thailand, 2005, pp. 400–405.
- [20] C.A. Ratanamahatana, E.J. Keogh, Three myths about dynamic time warping data mining, in: Proceedings of the Fifth SIAM International Data Mining Conference (SDM'05), Newport Beach, CL, 2005, pp. 506–510.
- [21] J.J. Rodriguez, C.J. Alonso, J.A. Maestro, Support vector machines of interval-based features for time series classification, Knowledge-Based Systems 18 (4–5) (2005) 171–178.
- [22] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 26 (1) (1978) 43–49.
- [23] Y. Sakurai, M. Yoshikawa, C. Faloutsos, FTW: Fast similarity search under the time warping distance, in: Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Baltimore, MD, 2005, pp. 326–337.
- [24] S. Salvador, P. Chan, Toward accurate dynamic time warping in linear time and space, Intelligent Data Analysis 11 (5) (2007) 561–580.
- [25] D. Srisai, C.A. Ratanamahatana, Time series shape averaging using time-warping alignment with re-sampling, in: Proceedings of the 6th International Joint Conference on Computer Science and Software Engineering (JCSSE'09), Phuket, Thailand, 2009.
- [26] H. Tang, S.S. Liao, Discovering original motifs with different lengths from time series, Knowledge-based Systems 21 (7) (2008) 666–671.
- [27] X. Weng, J. Shen, Classification of multivariate time series using locality preserving projections, Knowledge-Based Systems 21 (7) (2008) 581–587.

- [28] X. Weng, J. Shen, Classification of multivariate time series using two-dimensional singular value decomposition, *Knowledge-Based Systems* 21 (7) (2008) 535–539.
- [29] D. Yankov, E. Keogh, L. Wei, X. Xi, W. Hodges, Fast best-match shape searching in rotation-invariant metric spaces, *IEEE Transactions on Multimedia* 10 (2) (2008) 230–239.
- [30] B.-K. Yi, H.V. Jagadish, C. Faloutsos, Efficient retrieval of similar time sequences under time warping, in: *Proceedings of the Fourteenth International Conference on Data Engineering (ICDE'98)*, Orlando, FL, 1998, pp. 201–208.
- [31] Y. Zhu, D. Shasha, Warping indexes with envelope transforms for query by humming, in: *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD'03)*, San Diego, CA, 2003, pp. 181–192.