TIME SERIES CLASSIFICATION WITH DISSIMILARITY SPACE*

[1]*Shu-Kai Hung*, [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 University of Science and Technology

ABSTRACT

The time series classification problem is aimed to classify a queried time series to a suitable class and then to assign this time series a corresponding class label. In this paper, we first divide a set of time series in the training set into several groups by the clustering method, and then build one delegate for each group with the DTW barycenter averaging method. For constructing the features for support vector machine (SVM) classifiers, we invoke the concept of dissimilarity space with seven distance functions. In addition, to improve the classification accuracy, we use the behavior knowledge space (BKS) method to construct ensemble classifiers, each of which is constituted by three of the seven SVM classifiers. The experimental datasets come from the UCR time series classification/clustering web site. As the experimental results show, with only one SVM classifier, it gets about 3% accuracy improvement over the result of Jain and Stephan in 2017, which also uses the dissimilarity space. The ensemble classifiers have further improvement in the classification accuracy.
Keywords: Time series; Dynamic time warping; Dissimilarity space; SVM; Ensemble.

1. INTRODUCTION

The time series classification problem is aimed to classify a queried time series into a one class in a set of times series that have been labelled. The time series classification has a variety of applications, such as speech recognition, medical signal analysis, and recognition of gesture [1]. In addition, it can also be applied to various domains, including biometrics, finance, medicine, networking, industry, chemistry, astronomy and robotics [2, 3].

The distance measurement for time series classification is usually the *lock-step measurement* [4], such as the *Euclidean distance* (ED) [5], or the *elastic measurement*, such as the *dynamic time warping* (DTW) [6]. It seems that the classification accuracy with the DTW distance (or DTW-like distance) is higher than other measurements. The LCS-like (*longest common subsequence* [7, 8]) algorithms, including the *variable gap longest common subsequence* (VGLCS) [9] and *longest common subsequence with at least length* k (LCSk) [10], can also be used to compute the similarity of two time series.

The *dissimilarity space* approach, proposed by Jain and Spiegel [11], offers a way to combine the advantages of the DTW distance with the feature-based method. The main idea is to select a set of k_p prototypes as delegates of time series by using the prototype concept proposed by Pekalska *et al.* [12] The dissimilarity representation of a time series consists of k_p features, denoted by a vector $\mathbb{R}^{k_p}$, where each feature represents its DTW distance from one of the k_p prototypes. The collection of these k_p features can be used in the feature-based classification. In this paper, we consider seven measurements as the dissimilarity space features, including DTW [6], DDTW [13], DTWW [14], LCS [7], LCSk [15], VGLCS [9] and ED [5].

The prototype selection is also an important issue for the feature-based classification. This paper combines the *dynamic time warping barycenter averaging* (DBA) [16] and *K-mean* [17] to construct the prototype set. After building the classifiers with the seven various distance measurements, we utilize the *behavior knowledge space* (BKS) [18, 19] to build classifier ensembles for improving the classification accuracy.

In this paper, the experimental datasets were downloaded from the classification/clustering web site [20]. We divide the datasets into two parts, part I and part II, according to their size roughly. We compare the results of Guo *et al.* [21] and the dissimilarity space with the DTW method proposed by Jain and Stephan [11]. Our methods outperform the methods of Guo *et al.* in DTW, DDTW and DTWW, but not in LCS-like methods. The possible reason is that our algorithms uti-

*This research work was partially supported by the Ministry of Science and Technology of Taiwan under contract MOST 104-2221-E-110-018-MY3.

lizes the DTW-based method (not LCS-based method), DBA, to select prototype. To improve classification accuracy, we further apply the *behavior knowledge space* (BKS) method [18, 19] to combine several classifiers into an ensemble classifier. Among many possible combinations for the BKS, the combinations of DTWW, LCS, and ED have higher classification accuracy, because they have more diversity.

The organization of this paper is as follows. In Section 2, we present the background knowledge. In Section 3, the algorithm is proposed for classifying the time series. Section 4 shows the experimental results and performance comparison. Finally, Section 5 gives the conclusion.

2. PRELIMINARIES

2.1. Measurements between Sequences

A time series of length m is represented by an ordered sequence $X = x_1 x_2 \cdots x_m$ where each real value $x_i \in \mathbb{R}$, $1 \leq i \leq m$, is sampled at a discrete point of time [1]. Given two time series with same length, $X = x_1 x_2 \cdots x_m$, $Y = y_1 y_2 \cdots y_n$ and $m = n$, the distance calculation of the *Euclidean distance* (ED) is given by $ED(X,Y) = \sum_{i=1}^{m} (x_i - y_i)^2$.

In general, the lengths of two time series may not be identical. The *dynamic time warping* (DTW) [6] is a famous distance measurement for two time series. Let $X_{i..j} = x_i x_2 \cdots x_j$. Let $DTW(i,j)$ denote the DTW distance of $X_{1..i}$ and $Y_{1..j}$, and $d(x_i, y_j)$ denote the distance of two data points x_i and y_j. The *dynamic programming* (DP) formula for calculating $DTW(i,j)$ is shown in Equation 1.

The *derivative dynamic time warping* (DDTW) algorithm [13] is a modified algorithm from the DTW algorithm [6]. The DDTW algorithm utilizes the derivative to calculate the trend (such as increase, decrease, peak and valley) of data point x_i by Equation 2.

$$f(x_i) = \frac{(x_i - x_{i-1}) + (x_{i+1} - x_{i-1})/2}{2}. \qquad (2)$$

The *DTW distance with warping window* (DTWW) [14] is also a modified algorithm from the DTW [6] algorithm. It considers additional warping path constraints. The path of DTWW alignment tends to drift close to the diagonal by the path constraints. Suppose that the warping window size is r. By Equation 1, if $|i-j| > r$, we set $d(x_i, y_j) = \infty$. Otherwise, it performs the same operation of DTW in Equation 1.

The LCS problem has been studied for serval decades. The LCS between two sequences represents their similarity [7]. The *LCS with at least length k* (LCSk) problem was introduced by Benson *et al.* [15]. It measures the similarity by ignoring the matched subsequence of a small fragment. Each character in the answer of the LCSk problem must be a part of a common substring with at least k_c matches of both A and B. The LCSk problem is a generalized version of the traditional LCS problem, in which $k_c = 1$. The *variable gap LCS* (VGLCS) problem is a variant of the LCS problem with additional variable gapped constraints [9]. The VGLCS problem not only finds the LCS of two sequences but also considers the gap constraints between two neighboring matching characters. The DP algorithm takes $O(m^2 n^2)$ time for solving the VGLCS problem. An improved $O(mn)$ algorithm was proposed by Peng and Yang [9].

In LCS-like algorithms, it is necessary to determine match or mismatch by a threshold θ between two data points of real numbers in time series. If the absolute difference value of two data points is not greater than θ, we regard that the two data points is a match.

2.2. DTW Barycenter Averaging and Dissimilarity Space

The *DTW barycenter averaging* (DBA) method was proposed by Petitjean *et al.* [16]. It takes accounts of several warping paths to obtain an average time series as the delegate of the several time series. The DBA method outperforms all existing multiple alignment techniques on all UCR datasets [16]. Hence, DBA is a good method to find a delegate by averaging several time series for DTW.

For example, suppose that there are three time series $X_1 = \{1,3,4,3\}$, $X_2 = \{2,1,3,2\}$, and $X_3 = \{1,4,3,2\}$. It randomly selects one of X_i as the seed of the delegate, says that $X_2 = \{2, 1, 3, 2\}$ is selected. By aligning each X_i to X_2 with the DTW method, we three warping paths, representing by vectors $\langle 1, 1, (3,4), 3 \rangle$, $\langle 2, 1, 3, 2 \rangle$, and $\langle 1, 1, (3,4), 2 \rangle$. By averaging the corresponding values of each position, we get a new vector $\langle 1.33, 1, 3.4, 2.33 \rangle$, which is the delegate of time series X_1, X_2, and X_3.

The *dissimilarity space* (D-space) was proposed by Jain and Stephen [11]. Let T be the training dataset and P be the prototype set of k_p delegate time series selected from T, represented as follows.

$$P = \{P_1, P_2, \cdots, P_{k_p}\} \subseteq T \qquad (3)$$

Let $d(X,Y)$ be a distance function, which returns the distance of two time series X and Y. Let $\phi(X)$ be a k_p-dimensional vector in the D-space, where each element represents the dissimilarity (distance) between

$$DTW(i,j) = \begin{cases} 0 & \text{if } i = 0 \text{ and } j = 0, \\ \infty & \text{if } i = 0 \text{ and } j \neq 0, \\ \infty & \text{if } i \neq 0 \text{ and } j = 0, \\ d(x_i, y_j) + \min \begin{cases} DTW(i-1, j) \\ DTW(i, j-1) \\ DTW(i-1, j-1) \end{cases} & \text{otherwise.} \end{cases} \quad (1)$$

X and a prototype P_i, expressed as follows.

$$\phi(X) = \langle (d(X, P_1), d(X, P_2), \cdots d(X, P_i) \cdots, d(X, P_{k_p}) \rangle. \quad (4)$$

In addition, $DS(X, d(\cdot))$ is used to denote the k_p-dimensional vector of features in the D-space by the distance function $d(\cdot)$.

The classifier learning in the D-space is composed of three steps as follows.

1. Select a set P of prototypes from the training set T by the prototype selection algorithm, such as *RandomC* [12] or K-central [22].

2. Use the prototype set P to construct the D-space of each sequence in T.

3. Build a classifier or ensemble classifier for the D-space.

2.3. Support Vector Machine and Behavior Knowledge Space

The *support vector machine* (SVM) [23] has been widely used in classification. It is convenient to use the features or characteristics to perform the appropriate classification by SVM. When good features are given, the SVM usually obtains good classification with high accuracy. According to our preliminary experiments, we observe that the characteristic of dissimilarity space is highly suitable for SVM. Therefore, this paper applies the SVM to perform the classification of time series. Chang and Lin proposed the *LIBSVM* library [24], which is a famous programming package of SVM [23]. We use the LIBSVM to implement our algorithm.

For improving the classification accuracy, we use the *behavior knowledge space* (BKS) method to integrate multiple classifiers to build ensemble classifiers [18, 19]. Based on the classification results of q classifiers for a training dataset T with $|C|$ class labels, the BKS table of $|C|^q$ possible entries can be constructed. In the table, each entry indicates a possible combination of predicted labels by these q classifiers. In the testing stage, the query time series Y is also predicted by the q classifiers, then a corresponding entry is obtained, and finally the predicted label is got by the entry with the majority vote.

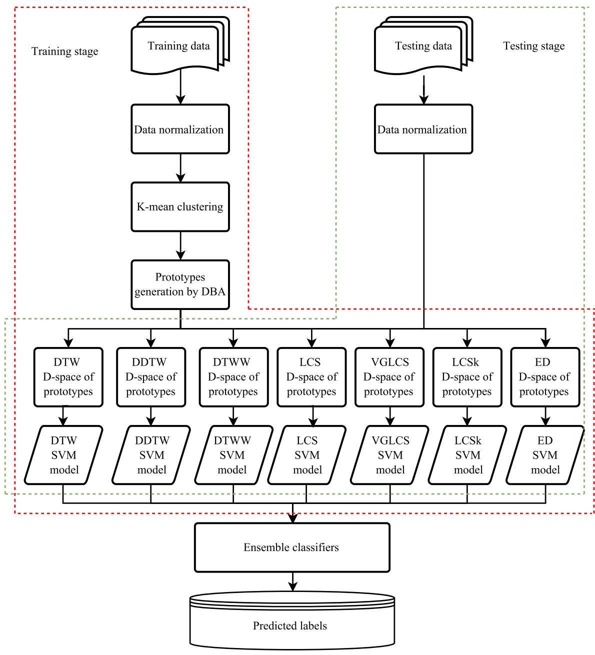

Figure 1: The flow chart of our algorithm.

3. THE PROPOSED ALGORITHM

The flow chart of our algorithm for solving the classification problem on time series is shown in Figure 1.

Our training algorithm for the training set is presented as follows.

1. Use the Z-score method to normalize the time series in the training dataset T [25]. It converts a data point z in time series into z' as follows.

$$z' = \frac{z - \mu}{\sigma}, \quad (5)$$

where μ is the mean of all points in the time series and σ, $\sigma \neq 0$, is the standard deviation of the time series. If $\sigma = 0$, the values of all data points are identical and their normalized values become 0 after the Z-score normalization. We denote the normalized training dataset as T'.

2. Perform the K-mean clustering method [26] on T' to produce k_p clusters for each class in T'. Now there are totally $k_p \times |C|$ clusters for T'.

3. Perform the DTW barycenter averaging (DBA) method to generate the delegate (prototype) of each cluster. It produces k_p prototypes for each class. Thus, there are $k_p \times |C|$ DBA prototypes for T'.

4. For each $T_i \in T'$, compute the D-space vector $DS(T_i, d(\cdot))$ with the $k_p \times |C|$ prototypes, where the vector is of size $k_p \times |C|$ and $d(\cdot)$ may be any one of the seven distance functions (such as DTW, LCS, and so on).

5. Use all $DS(T_i, d(\cdot))$, $T_i \in T'$ as the feature vectors to build the SVM classifiers. Here, seven SVM classifiers are obtained, since there are seven possible distance functions for $d(\cdot)$.

6. Utilize every three of the seven SVM classifiers to build one BKS table to construct an ensemble classifier. There are $\binom{7}{3} = 35$ combinations, thus 35 ensemble classifiers are got.

In the above algorithm, $d(\cdot)$ is one of the seven distance functions, including DTW [6], DDTW [13], DTWW [14], LCS [7], LCSk [15], VGLCS [9] and ED[5].

4. EXPERIMENTAL RESULTS

In our experiments, the datasets of time series were obtained from the UCR time series classification/clustering web site [20], which collected 85 datasets in August 2015. We used the 47 datasets collected before August 2015, and divided them into two parts: part I (21 datasets) and part II (26 datasets), based on their sizes roughly. Each dataset has been partitioned into one training set and one testing set [20]. Our experiments were performed on the computer, containing the Intel Core I5-3470 CPU and 8GB RAM, and running the operating system Microsoft Windows 7 Enterprise.

In the experiments on part I, we first train the parameters including k_p, θ, k_c, and w, for D-space, the threshold of the LCS and VGLCS, the length of consecutive matches in LCSk, and the warping window ratio of DTWW, respectively. The ranges of these parameters for training and their best values are shown in Table 1.

After getting the best values of these four parameters, we test the datasets of part I. We compare classification accuracies of the seven classifiers with the five classifiers of Guo *et al.* [21] in Table 2. The *arithmetic average accuracy* and the *weighted average accuracy* in Table 2 are calculated by Equations 6 and 7, respectively.

$$\text{Arithmetic average accuracy} = \frac{\sum_{i=1}^{|D|} acc_i}{|D|} \times 100\%, \quad (6)$$

$$\text{Weighted average accuracy} = \frac{\sum_{i=1}^{|D|} (acc_i \times |D_i|)}{\sum_{i=1}^{|D|} |D_i|} \times 100\%, \quad (7)$$

where acc_i, $|D_i|$ and $|D|$ denote the accuracy of each dataset, the size of the testing set and the number of datasets, respectively.

In Table 2, our method outperforms the results of Guo *et al.* with DTW, DTWW, and DDTW, but not with LCS and VGLCS. The possible reason is that the DBA prototypes are generated based on the DTW alignment, rather than the LCS alignment. The best accuracy 86.5% is obtained by our method with DTWW, and it has about 2% improvement against the DTWW result of Guo *et al.*

After the seven SVM classifiers on part I have been established, in order to improve the accuracy of classification prediction, we utilize the BKS method [18, 19] to build an ensemble classifier with every three SVM classifiers. Since there are seven SVM classifiers, $\binom{7}{3} = 35$ combinations are obtained for building ensemble classifiers. Thus, 35 ensemble classifiers are constituted. From the top three ensemble classifiers of the 35 ensemble classifiers, we select the five best SVM classifiers, which are DTW, DDTW, DTWW, LCS and ED.

The accuracies of the top three ensemble classifiers on each dataset of part I are shown in Table 3. As we can see, the classification accuracies of the ensemble classifiers is higher than those of the SVM classifiers.

In the experiments of part II, we only perform the above five best SVM classifiers with the best values of parameters K_p, θ, K_c and w. Table 4 shows the accuracy comparison of our five best classifiers and the four classifiers of Guo *et al.* in part II.

We use every three of the five best SVM classifiers (DTW, DDTW, DTWW, LCS and ED) to construct ensemble classifiers for part II. There are $\binom{5}{3} = 10$ combinations of ensemble classifiers. The accuracies of the top three ensemble classifiers on each dataset of part II are shown in Tables 5.

We observe that the best two combinations in part I, (DTWW, LCS, ED) and (DTW, LCS, ED), are also the best in part II. The distance calculation methods

Table 1: The best values of the parameters.

	K_p of D-space	θ of LCS and VGLCS	K_c of LCSk	w of DTWW
Possible values	2, 3, 4, 5, 6, 7, 8, 9, 10	0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5	2, 3, 4, 5, 6, 7, 8	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
Best value	6	0.15	4	0.5

Table 2: The accuracy (in percentage) comparison of our seven classifiers and the five classifiers of Guo *et al.* in part I. The red number indicates the better one under the same measurement between ours and the classifier of Guo *et al.* The underline number indicates the best one in each dataset.

Dataset \ Method	DTW ours	DDTW ours	DTWW ours	LCS ours	VGLCS ours	LCSK ours	ED ours	DTW Guo's	DDTW Guo's	DTWW Guo's	LCS Guo's	VGLCS Guo's
Synthetic control	98.3	**87.6**	**98.3**	88.7	93.0	85.3	95.0	**99.3**	56.0	97.7	**93.3**	**94.4**
Gun-Point	**92.7**	96.0	92.7	92.7	88.0	92.7	96.0	91.7	**99.4**	**93.4**	**98.0**	**98.0**
CBF	96.6	**91.7**	96.6	93.8	89.8	90.2	96.8	**99.3**	60.0	**99.9**	**99.3**	**99.7**
FaceAll	78.4	82.7	78.4	72.8	**78.7**	51.7	73.1	**80.8**	**87.4**	**78.6**	**78.6**	**78.7**
OSULeaf	**66.1**	72.3	**66.2**	65.2	67.8	66.9	54.1	63.7	**88.5**	63.3	**78.3**	**77.3**
SwedishLeaf	**93.4**	**93.3**	**93.4**	**94.2**	**92.1**	92.5	90.9	79.2	88.7	86.3	87.9	87.9
50Words	**73.7**	65.9	74.7	75.8	77.8	72.0	68.5	69.0	**69.7**	**77.6**	**79.6**	**79.2**
Trace	**100**	99.0	**100**	**98.0**	93.0	96.0	94.0	**100**	**100**	98.0	95.0	**95.0**
Two Patterns	**100**	99.4	**100**	96.2	88.4	88.9	90.9	**100**	**99.8**	99.9	**100**	**100**
Wafer	98.4	**95.4**	98.4	97.4	98.2	97.2	99.4	**99.5**	91.6	**99.5**	**99.1**	-
FaceFour	**94.3**	**86.1**	**90.9**	88.6	**95.5**	82.9	70.4	83.0	66.5	**90.9**	**92.1**	92.1
Lighting2	83.1	67.2	**89.2**	73.7	70.5	68.8	62.2	**87.0**	**67.3**	85.3	**78.7**	**80.4**
Lighting7	**81.4**	**65.3**	**82.3**	72.6	69.3	61.6	68.4	72.6	51.6	78.1	**74.0**	**75.4**
ECG200	**77.0**	81.0	77.0	76.0	79.0	78.0	80.0	**77.0**	**84.0**	**89.0**	**90.0**	**91.0**
Adiac	**67.7**	**65.4**	**72.0**	**71.3**	**68.5**	71.8	69.9	60.4	61.9	59.1	52.0	52.0
Yoga	**86.4**	**87.6**	**86.8**	**83.6**	78.4	83.3	78.2	84.5	82.1	84.5	81.7	-
Fish	75.4	86.8	**80**	81.7	88.5	84.0	74.8	**82.3**	**89.8**	76.9	**92.0**	-
Car	**100**	**100**	**100**	**100**	**100**	**100**	**100**	82.3	83.4	75.0	83.4	91.5
Beef	57.3	54.3	57.7	55.8	60.0	43.3	46.7	**63.3**	**67.0**	**63.4**	**66.7**	**66.7**
Coffee	97.2	89.2	95.3	71.4	75.0	88.2	92.8	**100**	**92.9**	**96.5**	**90.0**	**89.3**
OliveOil	**86.7**	73.3	**86.7**	70.0	**86.1**	70.0	76.7	83.3	**86.7**	83.4	**83.3**	83.4
Arithmetic average	**85.9**	**82.8**	**86.5**	81.9	82.7	79.3	79.9	83.8	79.7	84.6	**85.4**	**85.1**
Weighted average	**92.5**	**91.4**	**92.7**	90.1	88.3	86.1	88.5	92.1	87.7	92.3	**92.0**	**90.3**

Table 3: The classification accuracies (in percentage) of the top three ensemble classifiers on part I.

Dataset \ Ensemble	Accuracy: DTWW LCS ED	Accuracy: DTW LCS ED	Accuracy: DTW DDTW ED
Synthetic control	98.3	98.3	97.3
Gun-Point	97.7	97.7	97.7
CBF	96.6	96.6	96.6
FaceAll	79.1	79.1	76.2
OSULeaf	74.9	69.9	73.1
SwedishLeaf	94.2	94.2	93.7
50Words	76.8	76.4	78.4
Trace	100	100	100
Two Patterns	99.5	99.5	99.5
Wafer	98.4	98.4	98.4
FaceFour	90.9	91.3	90.9
Lighting2	78.1	77.9	77.0
Lighting7	79.5	79.3	79.5
ECG200	76.0	76.0	81.0
Adiac	79.3	78.9	66.3
Yoga	83.6	83.6	79.8
Fish	90.7	89.7	86.3
Car	100	100	100
Beef	61.3	59.3	61.7
Coffee	95.3	96.4	92.3
OliveOil	86.7	86.7	86.7
Arithmetic average	87.3	87.1	86.3
Weighted average	91.9	92.5	91.4

Table 4: The accuracy (in percentage) comparison of our five best classifiers and the four classifiers of Guo *et al.* in part II. The red number indicates the better one under the same measurement between ours and the classifier of Guo *et al.* The underline number indicates the best one in each dataset.

Dataset \ Method	DTW ours	DDTW ours	DTWW ours	LCS ours	ED ours	DTW Guo's	DDTW Guo's	DTWW Guo's	LCS Guo's
CinC ECG torso	**78.9**	78.2	91.3	63.0	83.9	69.1	**81.3**	**95.6**	**80.4**
ChlorineConcentration	60.0	57.0	**63.4**	55.6	61.2	**62.7**	**69.3**	62.8	**63.9**
DiatomSizeReduction	90.8	83.3	**93.1**	89.2	88.2	**96.1**	**93.8**	92.9	**95.1**
ECGFiveDays	65.3	**71.2**	77.7	**81.5**	91.0	**77.5**	68.2	**78.6**	**84.5**
FacesUCR	**89.7**	57.0	**63.4**	55.6	61.2	62.7	**69.3**	62.8	**63.9**
Haptics	**45.6**	**33.4**	**45.6**	**44.8**	42.5	36.7	27.0	36.7	37.1
InlineSkate	**49.9**	41.0	**44.9**	33.0	29.8	38.4	**44.0**	38.4	**40.0**
ItalyPowerDemand	91.5	**92.7**	**92.6**	87.9	96.7	**94.6**	91.6	89.8	**92.5**
MALLAT	**96.2**	**92.1**	96.2	87.9	96.7	94.6	91.6	**96.3**	**90.7**
MedicalImages	**76.3**	**70.5**	**76.3**	**69.3**	73.2	75.4	66.4	76.2	67.9
MoteStrain	**92.6**	63.0	**92.6**	**91.8**	83.1	89.6	**71.9**	76.6	89.0
SonyAIBORobotSurfaceII	81.8	83.0	**81.8**	79.0	85.3	**84.3**	**87.5**	76.8	**80.6**
SonyAIBORobotSurface	69.6	**89.7**	**69.6**	52.0	71.2	**71.3**	83.9	68.8	**64.6**
Plane	99.0	**100**	99.0	**99.0**	99.0	**100**	**100**	**100**	**99.0**
StarLightCurves	**94.8**	**97.4**	**94.8**	**95.7**	93.9	88.7	92.8	88.7	86.8
Symbol	**97.7**	92.5	**97.7**	93.0	80.9	95.3	**97.4**	94.3	**93.6**
TwoLeadECG	**94.1**	**99.7**	**94.1**	**94.5**	81.4	93.3	99.5	88.0	91.4
WordsSynonyms	68.1	58.1	74.2	62.0	63.4	**72.9**	**68.7**	**75.4**	**75.1**
Cricket_X	**77.4**	**64.2**	77.4	60.8	54.1	77.2	63.9	**77.7**	**72.4**
Cricket_Y	**79.5**	**66.8**	**79.5**	68.7	57.4	74.9	56.2	74.9	**73.6**
Cricket_Z	**80.5**	**59.6**	**80.5**	63.4	59.0	78.8	55.4	76.2	**74.4**
uWaveGestureLibrary_X	**78.3**	**70.5**	**88.3**	**77.7**	76.7	77.5	67.7	87.5	76.8
uWaveGestureLibrary_Y	69.5	**62.9**	69.5	69.7	67.9	**70.0**	58.7	**70.0**	**68.9**
uWaveGestureLibrary_Z	**72.2**	**66.4**	**72.2**	**69.8**	70.6	67.8	58.0	67.8	68.7
NonInvasiveFatalECG_Thorax1	**99.8**	**85.8**	**99.8**	**99.8**	87.0	80.2	70.0	80.2	74.8
NonInvasiveFatalECG_Thorax2	**90.7**	**87.0**	**90.7**	**90.7**	90.6	86.2	83.3	86.2	84.7
Arithmetic average	**80.4**	**75.2**	**82.2**	76.0	75.3	78.6	74.3	79.0	**77.4**
Weighted average	**76.6**	**70.4**	**78.2**	70.1	70.4	70.1	68.6	73.7	**70.3**

for DTWW (or DTW), LCS, ED are very diverse. As a result, the ensemble classifier coming from the SVM classifiers with more diversity will get better accuracy. In summary, we suggest that the two ensemble classifiers, (DTWW, LCS, ED) and (DTW, LCS, ED), should be good candidates for doing the classification of UCR time series datasets.

We also compare our results with those of Jain and Stephan [11] in Table 6. They performed experiments on only 42 UCR datasets, as shown in the table. As one can see, our results have about 3% improvement against theirs. The method of Jain and Stephan [11] selects randomly time series as the prototypes, while we invoke the K-mean clustering method and DBA to generate prototypes for more representatives.

5. CONCLUSION

In this paper, we focus on handling the time series classification problem. We proposed a new algorithm with the dissimilarity space and use seven different distance measurements to construct seven SVM classifiers. In the experimental results, our method outperforms the method of Guo *et al.* with DTW, DDTW and DTWW distance functions, but loses to their method with LCS-like measurements. Furthermore, we use the BKS ensembles to improve accuracy, and find the best ensemble classifiers are the combinations of (DTWW, LCS, ED) and (DTW, LCS, ED). In addition, we also compare the classification prediction results with Jain and Stephan [11], which also used the dissimilarity space. We get about 3% improvement in accuracy.

REFERENCES

[1] J. D. Hamilton, *Time series analysis.* Princeton university press, Princeton, 1994, vol. 2.

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

[3] 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, 2003.

[4] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[5] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and image processing*, vol. 14, no. 3, pp. 227–248, 1980.

[6] R. Bellman and R. Kalaba, "On adaptive control

Table 5: The arithmetic average accuracies (in percentage) of the top three ensemble classifiers on part II.

Dataset \ Ensemble	Accuracy DTWW LCS ED	DTW LCS ED	DDTW DTWW ED
CinC_ECG_torso	94.1	89.3	93.8
ChlorineConcentration	63.4	60.0	65.2
DiatomSizeReduction	93.1	90.8	93.1
ECGFiveDays	81.5	79.9	65.3
FacesUCR	94.1	92.4	94.1
Haptics	49.5	49.5	46.6
InlineSkate	48.1	51.2	43.9
ItalyPowerDemand	92.6	91.5	92.6
MALLAT	96.2	96.2	96.2
MedicalImages	76.4	76.4	75.1
MoteStrain	94.7	94.7	92.8
SonyAIBORobotSurfaceII	81.8	81.8	83.0
SonyAIBORobotSurface	89.7	89.7	89.7
Plane	100	100	100
StarLightCurves	95.7	95.7	95.1
Symbol	94.8	94.8	92.5
TwoLeadECG	99.7	99.7	94.8
WordsSynonyms	74.2	74.2	74.2
Cricket_X	77.4	77.4	77.4
Cricket_Y	74.9	74.9	75.4
Cricket_Z	80.5	80.5	79.7
uWaveGestureLibrary_X	77.4	78.3	77.4
uWaveGestureLibrary_Y	80.5	80.5	79.7
uWaveGestureLibrary_Z	69.8	69.8	72.5
NonInvasiveFatalECG_Thorax1	99.8	99.8	99.8
NonInvasiveFatalECG_Thorax2	91.3	91.3	90.7
Arithmetic average	83.7	83.2	82.6
Weighted average	79.9	78.8	79.3

processes," *IRE Transactions on Automatic Control*, vol. 4, no. 2, pp. 1–9, 1959.

[7] D. S. Hirschberg, "Algorithms for the longest common subsequence problem," *Journal of the ACM (JACM)*, vol. 24, no. 4, pp. 664–675, 1977.

[8] C.-B. Yang and R. C. T. Lee, "Systolic algorithms for the longest common subsequence problem," *Journal of the Chinese Institute of Engineers*, vol. 10, no. 6, pp. 691–699, 1987.

[9] 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.

[10] Y. Ueki, Diptarama, M. Kurihara, Y. Matsuoka, K. Narisawa, R. Yoshinaka, H. Bannai, S. Inenaga, and A. Shinohara, "Longest common subsequence in at least k length order-isomorphic substrings," in *Proceedings of SOFSEM 2017, the 43rd International Conference on Current Trends in Theory and Practice of Computer Science, Limerick, Ireland, January 16-20, 2017*. Springer, 2017, pp. 363–374.

[11] B. J. Jain and S. Spiegel, "Time series classification in dissimilarity spaces," in *Proceedings of 1st International Workshop on Advanced Analytics and Learning on Temporal Data*, Porto, Portugal, 2015.

[12] E. Pekalska, R. P. Duin, and P. Paclik, "Prototype selection for dissimilarity-based classifiers," *Pattern Recognition*, vol. 39, no. 2, pp. 189–208, 2006.

[13] 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.

[14] 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, 1978.

[15] G. Benson, A. Levy, S. Maimoni, D. Noifeld, and B. R. Shalom, "LCSk: a refined similarity measure," *Theoretical Computer Science*, vol. 638, pp. 11–26, 2016.

[16] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition*, vol. 44, no. 3, pp. 678–693, 2011.

[17] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data mining and knowledge discovery*, vol. 2, no. 3, pp. 283–304, 1998.

[18] C.-Y. Hor, C.-B. Yang, C.-H. Chang, C.-T. Tseng, and H.-H. Chen, "A tool preference selection method for rna secondary structure prediction by SVM with statistical tests," *Evolutionary Bioinformatics*, vol. 9, pp. 163–184, Apr. 2013.

[19] Š. Raudys and F. Roli, "The behavior knowledge space fusion method: Analysis of generalization error and strategies for performance improvement," *Multiple Classifier Systems*, pp. 160–160, 2003.

[20] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana, "The UCR time series classifi-

Table 6: The accuracy (in percentage) comparison of our method and the method of Jain and Stephan [11] with the DTW distance measurement in 42 UCR datasets.

Dataset \ Method	DTW ours	DTW Jain & Stephen
Synthetic_control	**98.4**	98.0
Gun_Point	**94.3**	82.9
CBF	83.1	**86.9**
FaceAll	**81.4**	72.6
OSULeaf	**77.0**	**77.0**
SwedishLeaf	**67.7**	60.4
50words	**73.7**	69.0
Trace	**100**	**100**
Two_Patterns	**100**	**100**
Wafer	**98.4**	98.0
FaceFour	**94.3**	82.9
Lighting2	83.1	**86.9**
Lighting7	**81.4**	72.9
ECG	**77.0**	**77.0**
Adiac	**67.7**	60.4
Yoga	**86.4**	83.6
Fish	75.4	**83.3**
Beef	**57.3**	50.0
Coffee	**97.2**	82.1
OliveOil	**86.7**	**86.7**
CinC_ECG_torso	**78.9**	65.1
ChlorineConcentration	60.0	**64.8**
DiatomSizeReduction	90.8	**96.7**
ECGFiveDays	65.3	**76.8**
FacesUCR	89.7	**90.5**
Haptics	**45.6**	37.7
InlineSkate	**49.9**	38.4
ItalyPowerDemand	91.5	**95.0**
MALLAT	**96.2**	93.4
MedicalImages	**76.3**	73.7
MoteStrain	**96.2**	83.5
SonyAIBORobotSurfaceII	81.8	**83.1**
SonyAIBORobotSurface	69.6	**72.5**
Symbol	**97.7**	95.0
TwoLeadECG	**94.1**	90.4
WordsSynonyms	**68.1**	64.9
Cricket_X	77.4	**77.7**
Cricket_Y	**79.5**	79.2
Cricket_Z	**80.5**	79.2
uWaveGestureLibrary_X	**78.3**	72.7
uWaveGestureLibrary_Y	**69.5**	63.4
uWaveGestureLibrary_Z	**72.2**	65.8
Weighted average	**85.4**	82.5
Arithmetic average	**81.2**	78.6

cation/clustering homepage," http://www.cs.ucr.edu/~eamonn/time_series_data/, 2011.

[21] G.-C. Guo, K.-S. Huang, and C.-B. Yang, "Time series classification based on the longest common subsequence similarity and ensemble learning." in *Proceedings of Symposium on Digital Life Technologies Symposium, DLT2016*, Pingtung, Taiwan, June 2016, pp. 84–91.

[22] N. Kapoor, A. Gandhi, A. Chaurasia *et al.*, "Central neurocytoma in the vermis of the cerebellum," *Indian Journal of Pathology and Microbiology*, vol. 52, no. 1, p. 108, 2009.

[23] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[24] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[25] C.-J. Hsu, K.-S. Huang, C.-B. Yang, and Y.-P. Guo, "Flexible dynamic time warping for time series classification," *Procedia Computer Science 51, International Conference On Computational Science, ICCS 2015*, vol. 51, pp. 2838–2842, June 2015.

[26] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.