A 2D Transform Based Distance Function for Time Series Classification

Cun Ji[1,4(✉)], Xiunan Zou[1], Yupeng Hu[2(✉)], and Shijun Liu[2,3(✉)]

[1] School of Information Science and Engineering, Shandong Normal University, Jinan, China
jicun@sdnu.edu.cn, Parker_nan@163.com

[2] School of Software, Shandong University, Jinan, China
{huyupeng,lsj}@sdu.edu.cn

[3] Engineering Research Center of Digital Media Technology, Ministry of Education, Jinan, China

[4] Shandong Provincial Key Laboratory of Software Engineering, Shandong University, Jinan, China

Abstract. Along with the arrival of Industry 4.0 era, time series classification (TSC) has attracted a lot of attention in the last decade. The high dimensionality, high feature correlation and typically high levels of noise that found in time series bring great challenges to TSC. Among TSC algorithms, the 1NN classifier has been shown as effective and difficult to beat. The core of the 1NN classifier is the distance function. The large majority of TSC have concentrated on alternative distance functions. In this paper, a two-dimensional (2D) transform based distance (2DTbD) function is proposed. There are three steps in 2DTbD. Firstly, we convert time series to 2D space by turning time series around the coordinate origin. Then we calculate distances of each dimension. Finally, we ensemble distances in 2D space to get the final time series distance. Our distance function raises the accuracy rate through the fusion of 2D information. Experimental results demonstrate that the classification accuracy can be improved by 2DTbD.

Keywords: Internet of Things · Data mining · Time series classification · 2D transform · Distance function

1 Introduction

The Internet of things (IoT) is made up of small sensors and actuators embedded in objects with internet access [20]. It plays a key role in solving challenges faced in today's society [22,34]. Usually, sensors in an IoT system collect data with equal intervals. Data collection in an IoT system is time series data.

Time series is always considered as a whole instead of individual numerical field. Time series has attracted significant interests in the data mining community [30,40]. A time series analysis foresees tasks such as: indexing, representation, pattern discovery, clustering, classification, anomaly detection [13,15].

H. Gao et al. (Eds.): CollaborateCom 2018, LNICST 268, pp. 543–559, 2019.
https://doi.org/10.1007/978-3-030-12981-1_38

As a fundamental research, time series classification (TSC) has been studied extensively. In TSC, an unlabeled time series is assigned to one of two or more predefined classes. The high dimensionality, high feature correlation and typically high levels of noise bring great challenges to TSC [21,23,39]. Now, TSC arises in many real world fields [29], such as: electrocardiogram classification, fault detection and identification of physical systems, automotive preventive diagnosis, gesture recognition, alarm interpretation of telecommunication networks, data sensor analysis, speaker identification and/or authentication, aerospace health monitoring.

A lot of TSC algorithms have been proposed. Among TSC algorithms, the 1NN classifier with dynamic time warping (DTW, [6]) has been shown as effective and difficult to beat [5,12,36]. The core of the 1NN classifier is the distance function. The large majority of TSC have concentrated on alternative distance functions [1,3].

In this paper, a new distance function based on 2D transform is proposed. This distance function raises the accuracy rate through the fusion of 2D information. The contributions of this paper can be summarized as follows:

1. We introduce a 2D transform method for time series. The 2D transform method can transform time series into a 2D space.
2. We propose a 2D transform based distance (2DTbD) function. The 2DTbD function calculates distance between two time series using information after 2D transform.
3. We evaluate 2DTbD on 43 data sets from UCR Time Series Classification Archive [9]. The experimental results show that our distance function is more accurate.

The rest of this paper is structured as follows. Section 2 discusses some related work on TSC. In Sect. 3, we describe how to classification time series using 1NN classifier with 2DTbD, 2D transform is also given in this section. Experimental results are presented in Sect. 4, and our conclusions are given in Sect. 5.

2 Related Work

In recent years, many new distance functions are proposed to measure similarity between two time series [10]. These distance functions can be divided into two categories: time domain distance functions and differential distance functions.

Suppose there are two time series, T and R of length m as (1) (2), the distance $d(T, R)$ between them can be measured by distance functions. Next, we'll describe how to calculate distance $d(T, R)$.

$$T = (t_1, t_2, \cdots, t_i, \cdots, t_m) \tag{1}$$

$$R = (r_1, r_2, \cdots, r_i, \cdots, r_m) \tag{2}$$

2.1 Time Domain Distance Functions

2.1.1 Euclidean Distance Function

A benchmark distance function is Euclidean distance (EU, [14]). The function of EU is described as (3). EU is only used to calculate the distance between two time series whose lengths are equal [32]. It is sensitive to outliers.

$$d_{EU}(T,R) = \sqrt{\sum_{i=1}^{m}(t_i - r_i)^2} \tag{3}$$

Base on EU, L_p norm distance (L_p norm, [37]) function is proposed. The function of L_p norm is described as (4). EU is a special case of L_p norm when p is 2. L_p norm is also sensitive to outliers.

$$d_{L_p norm}(T,R) = \sqrt[p]{\sum_{i=1}^{m}(t_i - r_i)^p} \tag{4}$$

2.1.2 Dynamic Time Warping Function

An other benchmark distance function is DTW. DTW is not sensitive to distortions in time dimension. The function of DTW is described as (5), (6) and (7). In (5), n is the length of T, m is the length of R. In (6) and (7), i and j should meet the conditions in (8).

$$d_{DTW}(T,R) = \sqrt{f(n,m)} \tag{5}$$

$$f(i,j) = (t_i - t_j)^2 + min\,(f(i,j-1), f(i-1,j), f(i-1,j-1)) \tag{6}$$

$$f(0,0) = 0, \quad f(i,0) = f(0,j) = \infty \tag{7}$$

$$0 < i \leq n \& 0 < j \leq m \tag{8}$$

The time complexity of DTW is $O(n^2)$. By using some constraint techniques, such as Sakoe–Chuba Band [31] and Itakura Parallelogram [18], the time complexity of DTW can be reduced [19]. Some lower bounding functions are introduced to speed up DTW, such as LB_Kim [26], LB_Yi [38], LB_Keogh [24]. Salvador and Chan introduce FastDTW [33], which is an accurate approximation of DTW that runs in linear time and space.

2.1.3 Other Time Domain Distance Functions

Besides EU and DTW, there are some alternative techniques taken from other fields [3], such as edit distance with real penalty (ERP, [7]) and longest common subsequence (LCSS, [11]). Marteau proposes the time warp edit (TWE) distance [28]. TWE is an elastic distance metric, which includes characteristics from LCSS and DTW. Stefan et al. propose the Move-Split-Merge (MSM) distance [35]. MSM is similar to other edit distance based distance function. MSM uses three fundamental operations: Move, Split, and Merge. These operations can be used to transform a time series into a target series.

2.2 Differential Distance Functions

Usually, the differential distance functions base on the values in the time domain and the difference domain. The first order differences of time series $T = (t_1, t_2, \cdots, t_i, \cdots, t_m)$ are calculated as (9).

$$t_i^{'} = t_i - t_{i+1}, i = 1, 2, \cdots, m-1 \tag{9}$$

2.2.1 Complexity Invariant Distance Function

Batista et al. propose complexity invariant distance (CID) [4]. CID uses the sum of squares of the first differences as a measure of complexity. The function of CID is described as (10), (11) and (12). In (10), $d(T, R)$ can be EU or DTW. In (11) and (12), $t_i^{'}$ and $r_i^{'}$ are calculated as (9).

$$d_{CID}(T, R) = d(T, R) * \frac{max(CE(T), CE(R))}{min(CE(T), CE(R))} \tag{10}$$

$$CE(T) = \sqrt{\sum_{i=1}^{m-1} t_i^{'}} \tag{11}$$

$$CE(R) = \sqrt{\sum_{i=1}^{m-1} r_i^{'}} \tag{12}$$

2.2.2 Derivative Distance Function

Górecki and Łuczak propose derivative distance (DD) [16]. DD uses a weighted combination of raw time series and first order differences. The function of DD is described as (13). In (13), $T^{'}$ and $R^{'}$ are the first order differences of time series T and R, respectively. The values in $T^{'}$ and $R^{'}$ is calculated as (9). In (13), the dist function for raw time series or first order differences is basic distance function, such as EU, DTW.

$$d_{DD}(T, R) = \alpha * d(T, R) + (1 - \alpha) * d(T^{'}, R^{'}) \quad (0 \leq \alpha \leq 1) \tag{13}$$

2.2.3 Derivative Transform Distance Function

Based on derivatives and transforms of time series, Górecki and Łuczak propose derivative transform distance (DTD) [17]. DTD is an extension of DD. The function of DTD is described as (14).

$$d_{DTD}(T, R) = \alpha * d(T, R) + (1 - \alpha) * d(f(T), f(R)) \quad (0 \leq \alpha \leq 1) \tag{14}$$

In (14), $f(T)$ and $f(R)$ are the transform of time series T and R, respectively. In addition to first order differences, DTD can use cosine transform, sine transform or Hilbert transform. For $T = (t_1, \cdots, t_i, \cdots, t_m)$, the transform is represented as $f(T) = (f(t_1), \cdots, f(t_k), \cdots, f(t_m))$. For $f(t_k)$, cosine transform, sine transform and Hilbert transform are calculated as (15), (16) and (17), respectively.

$$f_{cos}(t_k) = \sum_{i=1}^{m} t_i * cos\left[\frac{\pi}{m} * \left(i - \frac{1}{2}\right) * (k-1)\right] \tag{15}$$

$$f_{sin}(t_k) = \sum_{i=1}^{m} t_i * sin\left[\frac{\pi}{m} * \left(i - \frac{1}{2}\right) * k)\right] \tag{16}$$

$$f_{Hilbert}(t_k) = \sum_{i=1, i\neq k}^{m} \frac{t_i}{k-i} \tag{17}$$

3 1NN Classifier with 2DTbD

We select the 1NN classifier with our distance function to classify time series. As Fig. 1 shows, there are mainly two processes to classify one time series: (1) 2D transform and (2) classification.

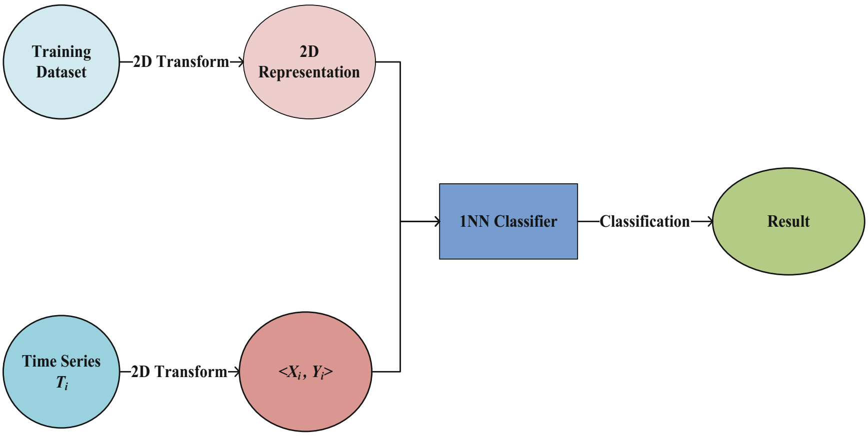


Fig. 1. Processes of classification based on 2D transform

3.1 2D Transform

Recently, many researchers convert the shapes into time series (read [5,8,25] for detail). The distance from each point to the center of gravity is calculated as value of time series. A demo of how convert a shape into time series is shown in Fig. 2. This demo refers to [25].

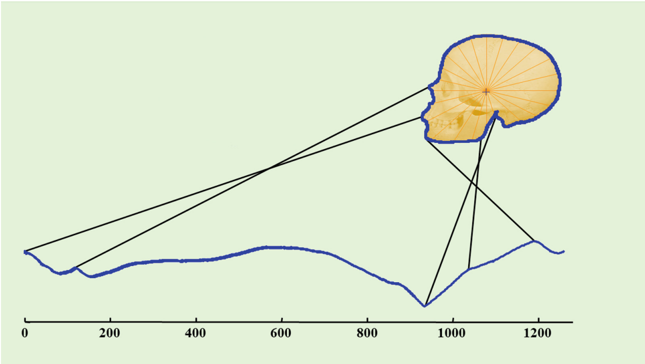


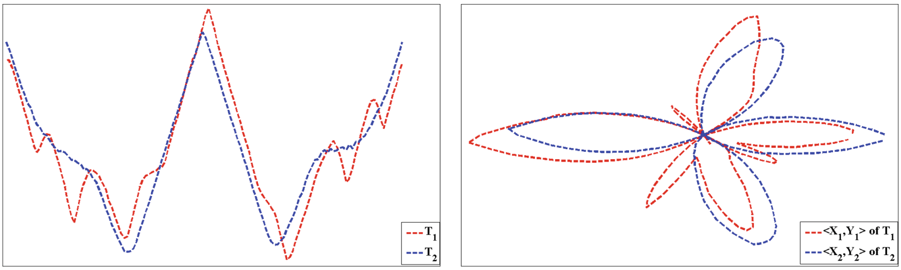
Fig. 2. A demo of how convert a shape into time series

2D transform acts in a diametrically opposite way. 2D transform turns time series around the coordinate origin. In this way, time series $T = (t_1, \cdots, t_i, \cdots, t_m)$ is converted into a 2D space. We use 2D representation $< X, Y >$ to represent T. The value x_i in $X = (x_1, \cdots, x_i, \cdots, x_m)$ is calculated as (18), the value y_i in $Y = (y_1, \cdots, y_i, \cdots, y_m)$ is calculated as (19). In (18) and (19), m is the length of time series T.

$$x(i) = t_i * cos\left[\frac{2 * \pi}{m} * (i - 1)\right] \tag{18}$$

$$y(i) = t_i * sin\left[\frac{2 * \pi}{m} * (i - 1)\right] \tag{19}$$

A demo of 2D transform is shown in Fig. 3. In Fig. 3(a), there are two time series, T_1 and T_2. Figure 3(b) shows 2D representation $< X_1, Y_1 >$ of T_1 and $< X_2, Y_2 >$ of T_2.



(a) Two time series, T_1 and T_2 (b) 2D representation for T_1 and T_2

Fig. 3. A demo of 2D transform

3.2 1NN Classifier with 2DTbD

After 2D transform, we use 1NN classifier with 2DTbD to classify the 2D representation of time series. 1NN classifier for 2D representation is shown in Algorithm 1. In Algorithm 1, we input 2D representation $< X, Y >$ of time series T and 2D representation of the training data set $2DTrain$. We calculate the dist between $< X, Y >$ and every 2D representation in $2DTrain$. We record the minimum distance $minDist$ and the corresponding class label C_T. Finally, the class label C_T is return as the classification result.

Algorithm 1. 1NN classifier for 2D representation

Input: 2D representation of time series T: $< X, Y >$, 2D representation of training data set: $2DTrain$
Output: Class label of T: c_T
1: $miDist = +\infty$
2: **for** each $(< X_i, Y_i >, c_i)$ in $2DTrain$ **do**
3: $\quad dist = d_{2DTbD}(< X, Y >, < X_i, Y_i >)$;
4: $\quad$ **if** $(dist < mixDist)$ **then**
5: $\quad\quad minDist = dist$
6: $\quad\quad c_T = c_i$
7: $\quad$ **end if**
8: **end for**
9: **return** c_T

In line 3 of Algorithm 1, 2DTbD is calculated. Calculation processes for 2DTbD are shown in Fig. 4. In Fig. 4, there are two time series, T_i and T_j. The 2D representation of them are $< X_i, Y_i >$ and $< X_j, Y_j >$, respectively. As Fig. 4 shows, there are mainly two steps to calculate 2DTbD:

Step 1: Calculate distance of each dimension. Each dimension is viewed as a new time series, we calculate $d(X_i, X_j)$ in X dimension and $d(Y_i, Y_j)$ in Y dimension by time series distance function, such as EU, L_p norm, DTW, ERP, CID.

Step 2: Calculate 2DTbD by ensemble dimension distances. As (20) shows, we use the means of $d(X_i, X_j)$ in X dimension and $d(Y_i, Y_j)$ in Y dimension as the distance of time series.

$$d_{2DTbD}(T_i, T_j) = \frac{d(X_i, X_j) + d(Y_i, Y_j)}{2} \tag{20}$$

4 Experiments

4.1 Experimental Setup

- **Data sets.** We performed experiments on 43 data sets from the UCR Time Series Classification Archive [9]. We used the UCR Time Series Classification

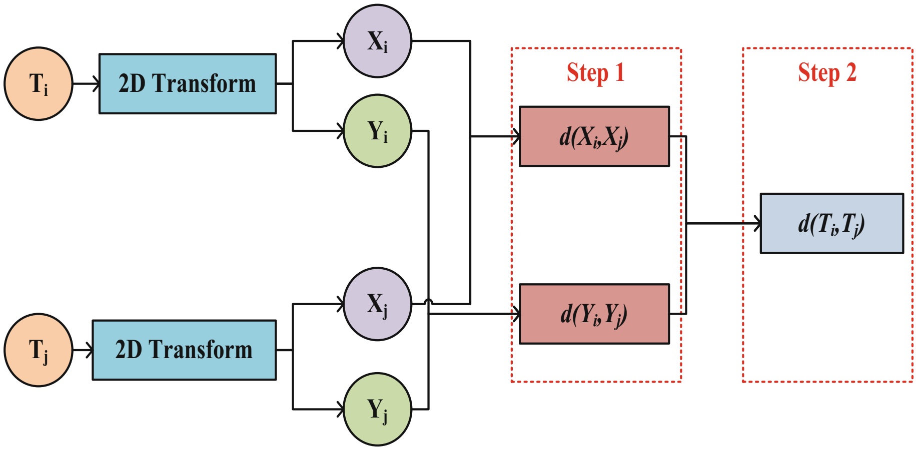


Fig. 4. Calculation processes for 2DTbD

Archive for two reasons [22,27]: (1) This archive has a large number of publicly accessible data sets; (2) These data sets cover a wide range of domains, from environmental monitoring to medical diagnosis.

- **Evaluation criterion.** We use error rate err as an evaluation criterion. The err is calculated as (21). In (21), n is the number of time series in test data set, n_{err} is the number of time series which is divided into an error class. The smaller err is, the better classifier effects.

$$err = \frac{n_{err}}{n} \tag{21}$$

- **Comparison distance functions.** We select CID, ERP and DTW as comparison distance functions. CID is presentation for differential distance functions. DTW and ERP is representant for time domain distance functions. Aslo, DTW is a benchmark distance function for time series.
- **Algorithm code.** We implemented our algorithm based on code that is freely accessible from an online repository [2]. **In order to promote reproducibility, our code and detailed results are open**[1]. Our code is carried out in Java using MyEclipse with JDK 1.8.

4.2 Contrast Experiments Between CID and 2DTbD$_{CID}$

When 2DTbD uses CID for each dimension, we call it 2DTbD$_{CID}$. In this subsection, we will contrast CID with 2DTbD$_{CID}$. As (10) in Sect. 2.2.1 shows, CID is based on EU or DTW, we call them CID$_{EU}$ or CID$_{DTW}$. Our distance function uses CID$_{EU}$ or CID$_{DTW}$ for each dimension, then we get 2DTbD$_{CIDEU}$ or 2DTbD$_{CIDDTW}$.

[1] Web page for our code: https://github.com/sdujicun/XY.

4.2.1 Contrasting CID$_{EU}$ with 2DTbD$_{CIDEU}$

Table 1 lists the classification error rates of 1NN classifier with CID$_{EU}$ or 2DTbD$_{CIDEU}$. The lower error rate for each data set is given in bold. As Table 1 shows, 1NN classifier with 2DTbD$_{CIDEU}$ produced better accuracy on 26 data sets, whereas 1NN classifier with CID$_{EU}$ was better on 10 and these two classifiers tied on the other 7 data sets.

To show the differences better, the data in Table 1 is drawn in Fig. 5. In Fig. 5, the points at the bottom left of the blue line indicate cases where 2DTbD$_{CIDEU}$ achieves a lower error rate than CID$_{EU}$. Most points is in the bottom left of the blue line.

Table 1. The error rates of 1NN classifier with CID$_{EU}$ or 2DTbD$_{CIDEU}$

Data set	CID$_{EU}$	2DTbD$_{CIDEU}$	Data set	CID$_{EU}$	2DTbD$_{CIDEDEU}$
Adiac	**0.371**	0.376	Lightning7	0.479	**0.411**
Beef	**0.367**	**0.367**	Mallat	0.078	**0.074**
Car	**0.267**	**0.267**	MedicalImages	0.308	**0.301**
CBF	0.018	**0.017**	MoteStrain	0.248	**0.221**
Chlorine	0.355	**0.351**	NonThorax1	0.160	**0.133**
CinCECG	0.112	**0.074**	NonThorax2	0.120	**0.108**
Coffee	**0.000**	**0.000**	OliveOil	**0.133**	**0.133**
CricketX	0.400	**0.390**	OSULeaf	**0.413**	0.430
CricketY	0.492	**0.418**	Plane	**0.029**	**0.029**
CricketZ	0.451	**0.415**	SonySurface1	**0.160**	0.180
DiatomSize	**0.059**	0.062	SonySurface2	0.114	**0.111**
ECGFiveDays	**0.246**	0.249	StarCurves	**0.041**	0.053
FaceAll	0.287	**0.221**	SwedishLeaf	0.126	**0.120**
FaceFour	**0.205**	0.250	Symbols	0.088	**0.062**
FacesUCR	0.266	**0.223**	Synthetic	0.073	**0.067**
FiftyWords	0.338	**0.330**	Trace	0.130	**0.110**
Fish	**0.217**	**0.217**	TwoLeadECG	**0.229**	**0.229**
GunPoint	0.080	**0.073**	TwoPatterns	0.157	**0.114**
Haptics	**0.562**	0.571	Wafer	**0.008**	0.009
InlineSkate	0.615	**0.613**	WordSyn	0.361	**0.354**
ItalyPower	0.042	**0.035**	Yoga	**0.165**	0.168
Lightning2	0.328	**0.213**			

The results in Table 1 and Fig. 5 demonstrate that 2DTbD$_{CIDEU}$ works better than CID$_{EU}$.

4.2.2 Contrasting CID$_{DTW}$ with 2DTbD$_{CIDDTW}$

Table 2 lists the classification error rates of 1NN classifier with CID$_{DTW}$ or 2DTbD$_{CIDDTW}$. The lower error rate for each data set is given in bold. As

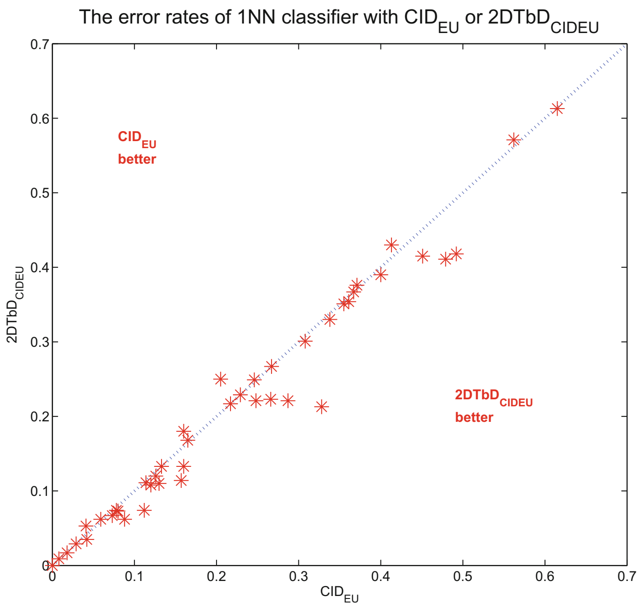


Fig. 5. The error rates of 1NN classifier with CID$_{EU}$ or 2DTbD$_{CIDEU}$ (Color figure online)

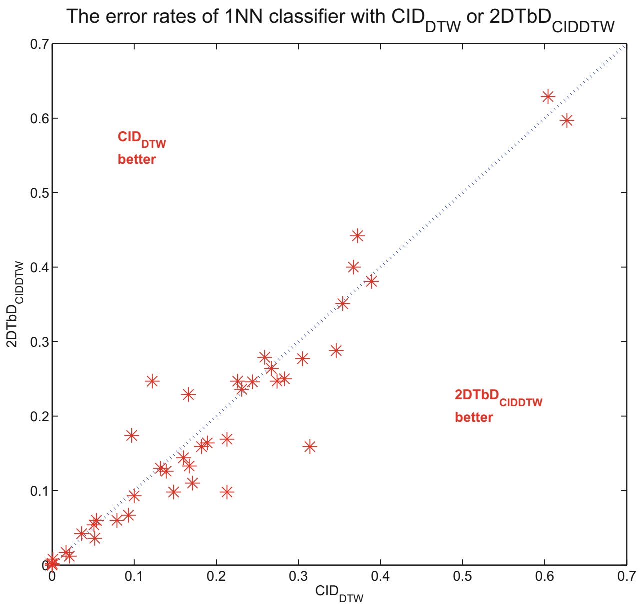


Fig. 6. The error rates of 1NN classifier with CID$_{DTW}$ or 2DTbD$_{CIDDTW}$ (Color figure online)

Table 2. The error rates of 1NN classifier with CID_{DTW} or 2DTbD_{CIDDTW}

Data set	CID_{DTW}	2DTbD_{CIDDTW}	Data set	CID_{DTW}	2DTbD_{CIDDTW}
Adiac	0.389	**0.381**	Lightning7	0.274	**0.247**
Beef	**0.367**	0.400	Mallat	**0.054**	0.060
Car	0.283	**0.250**	MedicalImages	**0.259**	0.279
CBF	**0.001**	0.008	MoteStrain	0.189	**0.164**
Chlorine	0.354	**0.351**	NonThorax1	0.213	**0.169**
CinCECG	0.314	**0.159**	NonThorax2	0.132	**0.130**
Coffee	**0.000**	**0.000**	OliveOil	0.167	**0.133**
CricketX	**0.231**	0.236	OSULeaf	**0.372**	0.442
CricketY	0.267	**0.264**	Plane	**0.000**	**0.000**
CricketZ	**0.244**	0.246	SonySurface1	0.213	**0.098**
DiatomSize	**0.036**	0.042	SonySurface2	**0.122**	0.247
ECGFiveDays	**0.226**	0.247	StarCurves	0.079	**0.060**
FaceAll	0.139	**0.126**	SwedishLeaf	0.171	**0.110**
FaceFour	0.182	**0.159**	Symbols	**0.051**	0.054
FacesUCR	0.100	**0.093**	Synthetic	**0.017**	**0.017**
FiftyWords	0.305	**0.277**	Trace	**0.000**	**0.000**
Fish	**0.166**	0.229	TwoLeadECG	**0.097**	0.174
GunPoint	0.093	**0.067**	TwoPatterns	**0.000**	0.002
Haptics	0.627	**0.597**	Wafer	0.021	**0.012**
InlineSkate	**0.604**	0.629	WordSyn	0.346	**0.288**
ItalyPower	0.052	**0.036**	Yoga	0.160	**0.144**
Lightning2	0.148	**0.098**			

Table 2 shows, 1NN classifier with 2DTbD_{CIDDTW} produced better accuracy on 24 data sets, whereas 1NN classifier with CID_{DTW} was better on 15 and these two classifiers tied on the other 4 data sets.

To show the differences better, the data in Table 2 is drawn in Fig. 6. In Fig. 6, the points at the bottom left of the blue line indicate cases where 2DTbD_{CIDDTW} achieves a lower error rate than CID_{DTW}. Most points is in the bottom left of the blue line.

The results in Table 2 and Fig. 6 demonstrate that 2DTbD_{CIDDTW} works better than CID_{DTW}.

4.3 Contrast Experiments Between ERP and 2DTbD_{ERP}

When 2DTbD uses ERP for each dimension, we call it 2DTbD_{ERP}. In this subsection, we will compare 1NN classifier with ERP to 1NN classifier with 2DTbD_{ERP}. In our experiments, the gap g for ERP is set to 0.5, the maximum allowed distance to the diagonal $bandSize$ is set to 0.5.

Table 3 lists the classification error rates of them. The lower error rate for each data set is given in bold. As Table 3 shows, 1NN classifier with 2DTbD_{ERP} produced better accuracy on 24 data sets, whereas 1NN classifier with DTW was better on 14 and these two classifiers tied on the other 5 data sets.

Table 3. The error rates of 1NN classifier with ERP or 2DTbD$_{ERP}$

Data set	ERP	2DTbD$_{ERP}$	Data set	ERP	2DTbD$_{ERP}$
Adiac	0.389	**0.373**	Lightning7	0.247	**0.192**
Beef	**0.367**	**0.367**	Mallat	0.087	**0.078**
Car	**0.200**	0.233	MedicalImages	0.329	**0.305**
CBF	0.033	**0.009**	MoteStrain	0.142	**0.131**
Chlorine	**0.340**	0.344	NonThorax1	**0.172**	0.173
CinCECG	0.276	**0.157**	NonThorax2	**0.107**	0.114
Coffee	**0.000**	0.036	OliveOil	**0.133**	**0.133**
CricketX	**0.236**	0.244	OSULeaf	0.430	**0.384**
CricketY	0.315	**0.249**	Plane	0.019	**0.010**
CricketZ	0.285	**0.274**	SonySurface1	0.225	**0.210**
DiatomSize	**0.046**	0.056	SonySurface2	**0.163**	0.174
ECGFiveDays	0.146	**0.084**	StarCurves	0.148	**0.142**
FaceAll	0.215	**0.200**	SwedishLeaf	0.138	**0.118**
FaceFour	**0.170**	**0.170**	Symbols	**0.075**	0.081
FacesUCR	0.071	**0.060**	Synthetic	**0.060**	**0.060**
FiftyWords	0.310	**0.273**	Trace	**0.050**	0.120
Fish	**0.097**	0.126	TwoLeadECG	**0.082**	0.215
GunPoint	**0.067**	0.080	TwoPatterns	**0.000**	**0.000**
Haptics	0.581	**0.565**	Wafer	0.010	**0.005**
InlineSkate	**0.565**	0.611	WordSyn	0.359	**0.321**
ItalyPower	0.061	**0.036**	Yoga	0.155	**0.143**
Lightning2	0.131	**0.115**			

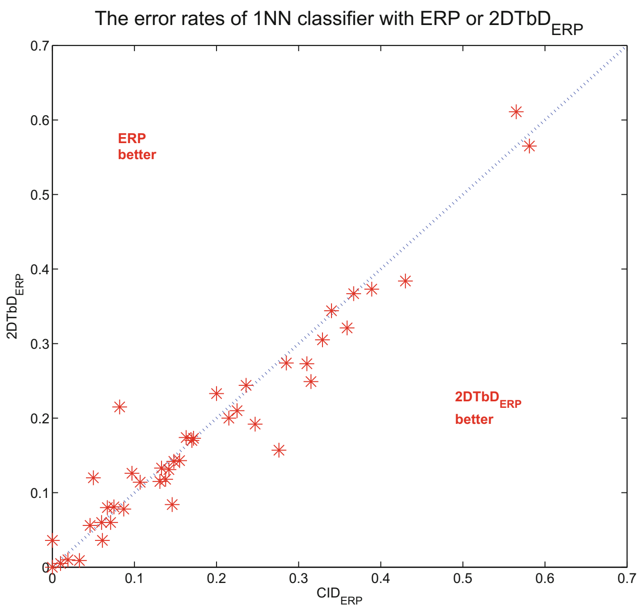


Fig. 7. The error rates of 1NN classifier with ERP or 2DTbD$_{ERP}$ (Color figure online)

To show the differences better, the data in Table 3 is drawn in Fig. 7. In Fig. 7, the points at the bottom left of the blue line indicate cases where 2DTbD$_{ERP}$ achieves a lower error rate than ERP. Most points is in the bottom left of the blue line.

The results in Table 3 and Fig. 7 demonstrate that 2DTbD$_{ERP}$ works better than ERP. This proves that the accuracy can be improved by 2DTbD.

4.4 Contrast Experiments Between DTW and 2DTbD$_{DTW}$

When 2DTbD uses DTW for each dimension, we call it 2DTbD$_{DTW}$. In this subsection, we will compare 1NN classifier with DTW to 1NN classifier with 2DTbD$_{DTW}$.

Table 4 lists the classification error rates of them. The lower error rate for each data set is given in bold. As Table 4 shows, 1NN classifier with 2DTbD$_{DTW}$ produced better accuracy on 21 data sets, whereas 1NN classifier with DTW was better on 17 and these two classifiers tied on the other 5 data sets.

To show the differences better, the data in Table 4 is drawn in Fig. 8. In Fig. 8, the points at the bottom left of the blue line indicate cases where 2DTbD$_{DTW}$ achieves a lower error rate than DTW. Most points is in the bottom left of the blue line.

Table 4. The error rates of 1NN classifier with DTW or 2DTdD$_{DTW}$

Data set	DTW	2DTbD$_{DTW}$	Data set	DTW	2DTbD$_{DTW}$
Adiac	0.396	**0.389**	Lightning7	0.274	**0.233**
Beef	0.367	**0.333**	Mallat	**0.066**	0.077
Car	**0.267**	**0.267**	MedicalImages	**0.263**	0.287
CBF	**0.003**	0.029	MoteStrain	0.165	**0.134**
Chlorine	0.352	**0.350**	NonThorax1	0.210	**0.188**
CinCECG	0.349	**0.185**	NonThorax2	0.135	**0.131**
Coffee	**0.000**	**0.000**	OliveOil	0.167	**0.133**
CricketX	0.246	**0.226**	OSULeaf	**0.409**	0.455
CricketY	**0.256**	0.272	Plane	**0.000**	**0.000**
CricketZ	0.246	**0.241**	SonySurface1	**0.275**	0.296
DiatomSize	**0.033**	0.042	SonySurface2	**0.169**	0.170
ECGFiveDays	**0.232**	0.239	StarCurves	0.093	**0.073**
FaceAll	**0.192**	0.193	SwedishLeaf	0.208	**0.130**
FaceFour	0.170	**0.159**	Symbols	**0.050**	0.061
FacesUCR	**0.095**	0.098	Synthetic	**0.007**	0.013
FiftyWords	0.310	**0.279**	Trace	**0.000**	**0.000**
Fish	**0.177**	0.229	TwoLeadECG	**0.096**	0.182
GunPoint	0.093	**0.060**	TwoPatterns	**0.000**	0.001
Haptics	0.623	**0.607**	Wafer	0.020	**0.011**
InlineSkate	**0.616**	0.658	WordSyn	0.351	**0.292**
ItalyPower	0.050	**0.036**	Yoga	0.163	**0.145**
Lightning2	**0.131**	**0.131**			

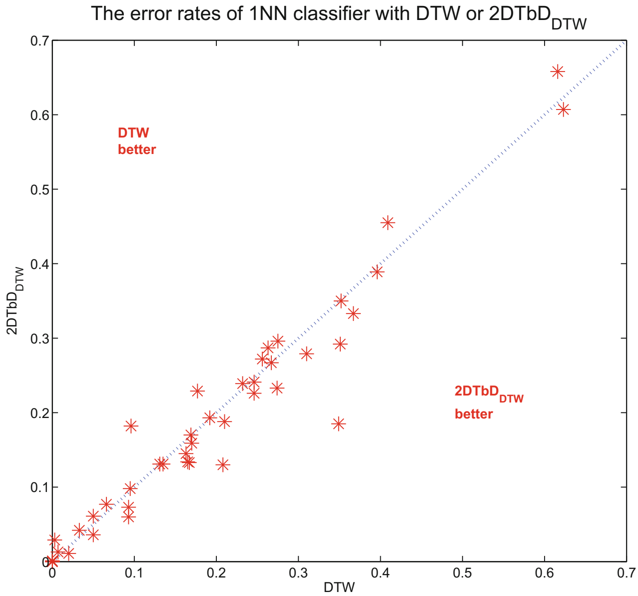


Fig. 8. The error rates of 1NN classifier with DTW or 2DTdD$_{DTW}$ (Color figure online)

The results in Table 4 and Fig. 8 demonstrate that 2DTbD$_{DTW}$ works slightly better than DTW. This proves that the accuracy can be slightly improved by 2DTbD.

4.5 Summary of Experimental Results

All results of our experiments are shown in Fig. 9. As Fig. 9 shows, 2DTbD performs better on all comparison experiments.

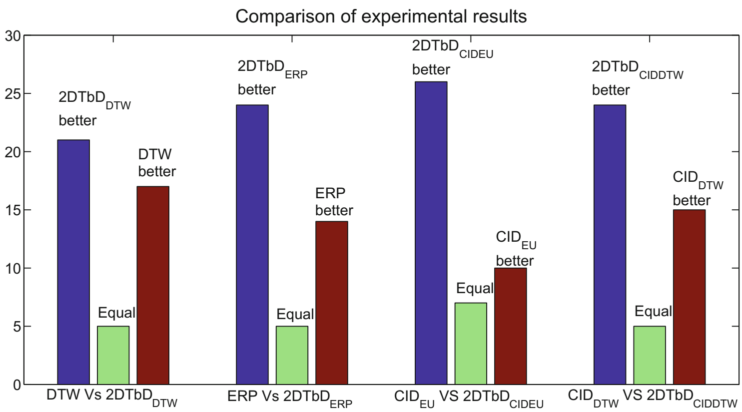


Fig. 9. Comparison of experimental results

5 Conclusion

TSC has attracted a lot of attention in the last decade. Among TSC algorithms, the 1NN classifier has been shown as effective and difficult to beat. The core of 1NN classifier is the distance function. In this paper, a new distance function 2DTbD is proposed. 2DTbD calculates distance by merging distances in 2D space. Our experimental results demonstrate that the classification accuracy can be improved by 2DTbD.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (61872222, 91546203), the National Key Research and Development Program of China (2017YFA0700601), the Major Program of Shandong Province Natural Science Foundation (ZR2018ZB0419), the Key Research and Development Program of Shandong Province (2017CXGC0605, 2017CXGC0604, 2018GGX101019).

References

1. Bagnall, A., Bostrom, A., Large, J., Lines, J.: The great time series classification bake off: an experimental evaluation of recently proposed algorithms. Extended version. arXiv preprint arXiv:1602.01711 (2016)
2. Bagnall, A., Bostrom, A., Lines, J.: The UEA TSC codebase (2016). https://bitbucket.org/aaron_bostrom/time-series-classification
3. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Disc. **31**(3), 606–660 (2017)
4. Batista, G.E., Keogh, E.J., Tataw, O.M., De Souza, V.M.: CID: an efficient complexity-invariant distance for time series. Data Min. Knowl. Disc. **28**(3), 634–669 (2014)
5. Batista, G.E., Wang, X., Keogh, E.J.: A complexity-invariant distance measure for time series. In: Proceedings of the 2011 SIAM International Conference on Data Mining, pp. 699–710. SIAM (2011)
6. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD Workshop, Seattle, WA, vol. 10, pp. 359–370 (1994)
7. Chen, L., Ng, R.: On the marriage of Lp-norms and edit distance. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases, vol. 30, pp. 792–803. VLDB Endowment (2004)
8. Chen, Y., Hu, B., Keogh, E., Batista, G.E.: 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, pp. 383–391. ACM (2013)
9. Chen, Y., et al.: The UCR time series classification archive (2015). http://www.cs.ucr.edu/~eamonn/time_series_data
10. Chhieng, V.M., Wong, R.K.: Adaptive distance measurement for time series databases. In: Kotagiri, R., Krishna, P.R., Mohania, M., Nantajeewarawat, E. (eds.) DASFAA 2007. LNCS, vol. 4443, pp. 598–610. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71703-4_51
11. Das, G., Gunopulos, D., Mannila, H.: Finding similar time series. In: Komorowski, J., Zytkow, J. (eds.) PKDD 1997. LNCS, vol. 1263, pp. 88–100. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63223-9_109

12. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proc. VLDB Endowment **1**(2), 1542–1552 (2008)
13. Esling, P., Agon, C.: Time-series data mining. ACM Comput. Surv. **45**(1), 12 (2012)
14. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases, vol. 23. ACM (1994)
15. Fu, T.: A review on time series data mining. Eng. Appl. Artif. Intell. **24**(1), 164–181 (2011)
16. Górecki, T., Łuczak, M.: Using derivatives in time series classification. Data Min. Knowl. Disc. **26**(2), 310–331 (2013)
17. Górecki, T., Łuczak, M.: Non-isometric transforms in time series classification using DTW. Knowl.-Based Syst. **61**, 98–108 (2014)
18. Itakura, F.: Minimum prediction residual principle applied to speech recognition. IEEE Trans. Acoust. Speech Signal Process. **23**(1), 67–72 (1975)
19. Jeong, Y.S., Jeong, M.K., Omitaomu, O.A.: Weighted dynamic time warping for time series classification. Pattern Recognit. **44**(9), 2231–2240 (2011)
20. Ji, C., et al.: A self-evolving method of data model for cloud-based machine data ingestion. In: 2016 IEEE 9th International Conference on Cloud Computing, pp. 814–819. IEEE (2016)
21. Ji, C., Liu, S., Yang, C., Pan, L., Wu, L., Meng, X.: A shapelet selection algorithm for time series classification: new directions. Procedia Comput. Sci. **129**, 461–467 (2018)
22. Ji, C., Zhao, C., Pan, L., Liu, S., Yang, C., Wu, L.: A fast shapelet discovery algorithm based on important data points. Int. J. Web Serv. Res. **14**(2), 67–80 (2017)
23. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: a survey and empirical demonstration. Data Min. Knowl. Disc. **7**(4), 349–371 (2003)
24. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowl. Inf. Syst. **7**(3), 358–386 (2005)
25. Keogh, E., Wei, L., Xi, X., Lee, S.H., Vlachos, M.: LB_Keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures. In: Proceedings of the 32nd International Conference on Very Large Data Bases, pp. 882–893. VLDB Endowment (2006)
26. Kim, S.W., Park, S., Chu, W.W.: An index-based approach for similarity search supporting time warping in large sequence databases. In: Proceedings of 17th International Conference on Data Engineering, pp. 607–614. IEEE (2001)
27. Li, D., Bissyandé, T.F., Klein, J., Le Traon, Y.: DSCo-NG: a practical language modeling approach for time series classification. In: Boström, H., Knobbe, A., Soares, C., Papapetrou, P. (eds.) IDA 2016. LNCS, vol. 9897, pp. 1–13. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46349-0_1
28. Marteau, P.F.: Time warp edit distance with stiffness adjustment for time series matching. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 306–318 (2009)
29. Prieto, O.J., Alonso-González, C.J., Rodríguez, J.J.: Stacking for multivariate time series classification. Pattern Anal. Appl. **18**(2), 297–312 (2015)
30. Raza, A., Kramer, S.: Ensembles of randomized time series shapelets provide improved accuracy while reducing computational costs. arXiv preprint arXiv:1702.06712 (2017)
31. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Signal Process. **26**(1), 43–49 (1978)

32. Sakurai, Y., Yoshikawa, M., Faloutsos, C.: FTW: fast similarity search under the time warping distance. In: Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 326–337. ACM (2005)
33. Salvador, S., Chan, P.: Toward accurate dynamic time warping in linear time and space. Intell. Data Anal. **11**(5), 561–580 (2007)
34. Sampaio, A., Lima Jr., R.C., Mendonça, N.C., Filho, R.H.: Implementation and empirical assessment of a web application cloud deployment tool. Int. J. Cloud Comput. **1**, 40–52 (2013). http://hipore.com/stcc/2013/IJCC-Vol1-No1-2013.pdf#page=46
35. Stefan, A., Athitsos, V., Das, G.: The move-split-merge metric for time series. IEEE Trans. Knowl. Data Eng. **25**(6), 1425–1438 (2013)
36. Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.: Experimental comparison of representation methods and distance measures for time series data. Data Min. Knowl. Disc. **26**(2), 1–35 (2013)
37. Yi, B.K., Faloutsos, C.: Fast time sequence indexing for arbitrary Lp norms. VLDB (2000)
38. Yi, B.K., Jagadish, H., Faloutsos, C.: Efficient retrieval of similar time sequences under time warping. In: Proceedings of 14th International Conference on Data Engineering, pp. 201–208. IEEE (1998)
39. Zhang, Z., Cheng, J., Li, J., Bian, W., Tao, D.: Segment-based features for time series classification. Comput. J. **55**(9), 1088–1102 (2012)
40. Zhang, Z., Wen, Y., Zhang, Y., Yuan, X.: Time series classification by modeling the principal shapes. In: Bouguettaya, A., et al. (eds.) WISE 2017. LNCS, vol. 10569, pp. 406–421. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68783-4_28