TIME SERIES CLASSIFICATION BASED ON THE LONGEST COMMON SUBSEQUENCE SIMILARITY AND ENSEMBLE LEARNING

[1] *Guan-Cheng Guo*, [2] *Kuo-Si Huang*, and [1,†] *Chang-Biau Yang*

[1]Dept. of Computer Science and Engineering, National Sun Yat-sen University
[†]Corresponding Author, E-mail: cbyang@cse.nsysu.edu.tw
[2]Dept. of Information Management, National Kaohsiung Marine University

ABSTRACT

The dynamic time warping (DTW) algorithm provides a powerful way to measure the distance between two time series. However, the DTW algorithm may not be suitable for all time series of various types. This paper proposes a similarity measurement for two time series consisting of real numbers based on the concept of the longest common subsequence (LCS) problem with the data diversity. In addition, for reducing the error rates of time series classification, the behavior knowledge space (BKS) method is used to build ensemble classifiers by combining three classifiers, including DTW with warping window (DTWW), derivative dynamic time warping (DDTW) and LCS. The experimental results show that the LCS similarity measurement with diversity can get good accuracy comparable to the DTW algorithm. In addition the BKS method improves the error rate about 20% over the previously best-known DTWW method.
Keywords: Time Series Classification; Dynamic Time Warping; Longest Common Subsequence; Diversity.

1. INTRODUCTION

A *time series* is a sequence of data points obtained from consecutive observations or measurements over a specific time interval. The time series classification problem plays an important role in data mining and it has widely application in various domains [1–4]. The most commonly used method for the time series classification problem is the *simple nearest neighbor* (1-NN) algorithm [1, 5–7]. The 1-NN algorithm measures the distance between the target time series and each time series in the trained clusters, then a nearest cluster can be found. Accordingly, the target time series is classified to this nearest cluster.

The accuracy of the 1-NN algorithm deeply depends on the method of distance calculation between time series sequences. The *dynamic time warping* (DTW) algorithm [8, 9] was proposed for efficiently measuring distance between time series sequences. By warping the time axis to align two sequences of time series, the DTW algorithm obtains a better distance measurement between them.

The DTW algorithm may not be suitable for all time series of various types. This paper proposes a similarity measurement for two time series composed of real numbers based on the concept of the longest common subsequence (LCS) problem with the *data diversity*. The diversity can be used to determine a better matching threshold for the LCS similarity measurement.

For evaluating the performance of our algorithm for time series classification, some experiments are performed over the 47 datasets obtained from the UCR web site[10]. In the experiments, we compare classification error rates of the DTW [8], the DTW with warping window (DTWW) [11, 12], the derivative DTW (DDTW) [6], and the LCS with diversity algorithms. Furthermore, we apply the *behavior knowledge space* (BKS) method [13, 14] to build an ensemble classifier. As the experimental results show, the LCS similarity measurement with diversity is also comparable to the DTW algorithms. In addition, the performances of the BKS ensembles are better than every individual classifier in most datasets.

The organization of this paper is as follows. Section 2 introduces the preliminary knowledge about DTW, LCS and BKS. Our method for the time series classification is proposed in Section 3. Section 4 presents the materials, results and discussions of our experiments with the datasets downloaded from the UCR time series classification/clustering homepage [10]. Finally, Section 5 concludes this paper.

2. PRELIMINARIES

2.1. Dynamic time warping

The dynamic time warping (DTW) is a dynamic programming algorithm for calculating the distance between two time series sequences. Suppose two sequences of time series $A = a_1a_2 \cdots a_m$ and $B = b_1b_2 \cdots b_n$. Let $DTW(i,j)$ denote the DTW distance of $A_{1..i}$ and $B_{1..j}$. The distance function $dis(a_i, b_j)$ represents the distance of a_i and b_j. The Euclidean distance is usually used to measure the distance between a_i and b_j. The dynamic programming formula of the DTW algorithm is given in Equation 1 [8].

However, there exists some weakness in the DTW algorithm. In an extreme alignment case, a lot of data points are aligned to a specific point. The *warping windows Sakoe-Chiba band* [12] and *Itakura paralleogram* [15], as shown in Figure 1, are two well-known constraint types of warping window to fix the extreme alignment issue.

The DTW with warping window (DTWW) [11, 12] adds the window constraint on the calculation of $dis(a_i, b_j)$. If $|i - j|$ is greater than the window constraint, then the calculation of $dis(a_i, b_j)$ is forbidden. In other words, $dis(a_i, b_j) = \infty$.

2.2. Longest common subsequence

The *longest common subsequence* (LCS) problem has been extensively studied for several decades [16–21]. Finding the LCS between two strings (sequences) can be used to measure their similarity. In the LCS problem, $A = a_1a_2 \cdots a_m$ and $B = b_1b_2 \cdots b_n$ denote input sequences where a_i in A, $1 \leq i \leq m$, and b_j, $1 \leq j \leq m$, in B are characters or symbols, instead of real numbers in the time series. By deleting the characters from a sequence without changing the order of the remaining characters, we can get a *subsequence*. The LCS is the *common subsequence* of both A and B with the maximum length. For example, suppose $A =$ `BABCADAB` and $B =$ `ACCEDABC`. Then, the LCS of A and B is `ACDAB` with length 5.

The LCS problem can be solved by the dynamic programming approach. Let $LCS(i,j)$ denote the LCS length of $A_{1..i}$ and $B_{1..j}$, where $A_{1..i}$ and $B_{1..j}$ are the prefix substrings of A and B, respectively, and $1 \leq i \leq m$ and $1 \leq j \leq n$. $LCS(i,j)$ can be calculated by Equation 2 [18].

2.3. Behavior knowledge space

The *behavior knowledge space* (BKS) can be used to integrate multiple classifiers to build a new classifier [13, 14]. Assume that q classifiers of a dataset D with c classes are combined to build an ensemble classifier. In the training stage, each classifier i generates a predicted class label l_i for every time series A in D, where $1 \leq i \leq q$. The count of each entry $En(A) = (l_1, l_2, \cdots, l_q)$, generated from all time series, is recorded in the BKS table. The BKS table has c^q possible entries because each classifier provides one predicted class for each target series. In other words, the entries constructed from the entire training set constitute a knowledge space which characterizes the preferences of these q classifiers.

In the BKS table, each entry is combined by the prediction set of class labels from q classifiers, and the counts of the time series with the true class labels. In the testing stage, the target time series B can be classified by the q classifiers and the set $En(B)$ of class labels is generated. Then we search the entries in the BKS table and select the entry which has the most appearance of the true class label.

In brief, the BKS method contains knowledge modelling (training) and recognition (testing) stages. In the knowledge modelling stage, it uses the training set to build the BKS table. In the recognition (testing) stage, the predicted class of the target time series in the testing set is decided according to the decisions generated from individual classifiers and the BKS table.

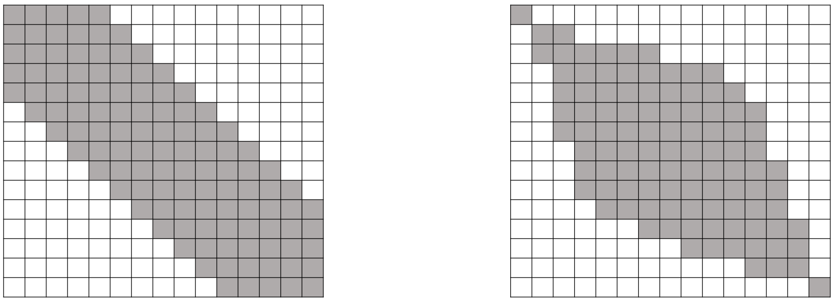
3. METHODS

3.1. LCS similarity measurement

The formula of the traditional LCS algorithm considers only two states of a character pair, *match* and *mismatch*. However, in the time series classification problem, one time series is composed of real numbers. The data point pair of real numbers cannot be determined to fall into the only two states of match and mismatch. The distance between two data points of real numbers is used to determine whether they are close enough to be regarded as a match or not. The *longest common subsequence similarity measurement* (LCSS) is modified from LCS for measuring the similarity of two times series of real numbers [22–24].

Unlike the traditional LCS problem, the LCSS algorithm measures the similarity of two sequences with a matching threshold θ. A threshold θ is used to determine the state of match or mismatch between two real numbers and it can be defined by users to obtain more meaningful alignment. If the Euclidean distance of two data points is not greater than θ, we regard the two data points to be a match. Otherwise, the two data points are said to be a mismatch. We can modify the LCS algorithm for measuring the similarity between two time series with the threshold θ. Let $LCSS(i,j)$ denote the

$$DTW(i,j) = \begin{cases} 0 & \text{if } i = 0 \text{ and } j = 0, \\ \infty & \text{if } i = 0 \text{ and } j \neq 0, \\ \infty & \text{if } i \neq 0 \text{ and } j = 0, \\ dis(a_i, b_j) + \min \begin{cases} DTW(i-1, j) \\ DTW(i, j-1) \\ DTW(i-1, j-1) \end{cases} & \text{otherwise.} \end{cases} \tag{1}$$



(a) (b)

Figure 1: Two constrained warping windows. (a) The Sakoe-Chiba band [12]. (b) The Itakura parallelogram [15].

LCS similarity of $A_{1..i}$ and $B_{1..j}$ by considering that a_i and b_j are matched if $|a_i - b_j| \leq \theta$. The formula for computing LCS similarity is given in Equation 3.

According to the distribution and characteristic of a dataset, we have to adjust the value of θ when the LCS algorithm is applied to calculate the similarity of two sequences of real numbers. If the accuracy of the time series classification is considered, the θ value becomes a critical factor.

3.2. Diversity

Each dataset of time series may have its own characteristic. For example, the line charts of two datasets, 50Words and Adiac, are shown in Figure 2. In the 50Words dataset, we can see that the time series are not similar. But in the Adiac dataset, the time series are almost overlapped. To distinguish the characteristic of each dataset, we propose a method to calculate the diversity of each dataset. Then the diversity will be applied to adjust the matching threshold θ for the dataset.

The *diversity*, denoted by d, can be used to represent the discrimination of a dataset of time series. The diversities of different time series datasets may be distinguishable. In a specific dataset whose time series are very similar, the difference between different classes would be very small.

To obtain d, we calculate the *arithmetic mean* (AM) and *harmonic mean* (HM) of all time series in the dataset on the time axis. Suppose that we have a dataset $D = \{S_1, S_2, \ldots, S_{|D|}\}$, where each time series S_i consists of L real numbers, $S_i = S_{i,1}S_{i,2}\ldots S_{i,L}$, $1 \leq i \leq |D|$. The formulas for calculating AM and HM are shown in Equations 4 and 5, respectively [27, 28].

$$AM_j = \frac{\sum_{i=1}^{|D|} S_{i,j}}{|D|}, 1 \leq j \leq L, \tag{4}$$

$$HM_j = \frac{|D|}{\sum_{i=1}^{|D|} \frac{1}{S_{i,j}}}, 1 \leq j \leq L. \tag{5}$$

The diversity d is defined as the average value of the differences of AM and HM in Equation 6.

$$d = \frac{\sum_{j=1}^{L} (AM_j - HM_j)}{L}. \tag{6}$$

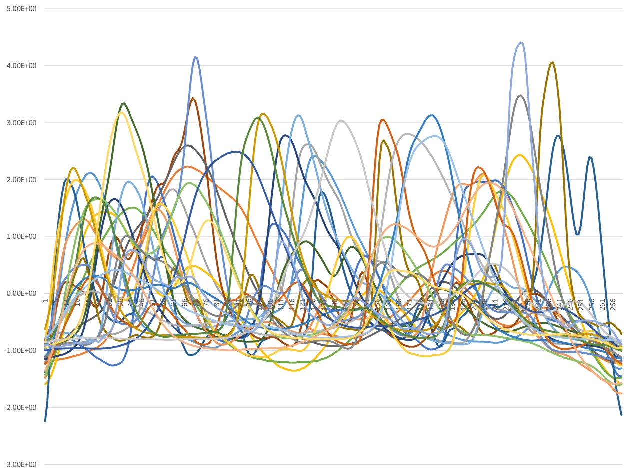
3.3. Training strategy

In a dataset D of time series, let $D_{\max}$ and $D_{\min}$ denote the maximum value and minimum value of all time series in D, respectively. Let θ' be a parameter to denote the percentage of the difference between $D_{\max}$ and $D_{\min}$. The matching threshold θ is defined in Equation 7.

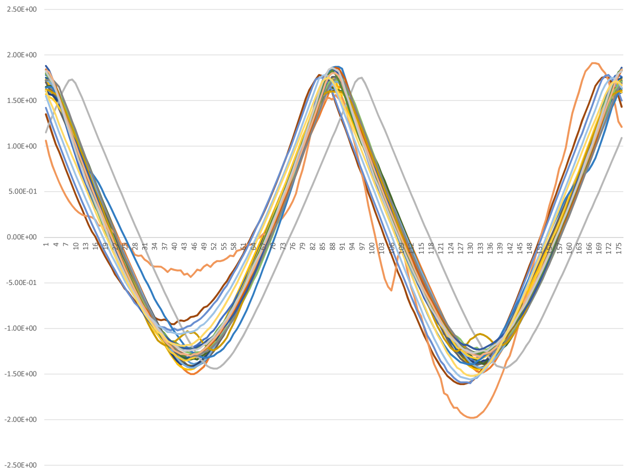
$$\theta = (D_{\max} - D_{\min}) \times \theta'. \tag{7}$$

$$LCS(i,j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0, \\ LCS(i-1, j-1) + 1 & \text{if } a_i = b_j, \\ \max\{LCS(i-1,j), LCS(i, j-1)\} & \text{if } a_i \neq b_j. \end{cases} \quad (2)$$

$$LCSS(i,j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0, \\ LCSS(i-1, j-1) + 1 & \text{if } |a_i - b_j| \leq \theta, \\ \max\{LCSS(i-1,j), LCSS(i, j-1)\} & \text{otherwise.} \end{cases} \quad (3)$$



(a)



(b)

Figure 2: Line charts of time series in training sets 50Words and Adiac, where the x-axis and y-axis denote time and raw data value of time series, respectively. (a) The training set of 50Words [25]. (b) The training set of Adiac [26].

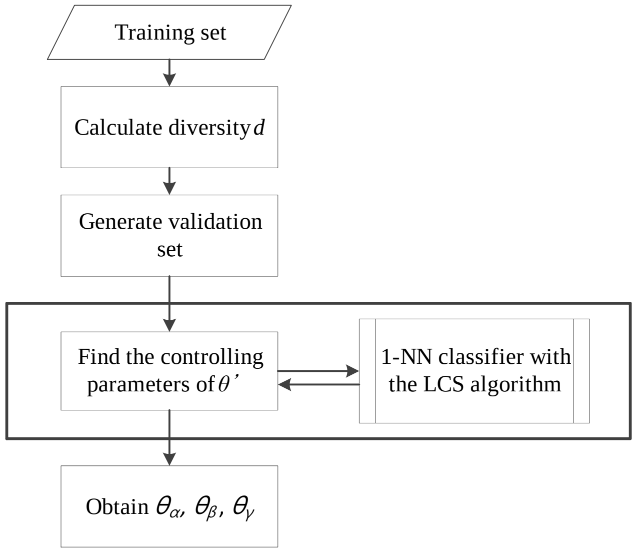


Figure 3: The flow chart of the training stage.

The parameter θ' can be adjusted by the diversity d. Equation 8 is the formula for computing θ'.

$$\theta' = \theta_\alpha \times (d^{\theta_\beta} + \theta_\gamma), \quad (8)$$

where θ_α, θ_β and θ_γ are the controlling parameters of θ'.

Our training strategy is to find suitable values of these three controlling parameters of θ' to set the proper threshold θ for improving accuracy. The flowchart of the training stage is shown in Figure 3.

For training the parameters, we select a subset of time series randomly from the training set as the validation set. The size of the validation set is the half size of the training set. The 1-NN classifier with the LCS algorithm is used to classify the validation set to find better parameters of θ'. We use a greedy method to search the better values of parameters θ_α, θ_β and θ_γ. For each value of $\theta_\beta \in \{0.5, 1, 2\}$, we search the better value of $\theta_\alpha \in \{0.1, 0.15, 0.2, \cdots, 0.95, 1, 1.5, 2, \cdots, 9.5, 10\}$. After one better θ_α is fixed, we search the better value of $\theta_\gamma \in \{1, 0.5, 0.01, 0.05, \cdots, 0.0005, 0.0001, 0\}$. The values of parameters $\theta_\alpha = 0.15$, $\theta_\beta = 0.5$ and $\theta_\gamma = 0$ are obtained with the lower error rate for all UCR datasets.

4. EXPERIMENTAL RESULTS

4.1. Experiment setting

For evaluating the performance of our algorithm applied to the time series classification, we perform some experiments and compare the performances of various algorithms, including our LCS algorithm with diversity and other previously published algorithms. The time series datasets for experiments are obtained from the UCR time series classification/clustering web site [10]. The UCR collects 47 time series datasets of various types and each dataset is partitioned into the training set and the testing set. These datasets can be divided into four types according to the source to the produce the time series data [29]. For example, the 50Words dataset is converted from the outline of handwritten words [25], and the Gun-Point dataset is converted from the motion of hands when lifting the gun [11]. Table 1 shows the detailed information of the 47 datasets in UCR.

4.2. Classification with behavior knowledge space

For improving the accuracy of classification, we apply the *behavior knowledge space* (BKS) method [13, 14] to construct an ensemble classifier for classification. The BKS method contains the training stage and the testing stage. First, we have to construct the BKS table. We choose DTWW (DTW with warping window), DDTW and LCS to build the BKS table because these three algorithms have lower error rates and they are relatively complementary in classification results. The BKS table records the largest occurrence for the combination of DTWW, DDTW and LCS. For example, assume that the predicted classes of one time series in the training set by DTWW, DDTW and LCS are classes C_3, C_1, and C_3, respectively. If the real class is C_3, the count of C_3 in the entry of C_3, C_1, and C_3 would be added one. Next, in the testing stage, if the predicted classes of a target time series predicted by DTWW, DDTW and LCS are C_3, C_1, and C_3, respectively, then the entry of C_3, C_1, C_3 will be examined. The largest occurrence of the real class in this entry will be assigned to be the predicted class of the target.

In this experiment, the approaches of the BKS ensemble in the training stage and in the test stage are described as follows. In the training stage, the predicted results of the k-NN classifier with DTWW, DDTW and LCS for each k value ($k \in \{1, 3, 5, \cdots, 15\}$) are used to build the BKS table. Each time series contributes 8 counts of predicted classes to the BKS table.

Table 1: The information of the 47 datasets in UCR [10, 29].

Size or Length / Dataset	Number of Classes	Size of Training Set	Size of Testing Set	Length of Time Series
Image Outline (15)				
OSU Leaf	6	200	242	427
SwedishLeaf	15	500	625	128
50Words	50	450	455	270
FaceFour	4	24	88	350
Adiac	37	390	391	176
Fish	7	175	175	463
Car	4	60	60	577
DiatomSizeReduction	4	16	306	345
FacesUCR	14	200	2050	131
MedicalImages	10	381	760	99
Symbols	6	25	995	398
WordsSynonyms	25	267	638	270
FaceAll	14	560	1690	131
Yoga	2	300	3000	426
Plane	7	105	105	144
Motion (9)				
Gun-Point	2	50	150	150
CricketX	12	390	390	300
CricketY	12	390	390	300
CricketZ	12	390	390	300
Haptics	5	155	308	1092
InlineSkate	7	100	550	1882
uWaveX	8	896	3582	315
uWaveY	8	896	3582	315
uWaveZ	8	896	3582	315
Sensor Reading (20)				
Trace	4	100	100	275
Lightning2	2	60	61	637
Lightning7	7	70	73	319
ECG	2	100	100	96
Beef	5	30	30	470
Coffee	2	28	28	286
OliveOil	4	30	30	570
ECGFiveDays	2	23	861	136
ItalyPowerDemand	2	67	1029	24
MoteStrain	2	20	1252	84
SonyAIBORobotII	2	27	953	65
SonyAIBORobot	2	20	601	70
TwoLeadECG	2	23	1139	82
Wafer	2	1000	6174	152
CinCECG	4	40	1380	1639
ChlorineConcentration	3	467	3840	166
MALLAT	8	55	2345	1024
StarLightCurves	3	1000	8236	1024
ECGThorax1	42	1800	1965	750
ECGThorax2	42	1800	1965	750
Simulation (3)				
Synthetic Control	6	300	300	60
CBF	3	30	900	128
Two Patterns	4	1000	4000	128

In the testing stage, the target time series is predicted by the k-NN classifier DTWW, DDTW and LCS for each k value ($k \in \{1, 3, \cdots, 15\}$). For DTWW, the predicted result of each k value represents one vote, thus 8 votes are obtained. And the most votes decide the predicted class of DTWW. It is done for DDTW and LCS similarly. Then, the entry of the BKS table representing the predicted classes of DTWW, DDTW and LCS can be found. Accordingly, the final predicted class of the target is extracted from the BKS table.

4.3. Performance Comparison

The experiments are performed on a computer with Microsoft Windows 7 Enterprise operating system, Intel Xeon E5640 CPU and 2GB RAM. It spends 20853 seconds totally to obtain controlling parameters $\theta_\alpha = 0.15$, $\theta_\beta = 0.5$ and $\theta_\gamma = 0$ for all UCR datasets. In the experiments, we compare classification error rates of DTW [8], DTW with warping window (DTWW) [11, 12], derivative DTW (DDTW) [6], and our LCS with diversity algorithms. Furthermore, we apply the *behavior knowledge space* (BKS) method [13, 14] to build an ensemble classifier.

In the comparison table of error rate, the arithmetic and weighted averages are calculated as the performance summary of each algorithm. The formulas for calculating the *arithmetic average* and the *weighted average* [4, 27] are given in Equations 9 and 10, respectively.

$$\text{Arithmetic Average} = \frac{\sum_{i=1}^{47} E_i}{47}, \tag{9}$$

$$\text{Weighted Average} = \frac{\sum_{i=1}^{47} (E_i \times |T_i|)}{\sum_{i=1}^{47} |T_i|}, \tag{10}$$

where E_i and $|T_i|$, $1 \leq i \leq 47$ denote the error rate and the size of the testing set, respectively.

The summarized error rate and the execution time comparisons of various methods are shown in Tables 2 and 3, respectively. In the error rate comparison, one can find that the LCS similarity measurement with diversity is competitive with the DTW, the DTWW, and the DDTW algorithms. In addition, the accuracy of the BKS method outperforms DTWW, DDTW and LCS in most datasets. The improvement of error rate by arithmetic or weighted average is about 20% over the previously best-known result of DTWW. One can also find that the DTWW algorithm requires less execution time than other algorithms. Our LCS algorithm requires more execution time than DTWW but it requires less time than DTW and DDTW algorithm. In the view point of time complexity, the required time of DTW, DTWW and LCS algorithms are comparable because of their dynamic programming formulas.

5. CONCLUSION

In this paper, we focus on handling the time series classification problem. We propose the LCS similarity measurement with diversity and a training approach to obtain better matching threshold for the LCS algorithm. The LCS algorithm with diversity is used to measure the distance between two time series. The training strategy provides the suitable thresholds for the LCS algorithm (or LCS-like algorithms). The matching threshold is used to determine whether a pair of real numbers is regarded as a match in the LCS-like algorithm. In the experimental results, the LCS similarity measurement with diversity is competitive to the DTW, the DTWW, and the DDTW algorithms. Furthermore, we build the BKS ensemble to improve the accuracies by combining the results of the DTWW, the DDTW, and the LCS algorithms. The performances of the BKS ensemble are better than every individual classifier in most datasets. The error improvement is about 20% for arithmetic and weighted averages over the previously best-known result of DTWW. In the future, it is worthy to develop more efficient training strategy or to find better characteristic for the LCS similarity measurement in time series classification.

ACKNOWLEDGMENTS

This research work was partially supported by the National Science Council of Taiwan under contract NSC 102-2221-E-110-033-MY2.

REFERENCES

[1] K. Buza, A. Nanopoulos, and L. Schmidt-Thieme, "Time-series classification based on individualised error prediction," in *Proceedings of IEEE 13th International Conference on Computational Science and Engineering (CSE)*, Dec. 2010, pp. 48–54.

[2] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 349–371, Oct. 2003.

[3] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, Pittsburgh, Pennsylvania, USA, 2006, pp. 1033–1040.

[4] C.-J. Hsu, K.-S. Huang, C.-B. Yang, and Y.-P. Guo, "Flexible dynamic time warping for time series clas-

Table 2: The error rates of various algorithms. Each entry marked with a red underline means the lowest error rate among all algorithms, and the blue boldface values represent the lowest error rates among them excluding BKS.

Dataset \ Algorithm	DTW	DTWW	DDTW	LCS	BKS
Image Outline (15)					
OSU Leaf	0.3636	0.3678	**0.1157**	0.2273	0.1116
SwedishLeaf	0.2096	0.1376	**0.1136**	0.1216	0.0672
50Words	0.2835	0.2242	0.3033	**0.2044**	0.1011
FaceFour	0.1591	0.0909	0.3750	**0.0795**	0.0227
Adiac	0.4118	0.4092	**0.3811**	0.4808	0.1688
Fish	0.1371	0.1314	0.1029	**0.0800**	0.0057
Car	0.2500	0.2500	0.2667	**0.1667**	0.0667
DiatomSizeReduction	**0.0392**	0.0719	0.0621	0.0490	0.1569
FacesUCR	0.0659	0.0620	0.1478	**0.0507**	0.0366
MedicalImages	0.2461	**0.2382**	0.3368	0.3211	0.2039
Symbols	0.0472	0.0573	**0.0261**	0.0643	0.2955
WordsSynonyms	0.3245	**0.2461**	0.3135	0.2492	0.2116
FaceAll	0.2284	0.2148	**0.1260**	0.2142	0.0225
Yoga	0.1613	**0.1550**	0.1797	0.1831	0.2117
Two Patterns	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Motion (9)					
Gun-Point	0.1200	0.0467	**0.0067**	0.0200	0.0200
CricketX	0.2282	**0.2231**	0.3615	0.2769	0.1769
CricketY	0.2513	**0.2513**	0.4385	0.2641	0.1359
CricketZ	**0.2128**	0.2385	0.4462	0.2564	0.1154
Haptics	0.6364	0.6331	0.7305	**0.6299**	0.4578
InlineSkate	0.6255	0.6164	**0.5600**	0.5909	0.4691
uWaveX	0.2691	**0.2250**	0.3236	0.2328	0.1689
uWaveY	0.3554	**0.3004**	0.4132	0.3113	0.2462
uWaveZ	0.3409	0.3222	0.4207	**0.3135**	0.2390
Sensor Reading (20)					
Trace	0.0100	0.0200	**0.0000**	0.0500	0.0000
Lightning2	0.1967	**0.1475**	0.3279	0.2131	0.2787
Lightning7	0.2329	**0.2192**	0.4247	0.2603	0.2603
ECG	0.2000	0.1100	0.1600	**0.1000**	0.1400
Beef	0.4333	0.3667	**0.3333**	**0.3333**	0.3333
Coffee	**0.0357**	**0.0357**	0.0714	0.1071	0.0000
OliveOil	0.1667	0.1667	**0.1333**	0.1667	0.1000
ECGFiveDays	0.2253	0.2149	0.3182	**0.1556**	0.3171
ItalyPowerDemand	0.0544	**0.0379**	0.0845	0.0933	0.0369
MoteStrain	**0.1094**	0.1342	0.2819	0.1102	0.0807
SonyAIBORobotII	0.1574	0.1322	**0.1259**	0.1941	0.1784
SonyAIBORobot	0.2879	0.3128	**0.2612**	0.3544	0.4276
TwoLeadECG	0.0674	0.1203	**0.0053**	0.0860	0.0053
Wafer	0.0177	**0.0050**	0.0248	0.0096	0.0084
CinCECG	0.3094	**0.0449**	0.2870	0.1964	0.2065
ChlorineConcentration	0.3734	0.3727	**0.3076**	0.3617	0.4190
MALLAT	0.0857	**0.0755**	0.1028	0.1552	0.0482
StarLightCurves	0.1130	0.1136	**0.0374**	0.1326	0.0243
ECGThorax1	0.2280	**0.1985**	0.3003	0.2529	0.0875
ECGThorax2	0.1486	**0.1389**	0.1674	0.1532	0.0631
Simulation (3)					
Synthetic Control	**0.0133**	0.0233	0.4400	0.0667	0.0100
CBF	**0.0000**	0.0011	0.4000	0.0033	0.0311
Two Patterns	**0.0000**	0.0013	0.0020	**0.0000**	0.0000
Arithmetic Average	0.2007	**0.1810**	0.2372	0.1903	0.1440
Weighted Average	0.1779	**0.1604**	0.1932	0.1741	0.1289

Table 3: The running time (seconds) of various algorithms.

Dataset \ Algorithm	DTW	DTWW	DDTW	LCS
Image Outline (15)				
OSU Leaf	307.01	102.68	747.26	175.13
SwedishLeaf	190.01	42.98	432.86	114.58
50Words	490.59	143.10	1236.76	255.33
FaceFour	10.56	2.51	24.15	5.82
Adiac	163.61	44.65	402.23	103.77
Fish	227.53	59.56	569.12	130.35
Car	35.46	12.57	96.72	20.25
DiatomSizeReduction	23.43	6.36	53.26	14.32
FacesUCR	307.35	145.85	630.21	192.05
MedicalImages	119.54	74.55	255.20	76.63
Symbols	149.43	68.98	368.57	97.24
WordsSynonyms	407.83	181.21	1048.17	222.38
FaceAll	716.86	149.87	1512.60	341.53
Yoga	4542.64	1332.79	12600.20	3042.69
Plane	11.83	7.10	25.85	9.47
Motion (9)				
Gun-Point	6.21	1.44	14.96	3.96
CricketX	493.18	181.66	1224.69	256.72
CricketY	465.46	251.88	1190.68	247.61
CricketZ	480.80	261.22	1173.00	247.53
Haptics	1667.70	574.13	4461.88	1157.62
InlineSkate	5778.11	3337.02	15928.70	3786.86
uWaveX	8835.02	2759.61	24089.20	6032.89
uWaveY	8856.77	2554.42	23851.80	6031.28
uWaveZ	8904.60	2905.14	23542.90	6067.77
Sensor Reading (20)				
Trace	27.58	6.57	68.16	13.20
Lightning2	54.07	16.46	123.51	28.33
Lightning7	18.99	5.43	45.91	10.36
ECG	4.59	1.37	9.08	3.17
Beef	7.21	1.65	17.91	4.23
Coffee	2.79	0.80	5.90	1.81
OliveOil	10.87	2.84	25.37	6.66
ECGFiveDays	25.65	12.03	44.07	17.47
ItalyPowerDemand	16.90	18.69	18.47	20.03
MoteStrain	33.20	26.54	40.08	29.84
SonyAIBORobotII	25.60	21.47	30.20	25.63
SonyAIBORobot	16.46	14.59	19.30	16.80
TwoLeadECG	35.01	33.76	51.54	35.96
Wafer	4307.72	1272.03	13849.00	2857.36
CinCECG	4375.75	1259.80	11619.20	2712.97
ChlorineConcentration	1965.21	319.46	4328.62	1046.14
MALLAT	3762.18	727.89	10673.70	2507.11
StarLightCurves	147465.00	67125.40	450870.00	60727.60
ECGThorax1	33034.90	4691.74	103849.00	15220.60
ECGThorax2	32848.60	4688.20	103805.00	14532.70
Simulation (3)				
Synthetic Control	15.76	5.62	30.72	10.97
CBF	21.93	10.95	43.87	16.52
Two Patterns	2570.23	713.97	5800.93	1364.29
Total	273837.73	96178.54	820850.51	129843.53

sification," in *Procedia Computer Science 51, International Conference On Computational Science, ICCS 2015*, Reykjavík, Iceland, June 2015, pp. 2838–2842.

[5] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognition*, vol. 44, no. 9, pp. 2231–2240, 2011.

[6] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *Proceedings of the First SIAM International Conference on Data Mining*, vol. 1, Chicago, IL, USA, Apr. 2001, pp. 5–7.

[7] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, China, Aug. 2012, pp. 262–270.

[8] L. Gupta, D. L. Molfese, R. Tammana, and P. G. Simos, "Nonlinear alignment and averaging for estimating the evoked potential," *IEEE Transactions on Biomedical Engineering*, vol. 43, no. 4, pp. 348–356, Apr. 1996.

[9] Y. Chen, B. Hu, E. Keogh, and G. E. Batista, "DTW-D: Time series semi-supervised learning from a single example," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, IL, USA, Aug. 2013, pp. 383–391.

[10] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana, "The UCR time series classification/clustering homepage," http://www.cs.ucr.edu/~eamonn/time_series_data/, 2011.

[11] C. A. Ratanamahatana and E. Keogh, "Making time-series classification more accurate using learned constraints," in *Proceedings of the Fourth SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, USA, Apr. 2004, pp. 11–22.

[12] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[13] C.-Y. Hor, "Machine learning approaches for the protein and RNA sequence analysis," Ph.D. dissertation, Ph. D. Dissertation, Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, 2014.

[14] Š. Raudys and F. Roli, "The behavior knowledge space fusion method: Analysis of generalization error and strategies for performance improvement," in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, T. Windeatt and F. Roli, Eds. Springer Berlin Heidelberg, 2003, vol. 2709, pp. 55–64.

[15] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 23, no. 1, pp. 67–72, Feb. 1975.

[16] H.-Y. Ann, C.-B. Yang, C.-T. Tseng, and C.-Y. Hor, "A fast and simple algorithm for computing the longest common subsequence of run-length encoded strings," *Information Processing Letters*, vol. 108, pp. 360–364, 2008.

[17] K.-Y. Cheng, K.-S. Huang, and C.-B. Yang, "The longest common subsequence problem with the gapped constraint," in *Proceedings of the 30th Workshop on Combinatorial Mathematics and Computation Theory*, Hualien, Taiwan, 2013, pp. 80–85.

[18] D. S. Hirschberg, "A linear space algorithm for computing maximal common subsequences," *Communications of the ACM*, vol. 18, pp. 341–343, 1975.

[19] J. W. Hunt and T. G. Szymanski, "A fast algorithm for computing longest common subsequences," *Communications of the ACM*, vol. 20, no. 5, pp. 350–353, 1977.

[20] Y.-H. Peng and C.-B. Yang, "Finding the gapped longest common subsequence by incremental suffix maximum queries," *Information and Computation*, vol. 237, pp. 95–100, Oct. 2014.

[21] Y.-H. Peng, C.-B. Yang, K.-S. Huang, C.-T. Tseng, and C.-Y. Hor, "Efficient sparse dynamic programming for the merged LCS problem with block constraints," *International Journal of Innovative Computing, Information and Control*, vol. 6, pp. 1935–1947, 2010.

[22] T. Górecki, "Using derivatives in a longest common subsequence dissimilarity measure for time series classification," *Pattern Recognition Letters*, vol. 45, no. 0, pp. 99–105, 2014.

[23] A. Kuzmanic and V. Zanchi, "Hand shape classification using DTW and LCSS as similarity measures for vision-based gesture recognition system," in *EUROCON, 2007. The International Conference on Computer as a Tool*, Warsaw, Poland, Sept. 2007, pp. 264–269.

[24] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proceedings of 18th International Conference on Data Engineering*, Washington, DC, USA, 2002, pp. 673–684.

[25] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, vol. 2, Madison, Wisconsin, USA, 2003, pp. 521–527.

[26] A. C. Jalba, M. H. Wilkinson, J. B. Roerdink, M. M. Bayer, and S. Juggins, "Automatic diatom identification using contour analysis by morphological curvature scale spaces," *Machine Vision and Applications*, vol. 16, no. 4, pp. 217–228, 2005.

[27] J. Medhi, *Statistical methods: an introductory text.* New Age International, 1992.

[28] D. W. Mitchell, "More on spreads and non-arithmetic means," *The Mathematical Gazette*, pp. 142–144, 2004.

[29] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with COTE: the collective of transformation-based ensembles," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2522–2535, 2015.