

# Piecewise Factorization for Time Series Classification

Qinglin Cai<sup>(✉)</sup>, Ling Chen, and Jianling Sun

College of Computer Science and Technology,  
Zhejiang University, Hangzhou, China  
{qlcai, lingchen, sunjl}@zju.edu.cn

**Abstract.** In the research field of time series analysis and mining, the nearest neighbor classifier (1NN) based on the dynamic time warping distance (DTW) is well known for its high accuracy. However, the high computational complexity of DTW can lead to the expensive time consumption of the classifier. An effective solution is to compute DTW in the piecewise approximation space (PA-DTW). However, most of the existing piecewise approximation methods must predefine the segment length and focus on the simple statistical features, which would influence the precision of PA-DTW. To address this problem, we propose a novel piecewise factorization model (PCHA) for time series, where an adaptive segment method is proposed and the Chebyshev coefficients of subsequences are extracted as features. Based on PCHA, the corresponding PA-DTW measure named ChebyDTW is proposed for the 1NN classifier, which can capture the fluctuation information of time series for the similarity measure. The comprehensive experimental evaluation shows that ChebyDTW can support both accurate and fast 1NN classification.

**Keywords:** Time series · Piecewise approximation · Similarity measure

## 1 Introduction

Time series classification is an important topic in the research field of time series analysis and mining. A plethora of classifiers have been developed for this topic [1, 2], e.g., decision tree, nearest neighbor (1NN), naive Bayes, Bayesian network, random forest, support vector machine, rotation forest, etc. However, the recent empirical evidence [3–5] strongly suggests that, with the merits of robustness, high accuracy, and free parameter, the simple 1NN classifier employing the generic time series similarity measure is exceptionally difficult to beat. Besides, due to the high precision of dynamic time warping distance (DTW), the 1NN classifier based on DTW has been found to outperform an exhaustive list of alternatives [5], including decision trees, multi-scale histograms, multi-layer perception neural networks, order logic rules with boosting, as well as the 1NN classifiers based on many other similarity measures. However, the computational complexity of DTW is quadratic to the time series length, i.e.,  $O(n^2)$ , and the 1NN classifier has to search the entire dataset to classify an object. As a result, the 1NN classifier based on DTW is low efficient for the high-dimensional time series. To address this problem, researchers have proposed to compute DTW in the alternative

piecewise approximation space (PA-DTW) [6–9], which transforms the raw data into the feature space based on segmentation, and extracts the discriminatory and low-dimensional features for similarity measure. If the original time series with length  $n$  is segmented into  $N(N \ll n)$  subsequences, the computational complexity of PA-DTW will reduce to  $O(N^2)$ .

Many piecewise approximation methods have been proposed so far, e.g., piecewise aggregation approximation (PAA) [6], piecewise linear approximation (PLA) [7, 10], adaptive piecewise constant approximation (APCA) [8], derivative time series segment approximation (DSA) [9], piecewise cloud approximation (PWCA) [11], etc. The most prominent merit of piecewise approximation is the ability of capturing the local characteristics of time series. However, most of the existing piecewise approximation methods need to fix the segment length, which is hard to be predefined for the different kinds of time series, and focus on the simple statistical features, which only capture the aggregation characteristics of time series. For example, PAA and APCA extract the mean values, PLA extracts the linear fitting slopes, and DSA extracts the mean values of the derivative subsequences. If PA-DTW is computed on these methods, its precision would be influenced.

In this paper, we propose a novel piecewise factorization model for time series, named piecewise Chebyshev approximation (PCHA), where a novel code-based segment method is proposed to adaptively segment time series. Rather than focusing on the statistical features, we factorize the subsequences with Chebyshev polynomials, and employ the Chebyshev coefficients as features to approximate the raw data. Besides, the PA-DTW based on PCHA (ChebyDTW) is proposed for the 1NN classification. Since the Chebyshev polynomials with the different degrees represent the fluctuation components of time series, the local fluctuation information can be captured from time series for the ChebyDTW measure. The comprehensive experimental results show that ChebyDTW can support the accurate and fast 1NN classification.

The structure of this paper is as follows: The related work on data representation and similarity measure for time series is reviewed in Sect. 2; Sect. 3 shows the proposed methodology framework; the details of PCHA are presented in Sect. 4; Sect. 5 describes the ChebyDTW measure; Sect. 6 provides the comprehensive experiment results and analysis; Sect. 7 concludes this paper.

## 2 Related Work

### 2.1 Data Representation

In many application fields, the high dimensionality of time series has limited the performance of a myriad of algorithms. With this problem, a great number of data representation methods have been proposed to reduce the dimensionality of time series [1, 2]. In these methods, the piecewise approximation methods are prevalent for their simplicity and effectiveness. The first attempt is the PAA representation [6], which segments time series into the equal-length subsequences, and extracts the mean values of the subsequences as features to approximate the raw data. However, the extracted single sort of features only indicates the height of subsequences, which may cause the local

information loss. Consecutively, an adaptive version of PAA named piecewise constant approximation (APCA) [8] was proposed, which can segment time series into the subsequences with adaptive lengths and thus can approximate time series with less error. As well, a multi-resolution version of PAA named MPAA [12] was proposed, which can iteratively segment time series into  $2^i$  subsequences. However, both of the variations inherit the poor expressivity of PAA. Another pioneer piecewise representation is the PLA [7, 10], which extracts the linear fitting slopes of the subsequences as features to approximate the raw data. However, the fitting slopes only reflect the movement trends of the subsequences. For the time series fluctuating sharply with high frequency, the effect of PLA on dimension reduction is not prominent. In addition, two novel piecewise approximation methods were proposed recently. One is the DSA representation [9], which takes the mean values of the derivative subsequences of time series as features. However, it is sensitive to the small fluctuation caused by the noise. The other is the PWCA representation [11], which employs the cloud models to fit the data distribution of subsequences. However, the extracted features only reflect the data distribution characteristics and cannot capture the fluctuation information of time series.

## 2.2 Similarity Measure

DTW [1, 2, 5] is one of the most prevalent similarity measures for time series, which is computed by realigning the indices of time series. It is robust to the time warping and phase-shift, and has high measure precision. However, it is computed by the dynamic programming algorithm, and thus has the expensive  $O(n^2)$  computational complexity, which largely limits its application to the high dimensional time series [13]. To overcome this shortcoming, the PA-DTW measures were proposed. The PAA representation based PDTW [14] and the PLA representation based SDTW [10] are the early pioneers, and the DSA representation based DSADTW [9] is the state-of-the-art method. Rather than in the raw data space, they compute DTW in the PAA, PLA, and DSA spaces respectively. Since the segment numbers are much less than the original time series length, the PA-DTW methods can greatly decrease the computational complexity of the original DTW. Nonetheless, the precision of PA-DTWs greatly depends on the used piecewise approximation methods, where both the segment method and the extracted features are crucial factors. As a result, with the weakness of the existing piecewise approximation methods, the PA-DTWs cannot achieve the high precision. In our proposed ChebyDTW, a novel adaptive segment method and the Chebyshev factorization are used, which overcomes the drawback of the fixed segmentation, and can capture the fluctuation information of time series for similarity measure.

## 3 Methodology Framework

Figure 1 shows the framework of the methods proposed in this paper, which consists of two parts:

- (a) Piecewise Chebyshev approximation (PCHA). The time series is first coded into the binary sequence, and then segmented into the subsequences with adaptive

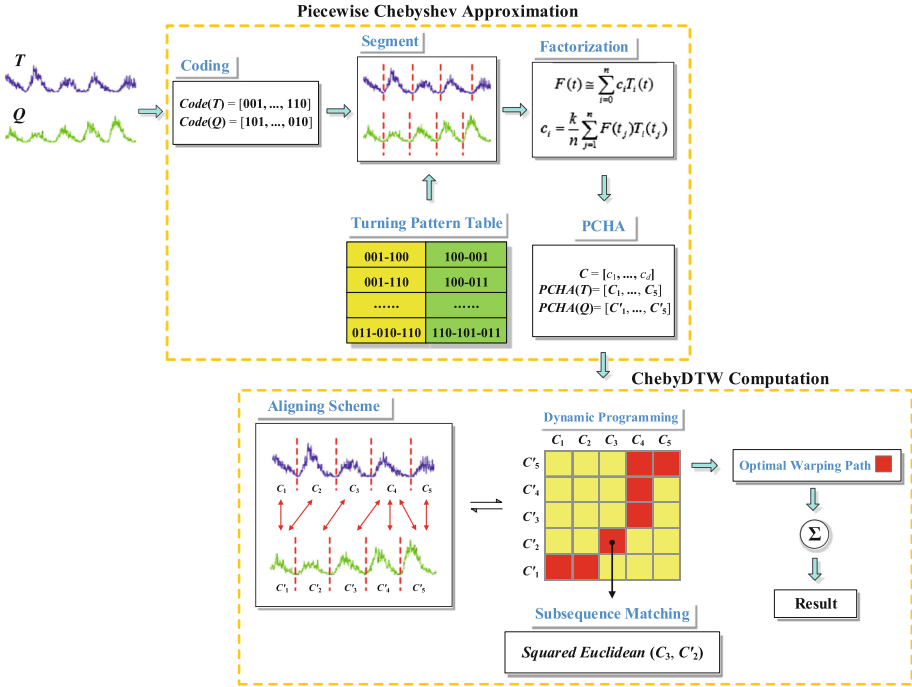


Fig. 1. The framework of the proposed methods.

lengths by matching the turning patterns. After that, the subsequences are factorized with the Chebyshev polynomials and projected into the Chebyshev factorization domain. The Chebyshev coefficients will be extracted as features to approximate the raw data.

- (b) ChebyDTW computation. DTW will be computed in the Chebyshev factorization domain. Concretely, in the dynamic programming computation of DTW, the subsequence matching over the Chebyshev features is taken as the subroutine, where the squared Euclidean distance can be employed.

### 4 Piecewise Factorization

Without loss of generality, the relevant definitions are first given as follows.

**Definition 1.** (Time Series): The sample sequence of a variable  $X$  over  $n$  contiguous time moments is called time series, denoted as  $T = \{t_1, t_2, \dots, t_i, \dots, t_n\}$ , where  $t_i \in R$  denotes the sample value of  $X$  on the  $i$ -th moment, and  $n$  is the length of  $T$ .

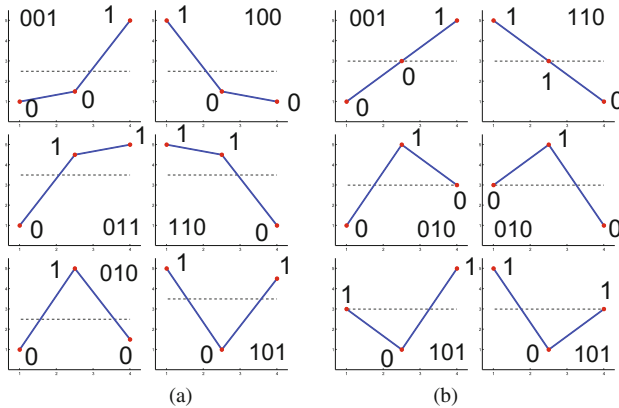
**Definition 2.** (Subsequence): Given a time series  $T = \{t_1, t_2, \dots, t_i, \dots, t_n\}$ , the subset  $S$  of  $T$  that consists of the continuous samples  $\{t_{i+1}, t_{i+2}, \dots, t_{i+l}\}$ , where  $0 \leq i \leq n-l$  and  $0 \leq l \leq n$ , is called the subsequence of  $T$ .

**Definition 3.** (Piecewise Approximation): Given a time series  $T = \{t_1, t_2, \dots, t_i, \dots, t_n\}$ , which is segmented into the subsequence set  $S = \{S_1, S_2, \dots, S_j, \dots, S_N\}$ , if  $\exists f: S_j \rightarrow V_j = [v_1, \dots, v_m] \in \mathbf{R}^m$ , the set  $V = \{V_1, V_2, \dots, V_j, \dots, V_N\}$  is called the piecewise approximation of  $T$ .

### 4.1 Adaptive Segmentation

Inspired by the Marr’s theory of vision [15], we regard the turning points, where the trend of time series changes, as a good choice to segment time series. However, the practical time series is mixed with a mass of noise, which results in many trivial turning points with small fluctuation. This problem can be simply solved by the efficient moving average (MA) smoothing method [16].

In order to recognize the significant turning points, we first exhaustively enumerate the location relationships of three adjacent samples  $t_{1-3}$  with their mean  $\mu$  in time series, as shown in Fig. 2. Six basic cell codes can be defined as Fig. 2(a), which is composed by the binary codes  $\delta_1-\delta_3$  of  $t_{1-3}$ , and denoted as  $\Phi(t_1, t_2, t_3) = (\delta_1\delta_2\delta_3)_b$ . Six special relationships that one of  $t_{1-3}$  equals to  $\mu$  are encoded as Fig. 2(b).



**Fig. 2.** Three adjacent samples with the basic cell codes of (A) basic relationships, and (B) specific relationships.

Based on the cell codes, all the minimum turning patterns (composed with two cell codes) at the turning points can be enumerated as Fig. 3. Note that, the basic cell codes 010 and 101 per se are the turning patterns. Then, we employ a sliding window of length 3 to scan the time series, and encode the samples within each window by Fig. 2. In this process, all the significant turning points can be found by matching Fig. 3, with which time series can be segmented into the subsequences with adaptive lengths.

However, the above segmentation is not perfect. Although the trivial turning points can be removed with the MA, the “singular” turning patterns may exist, i.e., the turning patterns appearing very close. As shown in Fig. 4, a Cricket time series from the UCR time series archive [17] is segmented by the turning patterns (dash line), where the raw data is first smoothed with the smooth degree 10 ( $sd = 10$ ).

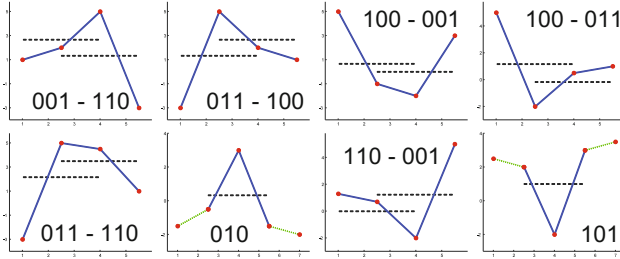


Fig. 3. The minimum turning patterns composed with two cell codes.

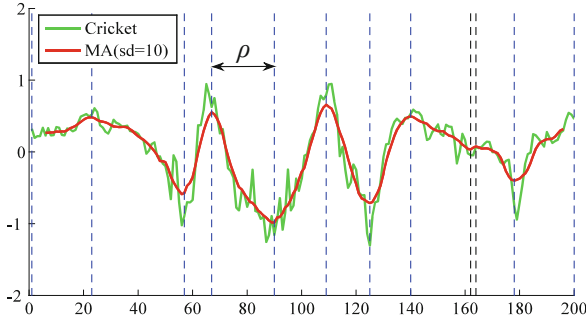


Fig. 4. Segmentation for the Cricket time series ( $sd = 10$ ).

Obviously, the dash lines can significantly segment time series, but the two black dash lines are so close that the segment between them can be ignored. In view of this, we introduce the segment threshold  $\rho$  that stipulates the minimum segment length. This parameter can be set as the ratio to the time series length. Since the time series from a specific field exhibit the same fluctuation characteristics,  $\rho$  is data-adaptive and can be learned from the labeled dataset. Nevertheless, the segmentation is still primarily established on the recognition of turning patterns, which determines the segment number or lengths adaptively, and is essentially different from the principles of the existing segmentation methods.

### 4.2 Chebyshev Factorization

At the beginning, it is necessary to  $z$ -normalize the obtained subsequences as a pre-processing step. Rather than focusing on the statistical features, PCHA will factorize each subsequence with the first kind of Chebyshev polynomials, and take the Chebyshev coefficients as features. Since the Chebyshev polynomials with different degrees represent the fluctuation components, the local fluctuation information of time series can be captured in PCHA.

The first kind of Chebyshev polynomials are derived from the trigonometric identity  $T_n(\cos(\theta)) = \cos(n\theta)$ , which can be rewritten as a polynomial of variable  $t$  with degree  $n$ , as Formula (1).

$$T_n(t) = \begin{cases} \cos(n \cos^{-1}(t)), & t \in [-1, 1] \\ \cosh(n \cosh^{-1}(t)), & t \geq 1 \\ (-1)^n \cosh(n \cosh^{-1}(-t)), & t \leq -1 \end{cases} \quad (1)$$

For the sake of consistent approximation, we only employ the first sub-expression to factorize the subsequences, which is defined over the interval  $[-1, 1]$ . With the Chebyshev polynomials, a function  $F(t)$  can be factorized as Formula (2).

$$F(t) \cong \sum_{i=0}^n c_i T_i(t) \quad (2)$$

The approximation is exact if  $F(t)$  is a polynomial with the degree of less than or equal to  $n$ . The coefficients  $c_i$  can be calculated from the Gauss-Chebyshev Formula (3), where  $k$  is 1 for  $c_0$  and 2 for the other  $c_i$ , and  $t_j$  is one of the  $n$  roots of  $T_n(t)$ , which can be get from the formula  $t_j = \cos[(j - 0.5)\pi/n]$ .

$$c_i = \frac{k}{n} \sum_{j=1}^n F(t_j) T_i(t_j) \quad (3)$$

However, the employed Chebyshev polynomials are defined over the interval  $[-1, 1]$ . If the subsequences are factorized with this “interval function”, they must be scaled into the time interval  $[-1, 1]$ . Besides, the Chebyshev polynomials are defined everywhere in the interval, but time series is a discrete function, whose values are defined only at the sample moments. To compute the Chebyshev coefficients, we would process each subsequence with the method proposed in [18], which can extend time series into an interval function. Given a scaled subsequence  $S = \{(v_1, t_1), \dots, (v_m, t_m)\}$ , where  $-1 \leq t_1 < \dots < t_m \leq 1$ , we first divide the interval  $[-1, 1]$  into  $m$  disjoint subintervals as follows:

$$I_i = \begin{cases} [-1, \frac{t_1+t_2}{2}), i = 1 \\ [\frac{t_{i-1}+t_i}{2}, \frac{t_i+t_{i+1}}{2}), 2 \leq i \leq m - 1 \\ [\frac{t_{m-1}+t_m}{2}, 1], i = m \end{cases}$$

Then, the original subsequence can be extended into a step function as Formula (4), where each subinterval  $[t_i, t_{i+1}]$  is divided by the mid-point  $(t_i + t_{i+1})/2$ . The first half takes the value  $v_i$ , and the second half takes  $v_{i+1}$ .

$$F(t) = v_i, t \in I_i, 1 \leq i \leq m \quad (4)$$

After the above processing, the Chebyshev coefficients  $c_i$  can be computed. For the sake of dimension reduction, we only take the first several coefficients to approximate the raw data, which can reflect the principal fluctuation components of time series.

Figure 5 shows the examples of (a) PAA, (b) APCA, (c) PLA, and (d) PCHA representations for the stock time series of Google Inc. (symbol: GOOG) from The NASDAQ Stock Market, which consists of the close prices at 800 consecutive

trading days (2010/10/4-2013/12/5). As shown in Fig. 5(a), PAA extracts the mean values of the subsequences with equal-length as features. In Fig. 5(b), APCA takes the mean values and spans of the subsequences with adaptive-length as features, e.g.,  $[-0.62, 134]$  for the first subsequence. In Fig. 5(c), PLA takes the linear fitting slopes and spans of the subsequences with adaptive-length as features, e.g.,  $[-0.0035, 96]$  for the first subsequence. In Fig. 5(d), PCHA factorizes each subsequence and takes the first four Chebyshev coefficients as features, e.g.,  $[-3.8, 0.34, 3, -0.39]$  for the first subsequence. It is obvious that the approximation of PCHA is different from the others, which can well fit the local fluctuation characteristics of time series.

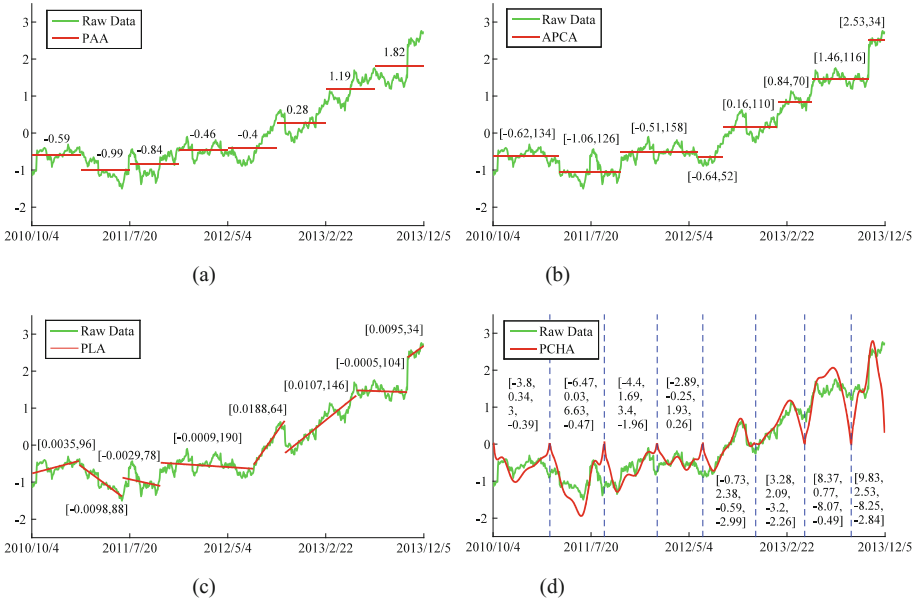


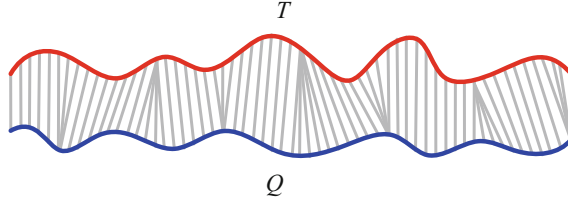
Fig. 5. PAA/APCA/PLA/PCHA representation examples.

In the entire procedure, the time series only needs to be scanned once for the adaptive segmentation and factorization. Thus, the computational complexity of PCHA is  $O(kn)$ , where  $k$  is the extracted Chebyshev coefficient number and much less than the time series length  $n$ .

## 5 Similarity Measure

DTW is one of the most prevalent similarity measures for time series [5]. It exploits the one-to-many aligning scheme to find the optimal alignment between time series, as shown in Fig. 6. Thus, DTW can deal with the intractable basic shape variations, e.g., time warping and phase-shift, etc. Given a sample space  $F$ , time series  $T = \{t_1, t_2, \dots, t_i, \dots, t_m\}$  and  $Q = \{q_1, q_2, \dots, q_j, \dots, q_n\}$ ,  $t_i, q_j \in F$ , a local distance measure  $d: (x, y) \rightarrow \mathbf{R}^+$  should be first set in DTW for measuring two samples. Then, a distance





**Fig. 6.** One-to-many aligning scheme of DTW.

matrix  $C \in \mathbf{R}^{m \times n}$  is computed, where each cell records the distance between each pair of samples from  $T$  and  $Q$  respectively, i.e.,  $C(i, j) = d(t_i, q_j)$ . There is an optimal warping path in  $C$ , which has the minimal sum of cells.

**Definition 4.** (Warping Path): Given the distance matrix  $C \in \mathbf{R}^{m \times n}$ , if the sequence  $p = \{c_1, \dots, c_l, \dots, c_L\}$ , where  $c_l = (a_l, b_l) \in [1: n] \times [1: m]$  for  $l \in [1: L]$ , satisfies the conditions that:

- (i)  $c_1 = (1, 1)$  and  $c_L = (m, n)$ ;
- (ii)  $c_{l+1} - c_l \in \{(1, 0), (0, 1), (1, 1)\}$  for  $l \in [1: L - 1]$ ;
- (iii)  $a_1 \leq a_2 \leq \dots \leq a_L$  and  $b_1 \leq b_2 \leq \dots \leq b_L$ ;

Then,  $p$  is called warping path. The sum of cells in  $p$  is defined as Formula (5).

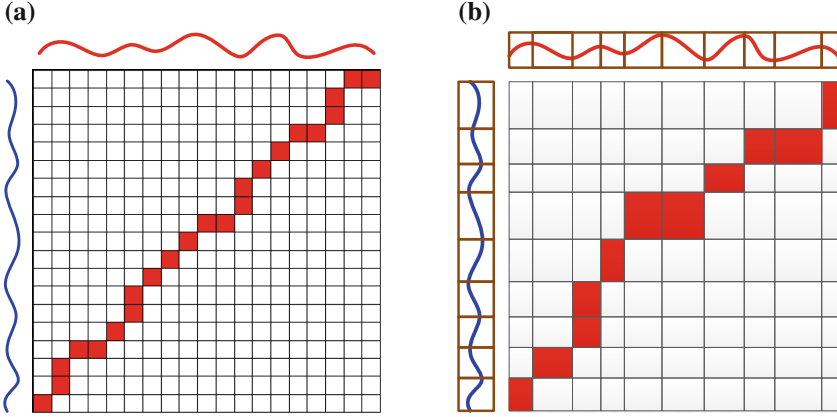
$$\Phi_p = C(c_1) + C(c_2) + \dots + C(c_L) \tag{5}$$

**Definition 5.** (Dynamic Time Warping Distance): Given the distance matrix  $C \in \mathbf{R}^{m \times n}$  over time series  $T$  and  $Q$ , and its warping path set  $\mathbf{P} = \{p_1, \dots, p_i, \dots, p_x\}$ ,  $i, x \in \mathbf{R}^+$ , the minimal sum of cells in the warping paths  $\Phi_{min} = \{\Phi_\xi \mid \Phi_\xi \leq \Phi_\lambda, \xi, \lambda \in \mathbf{P}\}$  is defined as the DTW distance between  $T$  and  $Q$ .

The computation of DTW performs with dynamic programming algorithm, which would lead to the quadratic computational complexity to the time series length, i.e.,  $O(n^2)$ . Figure 7(a) shows the dynamic programming table with the optimal warping path in DTW computation.

Based on PCHA, we propose a novel PA-DTW measure, named ChebyDTW, which contains two layers: subsequence matching and dynamic programming computation. Figure 7(b) shows the dynamic programming table with the optimal-aligned path (red shadow) of ChebyDTW, where each cell records the subsequence matching result over the Chebyshev coefficients. By the intuitive comparison with Fig. 7(a), ChebyDTW would have much lower computational complexity than the original DTW.

With high computational efficiency, the squared Euclidean distance is a proper measure for the subsequence matching. Given  $d$  Chebyshev coefficients are employed in PCHA, for the subsequences  $S_1$  and  $S_2$ , respectively approximated as  $C = [c_1, \dots, c_d]$  and  $\hat{C} = [\hat{c}_1, \dots, \hat{c}_d]$ , the squared Euclidean distance between them can be computed as Formula (6).



**Fig. 7.** (a) Dynamic programming table with the optimal-aligned path (red shadow) of DTW, (b) against that of ChebyDTW. (Color figure online)

$$D(\mathbf{C}, \hat{\mathbf{C}}) = \sum_{i=1}^d (c_i - \hat{c}_i)^2 \quad (6)$$

Over the subsequence matching, the dynamic programming computation performs. Given that time series  $T$  with length  $m$  is segmented into  $M$  subsequences, and time series  $Q$  with length  $n$  is segmented into  $N$  subsequences, ChebyDTW can be computed as Formula (7).  $\mathbf{C}^T$  and  $\mathbf{C}^Q$  are the PCHA representations of  $T$  and  $Q$  respectively;  $\mathbf{C}_1^T$  and  $\mathbf{C}_1^Q$  are the first coefficient vectors of  $\mathbf{C}^T$  and  $\mathbf{C}^Q$  respectively;  $rest(\mathbf{C}^T)$  means the rest coefficient vectors of  $\mathbf{C}^T$  except for  $\mathbf{C}_1^T$ ; the same meaning is taken for  $rest(\mathbf{C}^Q)$ .

$$ChebyDTW(T, Q) = \begin{cases} 0, & \text{if } m = n = 0 \\ \infty, & \text{if } m = 0 \text{ or } n = 0 \\ D(\mathbf{C}_1^T, \mathbf{C}_1^Q) + \min \left\{ \begin{array}{l} ChebyDTW[rest(\mathbf{C}^T), \mathbf{C}^Q], \\ ChebyDTW[\mathbf{C}^T, rest(\mathbf{C}^Q)], \\ ChebyDTW[rest(\mathbf{C}^T), rest(\mathbf{C}^Q)] \end{array} \right\}, & \text{otherwise} \end{cases} \quad (7)$$

## 6 Experiments

We evaluate the 1NN classifier based on ChebyDTW from the aspects of accuracy and efficiency respectively. 12 real-world datasets provided by the UCR time series archive [17] are employed, which come from the various application domains and are characterized by the different series profiles and dimensionality. All the datasets have been

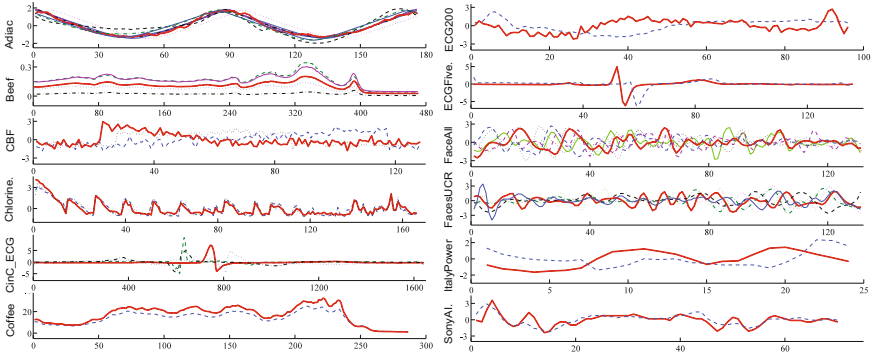


Fig. 8. Sample representative instances from each class of 12 datasets.

z-normalized and partitioned into the training and testing sets by the provider. Figure 8 shows the sample representative instances from each class of the datasets.

All parameters in the measures are learned on the training datasets by the DIRECT global optimization algorithm [19], which is used to seek for the global minimum of multivariate function within a constraint domain. The experiment environment is Intel (R) Core(TM) i5-2400 CPU @ 3.10 GHz; 8G Memory; Windows 7 64-bit OS; MATLAB 8.0\_R2012b.

### 6.1 Classification Accuracy

Firstly, we take four PA-DTWs based on the statistical features as baselines, i.e., PDTW [14], SDTW [10],  $DTW_{APCA}$  [8], and  $DTW_{DSA}$  [9], which are based on PAA, PLA, APCA, and DSA representations respectively. Secondly, since PA-DTW is computed over the approximate representation, its precision is regarded lower than the measures computed on the raw data. To test this assumption, we also take 4 DTW measures computed on the raw data as baselines, including the original DTW and its variations, i.e., CDTW [3], CIDDTW [20], DDTW [21].

Tables 1 and 2 present the 1NN classification accuracy based on ChebyDTW and the baselines respectively. The best results on each dataset are highlighted in bold. The learned parameters are also presented, which could make each classifier achieve the highest accuracy on each training dataset, including the segment threshold ( $\rho$ ), the smooth degree ( $sd$ ), and the extracted Chebyshev coefficient number ( $\theta$ ). For the sake of dimension reduction, we learn the parameter  $\theta$  in the range of [1, 10] for ChebyDTW.

By the comparison, we find that, (1) the 1NN classifier based on ChebyDTW wins all datasets over that based on the PA-DTW baselines. The superiority mainly derives from the distinctive features extracted in ChebyDTW, which can capture the fluctuation information for similarity measure. Concretely, as shown in Fig. 8, the practical time series in the datasets have the relatively complicated fluctuation that can be transformed into the wide Chebyshev domain, thus the difference between time series can be easily captured by the Chebyshev coefficients. Whereas the statistical features extracted in the baselines only focus on the aggregation characteristics of time series, which would result in much fluctuation information loss.

**Table 1.** The accuracy of INN classifiers based on ChebyDTW and four PA-DTW baselines.

Dataset	$\rho$	$sd$	$\theta$	Cheby DTW	PDTW	SDTW	DTW <sub>APCA</sub>	DTW <sub>DSA</sub>
Adiac	0.21	22	9	<b>0.72</b>	0.61	0.34	0.28	0.38
Beef	0.18	17	5	<b>0.57</b>	0.50	0.57	0.57	0.47
CBF	0.98	8	10	<b>0.98</b>	<b>0.98</b>	0.95	0.91	0.50
Chlorine.	0.73	25	8	<b>0.65</b>	0.60	0.55	0.56	0.62
CinC_ECG.	0.29	4	9	<b>0.81</b>	0.65	0.63	0.61	0.63
Coffee	0.51	14	9	<b>0.89</b>	0.79	0.75	0.82	0.61
ECG200	0.80	7	9	<b>0.89</b>	0.80	0.83	0.77	0.81
ECGFive.	0.73	17	9	<b>0.91</b>	0.79	0.68	0.68	0.57
FaceAll	0.51	29	10	<b>0.73</b>	0.63	0.50	0.63	0.71
FacesUCR	0.51	4	6	<b>0.80</b>	0.60	0.57	0.72	0.70
ItalyPower.	0.51	7	5	<b>0.94</b>	0.93	0.80	0.90	0.87
SonyAI.	0.95	25	6	<b>0.80</b>	0.76	0.73	0.76	0.70

**Table 2.** The accuracy of INN classifiers based on ChebyDTW and four DTW baselines.

Dataset	Cheby DTW	DTW	CDTW	CIDDTW	DDTW
Adiac	<b>0.72</b>	0.60	0.61	0.61	0.47
Beef	<b>0.57</b>	0.50	0.53	0.50	0.47
CBF	0.98	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	0.54
Chlorine.	0.65	0.65	0.65	0.64	<b>0.69</b>
CinC_ECG.	0.81	0.65	<b>0.93</b>	0.70	0.66
Coffee	<b>0.89</b>	0.82	0.82	0.82	0.79
ECG200	<b>0.89</b>	0.77	0.88	0.81	0.80
ECGFive.	<b>0.91</b>	0.77	0.80	0.76	0.65
FaceAll	0.73	0.81	0.81	<b>0.85</b>	0.80
FacesUCR	0.80	0.90	<b>0.91</b>	0.83	0.76
ItalyPower.	0.94	0.91	0.91	0.88	<b>0.96</b>
SonyAI.	0.80	0.95	<b>0.96</b>	0.92	0.87

(2) The classifier based on ChebyDTW has higher accuracy on more datasets than the original DTW and its variations. The reason is apparent that, the noise mixed in the time series can be filtered out by the Chebyshev factorization effectively, which is one of the principal factors affecting the precision of similarity measures. Thus, the above assumption that ChebyDTW has lower precision than the measures computed on the raw data is not supported.

## 6.2 Computational Efficiency

The speedup of computational complexity gained by PA-DTW over the original DTW is  $O(n^2/w^2)$ , where  $n$  is the time series length and  $w$  is the segment number. It is positively correlated with the data compression rate ( $DCR = n/w$ ) of piecewise

approximation over the raw data. In Table 3, we present the segment numbers and the DCRs of five PA-DTWs on all datasets. As above, the optimal segment numbers for the INN classifiers based on PDTW, SDTW, and DTW<sub>APCA</sub> are learned on the training datasets, while the average segment numbers on each dataset are computed for ChebyDTW and DTW<sub>DSA</sub>.

As shown in Table 3, the DCRs of ChebyDTW are not only much larger than the baselines on all datasets, but also robust to the time series length. Thus, it has the highest computational efficiency among the five PA-DTWs. The efficiency superiority of ChebyDTW mainly derives from the precise approximation of PCHA over the raw data, and the data-adaptive segment method, which can segment time series into the less number of subsequences with the adaptive lengths.

**Table 3.** The DCR results of five PA-DTWs.

Dataset	$n$	Cheby DTW		PDTW		SDTW		DTW <sub>APCA</sub>		DTW <sub>DSA</sub>	
		$w$	DCR	$w$	DCR	$w$	DCR	$w$	DCR	$w$	DCR
Adiac	176	3.99	<b>44.13</b>	36	4.89	13	13.54	43	4.10	70	2.51
Beef	470	5.18	<b>90.68</b>	61	7.70	10	47	61	7.70	192.32	2.44
CBF	128	1	<b>128.0</b>	30	4.27	27	4.74	15	8.53	46.19	2.77
Chlorine.	166	2	<b>83.00</b>	36	4.61	29	5.72	34	4.88	64.77	2.56
CinC.	1639	4	<b>409.9</b>	103	15.91	94	17.44	84	19.51	655.49	2.50
Coffee	286	2	<b>143.0</b>	60	4.77	33	8.67	40	7.15	117.34	2.44
ECG200	96	1.93	<b>49.74</b>	14	6.86	19	5.05	23	4.17	35.84	2.68
ECGFive.	136	1.61	<b>84.43</b>	9	15.11	9	15.11	5	27.20	48.24	2.82
FaceAll	131	2	<b>65.50</b>	32	4.09	32	4.09	32	4.09	53.96	2.43
FacesUCR	131	2	<b>65.50</b>	24	5.46	32	4.09	31	4.23	54.43	2.41
ItalyPower.	24	1.98	<b>12.13</b>	5	4.80	6	4.00	6	4.00	10.61	2.26
SonyAI.	70	1	<b>70.00</b>	13	5.38	9	7.78	8	8.75	27.41	2.55

**Table 4.** The average runtime of INN classification based on DTW and ChebyDTW (ms).

Dataset	DTW	ChebyDTW	$\Omega$
Adiac	172.21	5.36	32.11
Beef	105.02	0.54	194.95
CBF	9.34	0.37	25.40
Chlorine.	231.23	5.47	42.24
CinC_ECG	1721.93	0.79	2175.49
Coffee	38.17	0.40	96.05
ECG200	16.15	1.18	13.69
ECGFive.	6.89	0.29	23.60
FaceAll	169.94	8.26	20.57
FacesUCR	61.61	2.37	26.00
ItalyPower.	1.3	0.73	1.78
SonyAI.	2.1	0.25	8.38

In addition, the average runtime of 1NN classification based on DTW and ChebyDTW are presented in Table 4. According to the results, the efficiency speedup ( $\Omega$ ) of ChebyDTW over DTW can achieve as much as 3 orders of magnitude.

## 7 Conclusions

We proposed a novel piecewise factorization model for time series, i.e., PCHA, where a novel adaptive segment method was proposed, and the subsequences were factorized with the Chebyshev polynomials. We employed the Chebyshev coefficients as features for PA-DTW measure, and thus proposed the ChebyDTW for 1NN classification. The comprehensive experimental results show that ChebyDTW can support the accurate and fast 1NN classification.

**Acknowledgements.** This work was funded by the Ministry of Industry and Information Technology of China (No. 2010ZX01042-002-003-001), China Knowledge Centre for Engineering Sciences and Technology (No. CKCEST-2014-1-5), and National Natural Science Foundation of China (No. 61332017).

## References

1. Esling, P., Agon, C.: Time-series data mining. *ACM Comput. Surv.* **45**(1), 12 (2012)
2. Fu, T.: A review on time series data mining. *Eng. Appl. Artif. Intell.* **24**(1), 164–181 (2011)
3. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. In: *Proceedings of the VLDB Endowment*, pp. 1542–1552 (2008)
4. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. *Data Min. Knowl. Disc.* **28**(4), 851–881 (2014)
5. Serra, J., Arcos, J.L.: An empirical evaluation of similarity measures for time series classification. *Knowl.-Based Syst.* **67**, 305–314 (2014)
6. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. *Knowl. Inf. Syst.* **3**(3), 263–286 (2001)
7. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: a survey and novel approach. *Data Min. Time Ser. Databases* **57**, 1–22 (2004)
8. Chakrabarti, K., Keogh, E., Mehrotra, S., Pazzani, M.: Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Trans. Database Syst.* **27**(2), 188–228 (2002)
9. Gullo, F., Ponti, G., Tagarelli, A., Greco, S.: A time series representation model for accurate and fast similarity detection. *Pattern Recognit.* **42**(11), 2998–3014 (2009)
10. Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping to massive datasets. In: Żytkow, J.M., Rauch, J. (eds.) *PKDD 1999. LNCS (LNAI)*, vol. 1704, pp. 1–11. Springer, Heidelberg (1999). doi:[10.1007/978-3-540-48247-5\\_1](https://doi.org/10.1007/978-3-540-48247-5_1)
11. Li, H., Guo, C.: Piecewise cloud approximation for time series mining. *Knowl.-Based Syst.* **24**(4), 492–500 (2011)
12. Lin, J., Vlachos, M., Keogh, E., Gunopulos, D., Liu, J., Yu, S., Le, J.: A MPAA-based iterative clustering algorithm augmented by nearest neighbors search for time-series data streams. In: Ho, T.B., Cheung, D., Liu, H. (eds.) *PAKDD 2005. LNCS (LNAI)*, vol. 3518, pp. 333–342. Springer, Heidelberg (2005). doi:[10.1007/11430919\\_40](https://doi.org/10.1007/11430919_40)

13. Rakthanmanon, T., Campana, B., Mueen, A.: 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 (2012)
14. Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping for data mining applications. In: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 285–289 (2000)
15. Ullman, S., Poggio, T.: Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. MIT Press, Cambridge (1982)
16. Gao, J., Sultan, H., Hu, J., Tung, W.: Denoising nonlinear time series by adaptive filtering and wavelet shrinkage: a comparison. *IEEE Sig. Process. Lett.* **17**(3), 237–240 (2010)
17. Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei, L., Ratanamahatana, C. A.: UCR time series classification/clustering (2011). [www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/)
18. Cai, Y., Ng, R.: Indexing spatio-temporal trajectories with Chebyshev polynomials. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, pp. 599–610 (2004)
19. Björkman, M., Holmström, K.: Global optimization using the DIRECT algorithm in matlab. *Adv. Model. Optim.* **1**(2), 17–37 (1999)
20. Batista, G.E., Keogh, E.J., Tataw, O.M., Souza, V.M.: CID: an efficient complexity-invariant distance for time series. *Data Min. Knowl. Disc.* **28**(3), 634–669 (2014)
21. Keogh, E.J., Pazzani, M.J.: Derivative dynamic time warping. In: Proceedings of SDM, pp. 5–7 (2001)