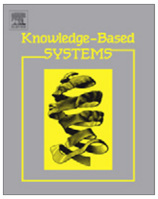
Knowledge-Based Systems 24 (2011) 492–500

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Piecewise cloud approximation for time series mining

Hailin Li, Chonghui Guo [*]

Institute of Systems Engineering, Dalian University of Technology, Dalian 116024, China

ARTICLE INFO

Article history:
Received 6 July 2010
Received in revised form 19 December 2010
Accepted 20 December 2010
Available online 28 December 2010

Keywords:
Piecewise cloud approximation
Time series mining
Dimensionality reduction
Cloud model
Time series representation

ABSTRACT

Many researchers focus on dimensionality reduction techniques for the efficient data mining in large time series database. Meanwhile, corresponding distance measures are provided for describing the relationships between two different time series in reduced space. In this paper, we propose a novel approach which we call piecewise cloud approximation (PWCA) to reduce the dimensionality of time series. This representation not only allows dimensionality reduction but also gives a new way to measure the similarity between time series well. Cloud, a qualitative and quantitative transformation model, is used to describe the features of subsequences of time series. Furthermore, a new way to measure the similarity between two cloud models is defined by an overlapping area of their own expectation curves. We demonstrate the performance of the proposed representation and similarity measure used in time series mining tasks, including clustering, classification and similarity search. The results of experiments indicate that PWCA is an effective representation for time series mining.

1. Introduction

Time series is a common kind of data stored in financial, educational, medical and meteorological database. Many researchers have been focusing on the knowledge and information hiding in time series. Meanwhile, various techniques and algorithms were developed to mine time series, including decision tree [1], rough set [2], fuzzy mathematics [3,4], neural network [5] and statistic analysis [6]. Various tasks are applied to mine time series [7,8], such as clustering, classification, forecast, association rule and indexing, which can discover the valuable knowledge and information from a large amount of time series data.

It is well known that dimensionality curse destructively impacts on the result of time series data mining, so we need to address the problem of high dimensionality. The most promising methods to alleviate the pressure of dimensionality curse in time series database are the techniques of dimensionality reduction. So far, there are many methods used to reduce the high dimensionality of time series, such as discrete fourier transform (DFT) [9], discrete wavelet transform (DWT) [10], singular value decomposition (SVD) [11], piecewise aggregate approximation (PAA) [12] and piecewise linear approximation (PLA) based on line segments [13]. In addition, some symbolic representations are also used to do the work such as symbolic aggregate approximation (SAX) base on PAA [14], which transforms the mean values deriving from PAA into discrete symbolic strings. All the above mentioned techniques have an ability of representing time series in common. In the reduced dimensionality, the corresponding representations are simple and are conveniently used to mine the valuable knowledge and information.

If time series are transformed into any of the above representations, then it is possible to measure the similarity or distance between two time series in the reduced space. Different representations employ their own distance measure for time series mining. Especially for indexing time series fast, distance measure used in the reduced space should be guaranteed to lower bound the true distance measure between two original time series. It can keep indexing without any false dismissals [15]. Unfortunately, most of the above mentioned techniques can't find a remarkable way to compare between the estimated distance and the true one.

It is well known that SAX is more superior in many aspects. It not only validly reduces the high dimensionality but also produces effective results of time series mining. One of the most important merits is that it is a better approximation to mine time series in reduced space than the existing techniques, such as DWT, DFT and SVD, but its performance is degraded in much lower dimensionality. In other words, the larger the compression ratio is, the worse the performance of SAX may be. The goal of this paper is to develop a technique which can not only approximate time series well but also deal with the degradation of the algorithms' performance in the much lower space. It means that the technique should be efficient and effective in the much lower space as good as in the high one, even better than the traditional ones in the much lower space. This proposed technique to represent time series for dimensionality

[*] Corresponding author. Tel.: +86 41184708007.
E-mail addresses: hailin@mail.dlut.edu.cn (H. Li), guochonghui@tsinghua.org.cn (C. Guo).

0950-7051/$ - see front matter Crown Copyright © 2010 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2010.12.008

reduction is piecewise cloud approximation (PWCA) based on cloud which is a quantitative and qualitative transformation model [16]. First, we need to partition a time series into some equal-length "frames" for approximation. Second, we transform each "frame" into cloud model and finally obtain a cloud sequences for a time series. After the cloud representations, we propose a new method to calculate the similarity between two cloud models to further describe relationships between two time series.

At least, there are two advantages existing in our approach. One is that three numerical characteristics of a cloud model to represent one subsequence of time series retain more information in the reduced space than the existing methods such as PAA and SAX, so our approach can resolve the problem appeared in PAA and SAX (see Section 6.1). The other is that our approach can measure the similarity between time series well in much lower space, which relieves the limitation of the traditional methods. It means that our approach does well in representing time series under a high compression. Thereby, PWCA is a more effective approach for time series mining while the reduced dimensionality is much lower. We demonstrate the performance of the proposed approximation for time series mining tasks including clustering, classification, and similarity search.

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 briefly introduces the theory of cloud model. Section 4 proposes our novel piecewise cloud approximation and Section 5 gives a new similarity measure between two cloud models. Section 6 contains some experiments and their evaluations. In the last section we offer some conclusions and suggestions of future work.

2. Related work

It is well known that high dimensionality is one of the most difficult issues for time series mining. To address this problem, some techniques of dimensionality reduction are used and their own representations in reduced space are efficient to approximate time series.

The first technique of dimensionality reduction is discrete wavelet transform (DWT) [10], which can be used to represent time series in terms of a fast decaying and discretely sampled waveform (such as mother wavelet). The basic idea of DWT to represent time series of length n is a choice of n wavelet coefficients. Generally speaking, dimensionality reduction in DWT is achieved by retaining the first p coefficients, where $p \ll n$. However, discrete fourier transform (DFT) [9] is another approach to reduce the dimensionality of time series. It transforms a signal or a time series from time domain to frequency domain. Moveover, Euclidean distance between two time series in the time domain can be preserved in the frequency domain. It is quite close to DWT because a set of orthogonal functions is used. DWT and DFT differ in several aspects. One of the most obvious differentiae is that DWT is localized in time which means that some of wavelet coefficients represent small and local subsection of time series, but in DFT the fourier coefficients always represent global contributions to time series.

Singular value decomposition (SVD) [11] consists of space rotation and truncation and is an optimal transformation in some senses. However, the drawbacks of SVD limit its applications. The most important one is its complexity, including space complexity $O(mn)$ and time complexity $O(mn)$. Additionally, inserting a new object into the dataset should require recomputing the entire procedure. The reason is that SVD considers all time series in the dataset other than one time series at a time.

Piecewise linear approximation (PLA) represents time series with some line segments. There are many kinds of PLA [13] to reduce the dimensionality of time series. They can be grouped into three classes, window-sliding, top-down and bottom-up. Most of the algorithms have a low computational complexity which is linear to the length of time series, but some have a high time complexity [17,18] because they are in pursuit of optimal results.

SAX transforms an original time series of length n into a discrete symbolic string. The whole process includes two phases. The first, piecewise aggregate approximation (PAA), is a process of high dimensionality reduction for time series. In this process, time series is divided into w equal-length "frames", of which the length is k ($k = n/w$) and the values can be respectively represented by the mean of data points within the corresponding frame. The time complexity of PAA for measuring the similarity is linear to the length of time series, i.e., $O(n)$. The representative value of the original time series in reduced space is shown in Fig. 1(a). The second process is symbolization. It is a transformation from mean values to a discrete symbolic string according to the equiprobably divided regions, of which the number is *alphabet_size*(as). Fig. 1(b) shows the result of SAX after PAA procedure. SAX has two important advantages, dimensionality reduction and lower bounding [14,22]. In particular, the extended SAX (ESAX) [19] improves the representation of time series. ESAX not only considers the mean of every segment which obtained by PAA but also takes the maximum and the minimum of each segment into consideration.

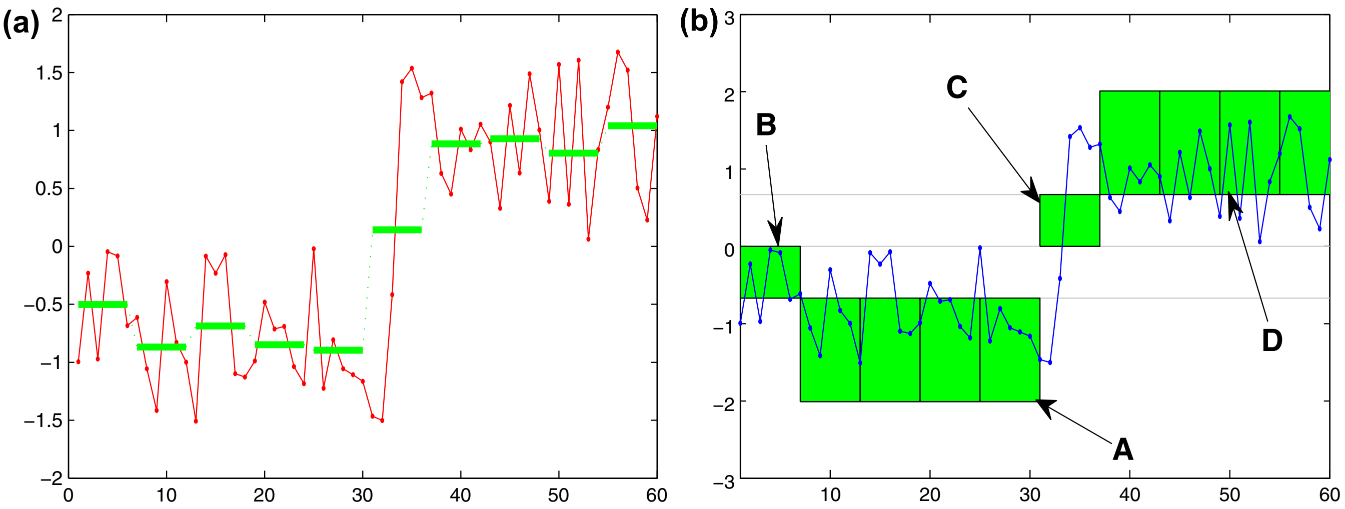


Fig. 1. (a) New values in lower space by PAA and (b) symbolic representation of time series by SAX.

In addition to SAX, other symbolizations are also applied to represent time series for special motivations. SDA [20] computes the changes between values from one point to the next point, and divides the range of changes into some regions which depend on artificial preset. It means that prior knowledge of the data distribution is important for the region division. Another method IMPACTS proposed by Huang and Yu [21] transforms time series into symbols by dividing the change ratio (between two adjacent points) into equal-sized sections and mapping each region to a symbol. The IMPACTS often exploits the suffix tree structure and is provided to process pattern matching queries on time series. However, the authors [14] had stated that SDA and IMPACTS are not superior to SAX due to smoothing effect of dimensionality reduction.

All the above mentioned approaches can approximate time series and are often used to mine time series, but in the much lower space (after reducing the dimensionality) they often don't have a good performance of representation for time series. Since high compress ratio causes a much more loss, a technique used to approximate time series, which can retain the information of time series as much as possible, is required. Thereby, our approach based on cloud model is proposed to resolve the problem.

3. Cloud model

Cloud proposed by Li [23] is used to describe the uncertainty of concepts. It is well known that concepts in natural languages are qualitative, and understanding the concepts by human is uncertain, which is often embodied in randomness and fuzziness [24]. Cloud is a model to address the relationship between randomness and fuzziness for qualitative concepts and quantitative values. In other words, cloud model is a significant approach to establish an uncertain transformation between the qualitative concepts and the quantitative description. Usually, we call the model cloud for short. It has been widely used in various fields, such as data mining [25], control [26] and decision [27].

The concept of abstract cloud in this paper seems to be the one in nature, which is composed of many cloud drops. Now we define the abstract cloud and cloud drops for describing the concept [16].

Definition 1. Let U be a universal set described by precise numbers, C be the qualitative concept related to U. If there is a number $x \in U$, which is the value of randomly realizing the concept C, and the certainty degree of x for C, i.e., $y = \mu_C(x) \in [0,1]$, is a random value with stabilization tendency, then the distribution of x on U is regard as a cloud and each point $[x,y]$ is defined as a cloud drop.

In this definition there is a random realization, which is the realization in terms of probability. Similarly, the certainty degree is the membership degree of fuzzy set and has the probability distribution. All these demonstrate the correlation of fuzziness and randomness. The detailed definition can be seen in Definition 3.

Cloud model is used to describe the concept. The overall property of a concept can be represented by the numerical characteristics of cloud. In cloud model, let a vector V consisted of three numerical characteristics represent a concept as a whole, i.e., $V = (Ex, En, He)$.

Definition 2. $V = (Ex, En, He)$ is a cloud model. As shown in Fig. 2, Ex is the expected value, which is a mathematical expectation of the cloud drops. In other words, it is one of the most representative drops for the qualitative concept. En is an entropy, which is an uncertainty measurement of the qualitative concept, and reflects the dispersing extent of the cloud drops and the acceptable values in the region. He is a hyper-entropy, which is an uncertainty measurement of the entropy En, and reflects the thickness of the cloud.

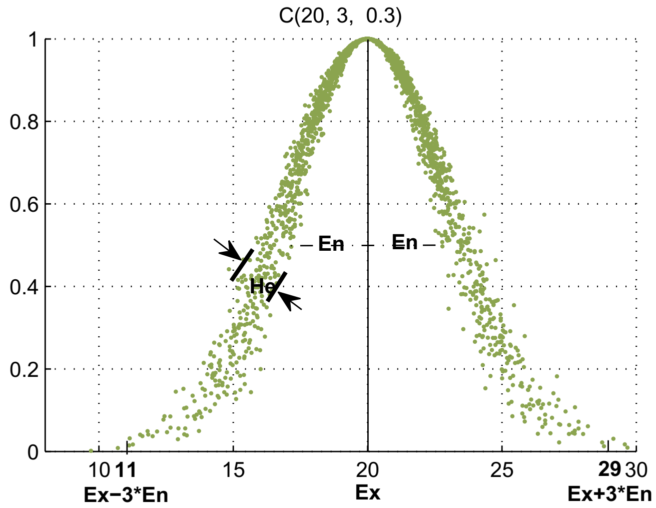


Fig. 2. The numerical characteristic vector of a cloud is $V = (20,3,0.1)$. $Ex = 20$ is the expected value, $En = 3$ reflects the dispersing extent of the cloud drops and $He = 0.1$ represents the thickness of cloud. Most of the cloud drops are in the range $[Ex - 3En, Ex + 3En]$.

We can obtain different kinds of clouds according to various implementation approaches, such as symmetric cloud model, half cloud model and combined cloud model. However, the most common and important one is the normal cloud model. Normal distribution is one of the most important distributions and often uses two characteristics, the mean and the variance, to describe data. At the same time, the bell-shaped membership function is also one of most useful functions in fuzzy set to reflect certainty degree. Therefore, the normal cloud model based on these two cases is very useful to describe the relationships between the qualitative concepts and the quantitative values.

Definition 3. Let U be a quantitative universal and C be the qualitative concept related to U. If $x \in U$ is a random realization of the concept C and x satisfies $x \sim N(Ex, En'^2)$, where $En' \sim N(En, He^2)$, $En \neq 0$ and the certainty degree of x on C is $\mu = e^{-\frac{(x-Ex)^2}{2(En')^2}}$, then the distribution of x on U is a normal cloud and the "expectation curve" of normal cloud is

$$y = e^{-\frac{(x-Ex)^2}{2En^2}}. \quad (1)$$

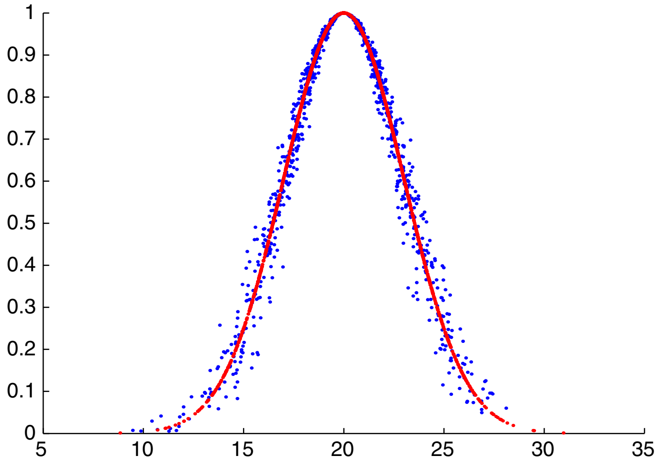


Fig. 3. The expectation curve of a cloud.

As shown in Fig. 3, the expectation curve goes through the center of cloud drops smoothly and shows the overall outline of the cloud. All the cloud drops fluctuate randomly around it.

To denote the qualitative concept by the quantitative method, the "forward cloud generator" is developed to transform the qualitative concept into quantitative values. The reverse one is "backward cloud generator", which obtains three numerical characteristics from the cloud drops or a vector with many real values. The algorithms of forward normal cloud generator and backward normal cloud generator [16] are described as follows.

```
Forward normal cloud generator ($cloud(Ex,En,He,n)$):
    Input: Cloud model $V = (Ex,En,He)$ and the number of cloud drops, $n$.
    Output: A matrix of cloud drops, $[x,y]$.
    Step 1. Set a variable $i = 1$ initially.
    Step 2. $En' = randn(1)He + En$, where $randn(1)$ produces a random value of standard normal distribution.
    Step 3. $x(i) = randn(1)En' + Ex$.
    Step 4. $y(i) = e^{-\frac{(x(i)-Ex)^2}{2(En')^2}}$.
    Step 5. If $i \neq n$, then $i = i + 1$ and go back to step 2. Otherwise, stop this procedure and return the matrix of cloud drops.
Backward normal cloud generator ($back\_cloud(X)$):
    Input: A vector $X = (x_1,x_2,\ldots,x_n)$.
    Output: A cloud model with three numerical characteristics $V = (Ex,En,He)$.
    Step 1. $Ex = mean(X)$, where $mean(\cdot)$ is the mean function.
    Step 2. $En = mean(|X - Ex|)\sqrt{\frac{\pi}{2}}$.
    Step 3. $He = \sqrt{var(X) - En^2}$, where $var(\cdot)$ is the variance function.
```

Note that in normal cloud model the contribution of each cloud drop to its corresponding concept is different. The cloud drops contributing to a concept in the universal domain U mainly lie in the range $[Ex - 3En, Ex + 3En]$, which we call "$3En$ rule" of normal cloud similarly to the "3σ rule" of the normal distribution.

4. Piecewise cloud approximation

There is a time series $Q = \{q_1, q_2, \ldots, q_m\}$, and $Q(i:j) = \{q_i, \ldots, q_j\}$ denotes a subsequence of Q, where $i < j$. If time series Q can be represented in a lower space of dimension w by $Q' = \{q'_1, q'_2, \ldots, q'_w\}$, then the compress ratio is $k = m/w$, where $q'_i = back\_cloud(Q(k(i-1)+1 : ki))$. $back\_cloud(\cdot)$ is the backward normal cloud generator, and q'_i with three characteristics $[Ex, En, He]$ can be used to produce n cloud drops by the forward normal cloud generator, i.e., $cloud(q'_i, n)$. These cloud drops compose the cloud. Simply stated, to reduce the dimensionality of time series from dimension m to w, the time series should be divided into w equal-length "frames" like that of PAA. Each "frame" can be transformed into a cloud with three characteristics by the backward normal cloud generator. Inversely, its cloud drops can be obtained by forward normal cloud generator. The original time series is represented by w clouds as shown in Fig. 4.

Fig. 4 shows that PWCA is very simple and intuitive. It not only reduces the dimensionality but also reflects the feature of the distribution of the data points within each "frame". For example, q'_2 is the cloud of the subsequence $Q(11{:}20)$ whose elements are close to the mean of the "frame". The entropy En and the hyper-entropy He of q'_2 are smaller than that of q'_4 within the subsequence $Q(31{:}40)$ of which the elements crazily deviate from the mean of its corresponding "frame". Therefore, PWCA, which could reflect the distribution of the data points, is a more effective technique to reduce the dimensionality than the traditional ones, such as PAA and SAX [22].

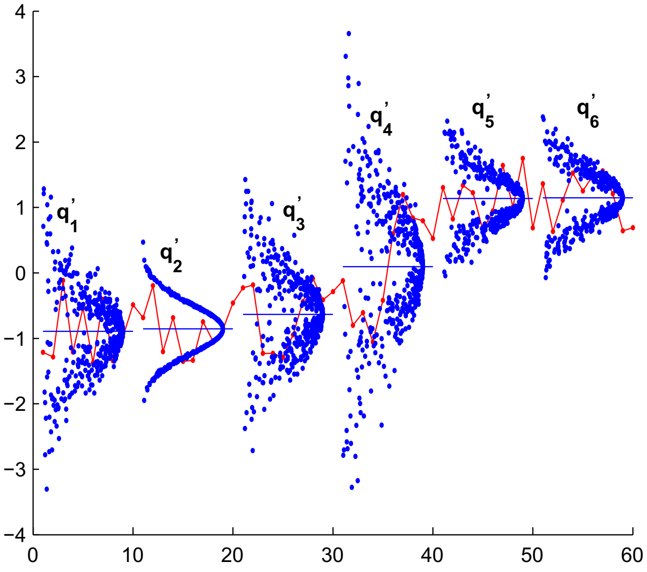


Fig. 4. One of PWCA representations is a cloud model of each "frame". A time series of length 60 can be reduced into 6 cloud models.

PWCA can transform a time series of length m into w cloud models. Moreover, considering the randomness and fuzziness, each cloud model with three characteristics is able to describe the feature of the data points' distribution within its corresponding "frame". Inversely, if there are w cloud models describing a time series of length m, each cloud model can be used to produce k cloud drops by forward normal cloud generator to approximate the subsequence within the "frame". As shown in Fig. 5, the cloud drops produced by 6 cloud models can approximate the trend of the original time series and reflect the distribution of data points within each "frame". In addition, to compare different time series reasonably before dimensionality reduction, we should normalize the time series into a new one with a mean of 0 and a standard deviation of 1 as SAX does [14].

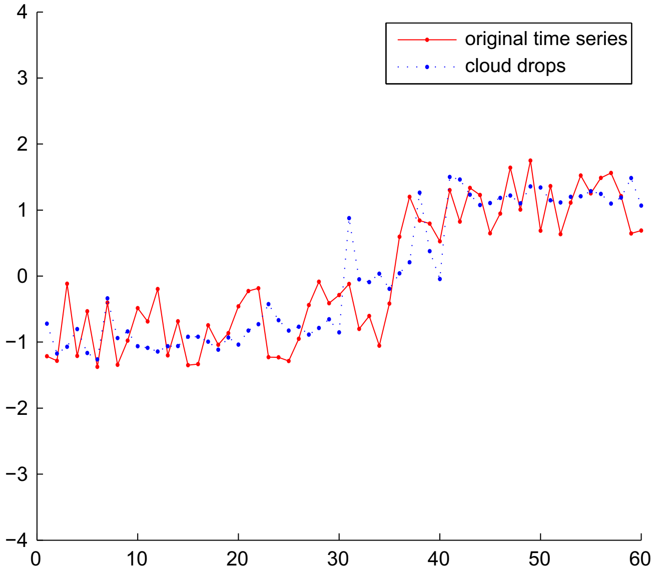


Fig. 5. The 6 cloud models produce 60 data points to approximate the original time series, which reflects the distribution of data points.

5. Similarity measure of time series based on cloud models

After dimensionality reduction by PWCA, a similarity or distance measure should be used to describe the relationship between two different time series. Euclidean distance is one of the most important and common measure functions. However, since the characteristics of each cloud model in reduced space have different significance and units, it is unreasonable to use Euclidean distance to calculate the similarity measure between two cloud models. Therefore, according to the particularity of cloud model, we propose a similarity measure to describe their relationship objectively.

Given two cloud models, $V_1 = (Ex_1, En_1, He_1)$ and $V_2 = (Ex_2, En_2, He_2)$. The similarity measure between two cloud models can be denoted as $ECM(V_1, V_2)$. To calculate the similarity objectively, expectation curve of the cloud model is used. In other words, the similarity measure denoted by $ECM(V_1, V_2)$ is based on the expectation curves of two cloud models.

We know that a cloud graph $[x, y]$ produced by the forward normal cloud generator is rich in geometrical properties. However, the expectation curve could reflect the overall properties of cloud geometric shape [16] and could be used to approximately represent the corresponding cloud model. Therefore, the similarity of the two expectation curves can be regard as the similarity of the two cloud models and is indicated by the overlapping area S as shown in Fig. 6. The more similar the two cloud models are, the larger the overlapping area will be. If the two expectation curves intersect at x_0, then the shaded part of the area S can be calculated by the sum of $\int_{-\infty}^{x_0} y_2(x)dx$ and $\int_{x_0}^{\infty} y_1(x)dx$, i.e.,

$$S = \int_{-\infty}^{x_0} y_2(x)dx + \int_{x_0}^{\infty} y_1(x)dx, \tag{2}$$

where $y_1(x)$ and $y_2(x)$ are the expectation curves of cloud V_1 and V_2, respectively. Unfortunately, since the formula (1) is non-integrable, these two definite integrals in formula (2) have no analytic expression. Moreover, the numerical integration method is not suitable to calculate the similarities of a large number of cloud models because of the heavy computational time.

We transform the formula (1) into

$$y = \sqrt{2\pi}En\frac{1}{\sqrt{2\pi}En}e^{-\frac{(x-Ex)^2}{2En^2}} = \sqrt{2\pi}Enf(x), \tag{3}$$

where $f(x)$ is the probability density function of the normal distribution. Therefore, S can be rewritten as

$$S = \int_{-\infty}^{x_0} \sqrt{2\pi}En_2 f_2(x)dx + \int_{x_0}^{\infty} \sqrt{2\pi}En_1 f_1(x)dx. \tag{4}$$

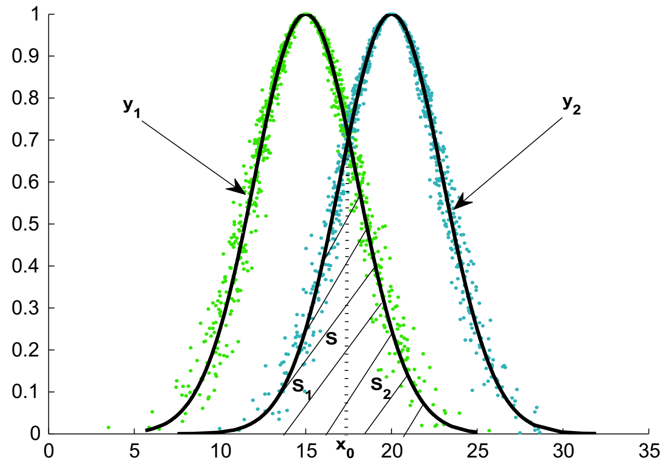


Fig. 6. The overlapping area S indicates the similarity between the two cloud models, V_1 and V_2.

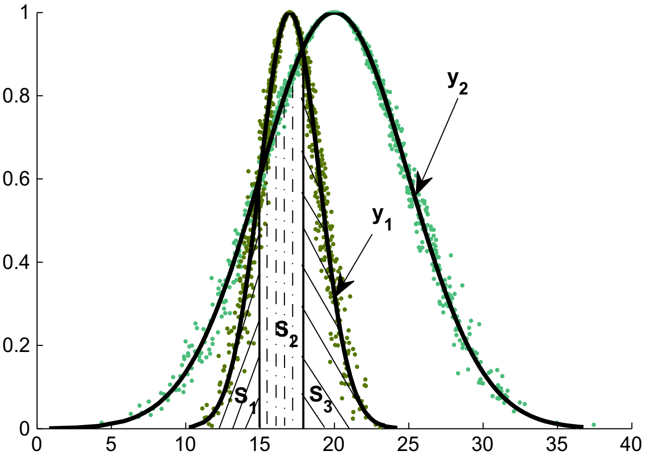


Fig. 7. There are three parts (S_1, S_2 and S_3) constituting the overlapping area when two intersect points locate in $[Ex - 3En, Ex + 3En]$.

Next, we normalize the normal distribution to be the standard normal distribution by $z = \frac{x-Ex}{En}$. We obtain

$$S = \sqrt{2\pi}En_2\int_{-\infty}^{z_2}\Phi_2(z)dz + \sqrt{2\pi}En_1\int_{z_1}^{\infty}\Phi_1(z)dz, \tag{5}$$

where $z_1 = \frac{x_0-Ex_1}{En_1}$ and $z_2 = \frac{x_0-Ex_2}{En_2}$. Then we can efficiently calculate S by looking them up in the table of standard normal distribution.

Since x_0 is the intersect point of the two expectation curves, there is $y_1(x) = y_2(x)$, that is, $e^{-\frac{(x-Ex_1)^2}{2(En_1)^2}} = e^{-\frac{(x-Ex_2)^2}{2(En_2)^2}}$. Solve it and obtain

$$\begin{cases} x_1 = \frac{Ex_2En_1 - Ex_1En_2}{En_1 - En_2}, \\ x_2 = \frac{Ex_1En_2 + Ex_2En_1}{En_1 + En_2}. \end{cases} \tag{6}$$

Two intersect points are obtained, which seems to be contradictory to one intersect point in Fig. 6. In fact, it does have two intersect points. In Fig. 6, we only need to consider one of the two intersect points which is in the interval $[Ex - 3En, Ex + 3En]$, and the other out of $[Ex - 3En, Ex + 3En]$ can be neglected. According to "3En rule", it is sufficient to consider the contribution of the cloud drops in the interval, so we only see one intersect point in Fig. 6. If the two intersect points, x_1 and x_2, all locate in $[Ex - 3En, Ex + 3En]$, we should use the stepwise method to calculate the overlapping area as shown in Fig. 7, that is, $S = S_1 + S_2 + S_3$.

In order to compare each pair of cloud models better, we should normalize the area for the standard similarity measure given by

$$ECM(V_1, V_2) = \frac{2S}{\sqrt{2\pi}En_1 + \sqrt{2\pi}En_2}, \tag{7}$$

where $\sqrt{2\pi}En_1$ and $\sqrt{2\pi}En_2$ denote the area of the two cloud models, respectively.

Thus, given two time series Q and C, after dimensionality reduction by PWCA we can get cloud model vectors Q' and C' of length w. The similarity measure between Q' and C' in the reduced space is

$$D(Q', C') = \sqrt{\frac{1}{w}\sum\nolimits_{i=1}^{w} ECM(q_i', c_i')}. \tag{8}$$

6. Experimental validation of PWCA

In this section, we make some experiments to demonstrate the validation of PWCA. To further illustrate the idea of PWCA for time series, we first give a simple instance to show its prominent advantages. We also use PWCA to perform various time series mining tasks including clustering, classification and similarity search.

The previous work [14,22] have already demonstrated that SAX is superior to the existing methods such as DWT, DFT, SVD, SDA and IMPACTS, which tells us that SAX is a good approximation for time series. Thereby, in our experiments we mainly compare PWCA with SAX using various UCI datasets in the low space. To further demonstrate the performance of PWCA, we also compare it with other existing methods including PLA and ESAX.

6.1. A simple instance

Suppose we have two subsequences Q and C, of which the elements are produced by two functions respectively, i.e., $y_q = sin(t)$ and $y_c = \frac{0.4}{\pi}t - 0.4$, where $t = 0, 0.1, 0.2, \ldots, 2\pi$, as shown in Fig. 8.

If we compare the two subsequences Q and C by PAA (or SAX), we obtain the same result because of their same mean. If we use PWCA to transform the two time series, we will obtain two clouds, $Q' = (0, 0.7957, 0.3556)$ and $C' = (0, 0.2513, 0.0931)$. The characteristics of the two cloud models are different, which indicates the obviously different features of two subsequences Q and C. Their similarity is 1 by PAA (or SAX) and 0.48 by PWCA.

This instance demonstrates that PWCA could recognize the features of the two time series. However, PAA and SAX ignore some cases as mentioned in the above instance.

6.2. Clustering

Hierarchical clustering is a very good way to contrast similarity or distance measure. We cluster 15 time series arbitrarily chosen from the Control Chart dataset [28], which has six groups, that is, normal{1,2}, cyclic{3,4}, increasing trend{5,6,7}, decreasing trend{8,9}, upward shift{10,11,12} and downward shift{13,14,15}. In this experiment, we perform our approach (PWCA) and other three existing methods (SAX, PLA and ESAX) on the 15 time series using hierarchical clustering. To compare the clustering quality and analyze the influence of the reduced dimensionality, we cluster them using every approach three times according to the reduced dimensions $w = [3, 6, 9]$. Moreover, to obtain better results of time series mining for SAX and ESAX, we set a large value of *alphabet_size* (*as*) such as *as* = 9 in this paper.

Figs. 9–11 show the clustering results using the four approaches according to different reduced dimensions. Fig. 9 shows the clustering quality of PWCA in the much lower space $w = 3$ is better than that of the other three existing methods. However, in the spaces $w = 6$ and $w = 10$, The clustering quality of PWCA is the same as that of SAX but better than that of PLA and ESAX. It means that PWCA has a good performance of clustering in much lower

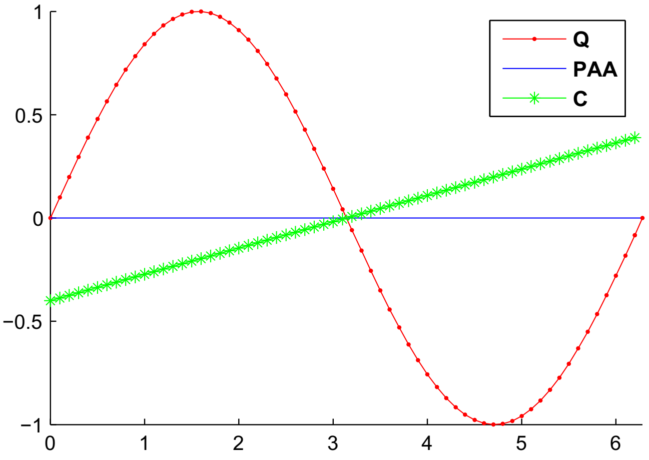


Fig. 8. Two time series have the same mean, but they are apparently different.

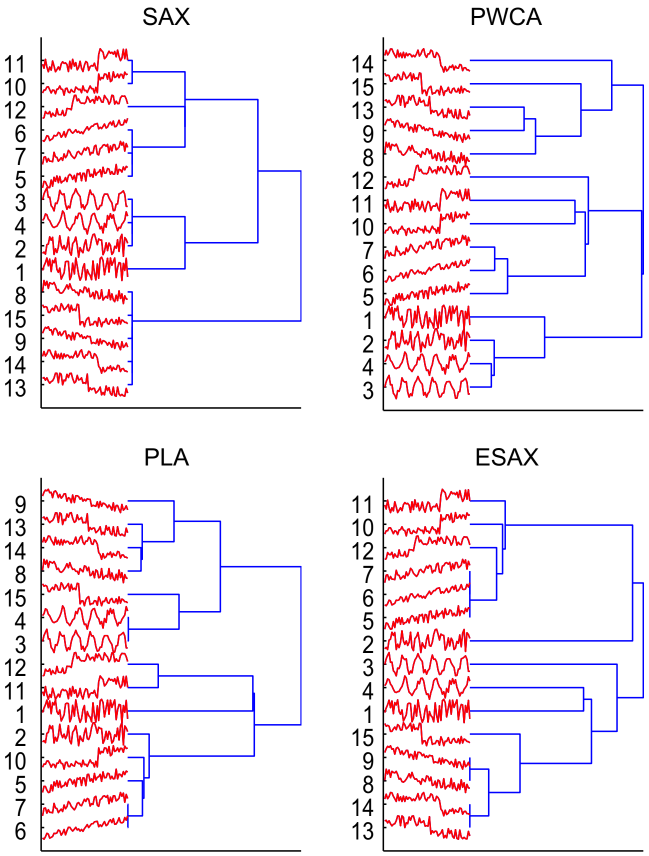


Fig. 9. The clustering results according to the reduced dimension $w = 3$.

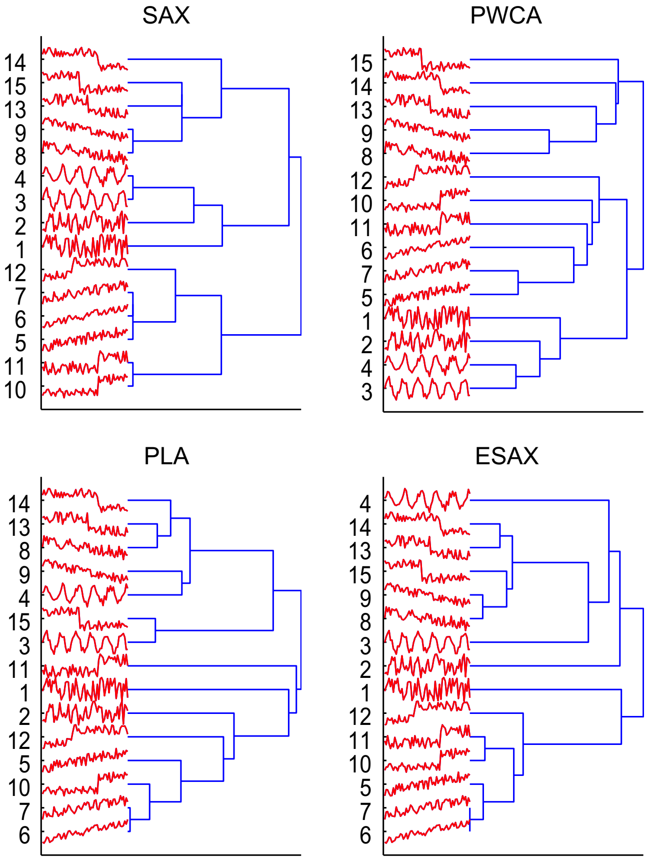


Fig. 10. The clustering results according to the reduced dimension $w = 6$.

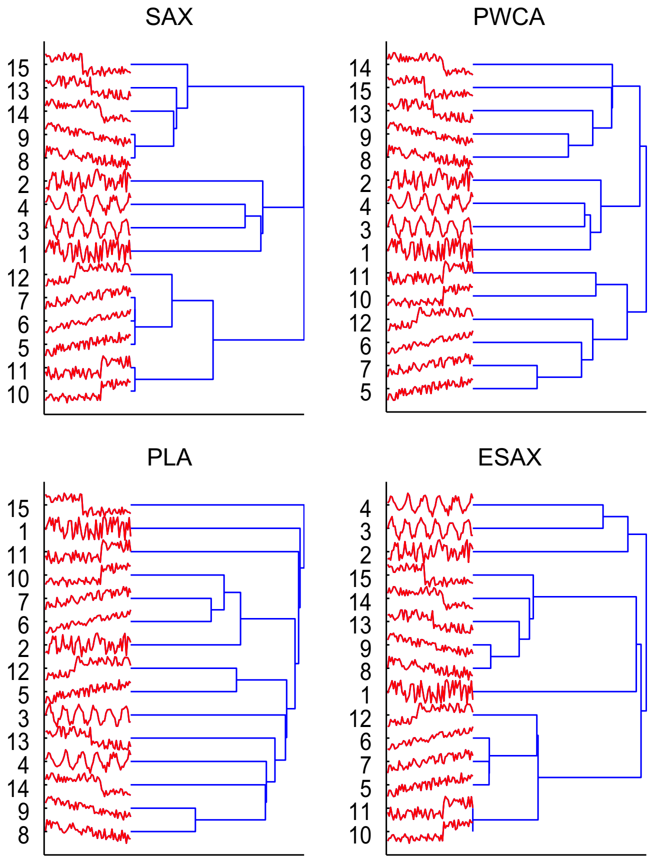


Fig. 11. The clustering results according to the reduced dimension $w = 10$.

space. In other words, PWCA is suitable for clustering under high data compression. In addition, the number of unknown parameters of PWCA, which needs only one as same as PLA, is less than SAX and ESAX.

6.3. Classification

Classification, which is one of the common data mining tasks, has attracted many researchers. We use 1-nearest-neighbor classification method to classify time series in two UCI time series datasets, CC (Control Chart) [28] and CBF (Cylinder-Bell-Funnel) [29,30]. Meanwhile, we compare the classification performance of PWCA with that of other three methods, i.e., SAX, ESAX and PLA.

We divide CC dataset into training group (534) and testing group (66). The test group is composed of the last 11 time series in each class and the remaining of each class compose the training group. We make six experiments according to the reduced dimensions $w = (2,3,6,10,15,20)$. The smaller the w is, the larger compress ratio will be. In every experiment each time series in testing group is used to search the most similar one in training group by the 1-nearest-neighbor classification and statistically count the number of wrong classifications. The classification result is shown in Fig. 12(a), which indicates the error rate of classification. We find that PWCA has better classification performance and is more effective in the larger compression ratio than other three existing methods.

The function of CBF [29,30] is used to produce 150 training time series and 60 testing time series, of which the length is 128. We also perform the 1-nearest-neighbor classification six times according to the reduced dimensions $w = (2,4,8,16,32,64)$. The result of classification is shown in Fig. 12(b), which means that PWCA in much lower spaces ($w = 2,4,8,16$) has better classification performance than the other methods.

From the results of the above experiments, we know that PWCA is also suitable for classification in the much lower space. In other words, comparing to other existing methods, the lower the reduced dimension is, the better the classification performance of PWCA will be.

6.4. Similarity search

To further testify the superior quality of PWCA, we perform similarity search on two time series datasets, CC and TP (Two

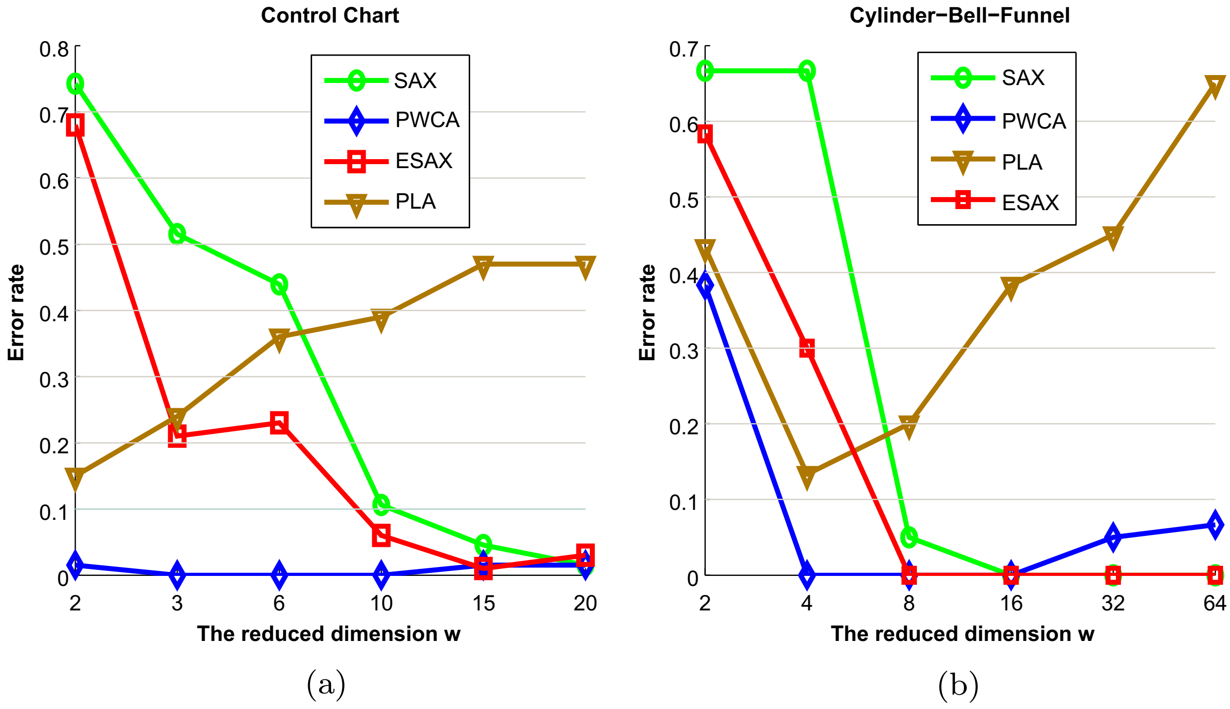


Fig. 12. The classification results in the two databases are obtained by SAX, PWCA, PLA and ESAX according to different reduced dimensions.

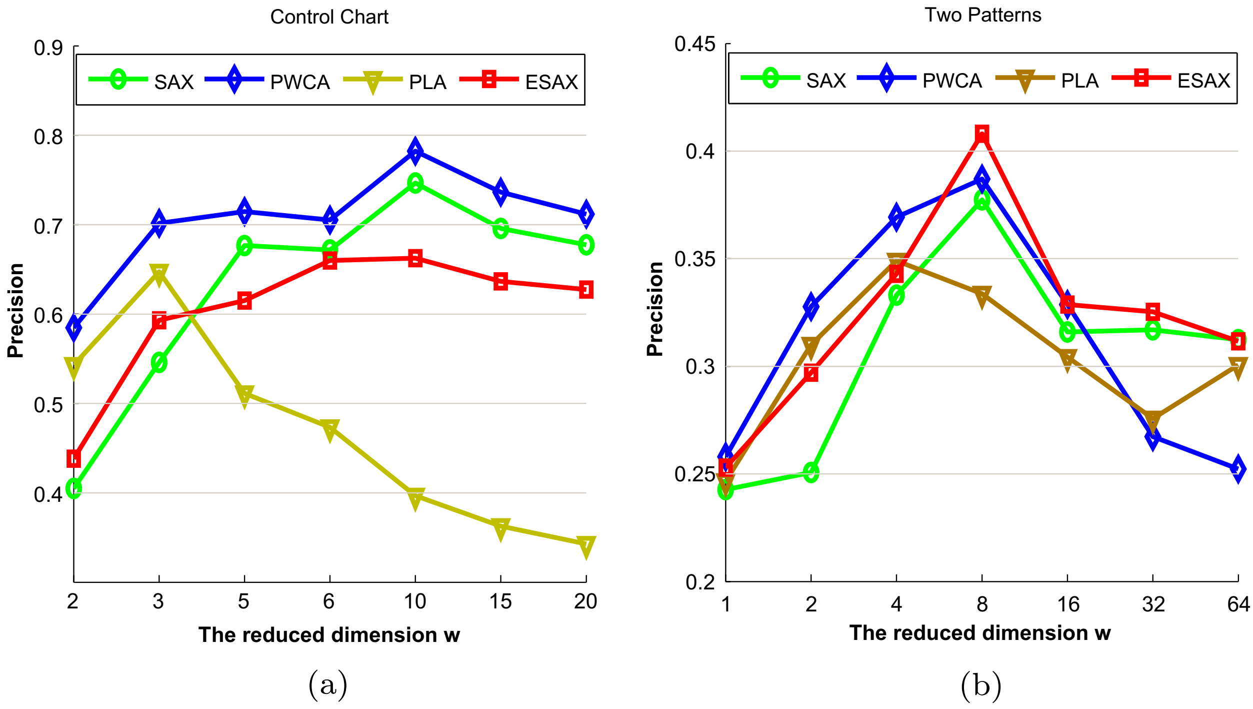


Fig. 13. The results of similarity search are obtained by SAX, PWCA, PLA and ESAX according to different reduced dimensions.

Patterns) [31,32]. We choose n ($n = 66$) time series as query series from the CC dataset and regard the remaining ones as queried time series dataset. We take the similarity search experiments seven times according to the reduced dimensions $w = (2,3,5,6,10,15,20)$. At the same time, we let the query operation of every query time series return the first fs (fs should be smaller than the number of objects, of which the class labels are the same to that of the query object. In this experiment we set $fs = 60$.) most similar time series in the remaining group, and count how many objects in the fs time series of which the class labels are the same to that of the query one. If there are rs right objects of which the class labels are the same to that of the query one, the rate of right search is $rf = \frac{rs}{fs}$. Similarly, we arbitrarily choose n ($n = 30$) time series from TP dataset as query series, search the similar time series in the remaining group according to the different reduced dimensions $w = (1,2,4,8,16,32,64)$ and set $fs = 100$. We use the *precision* to denote the performance of similarity search, i.e.,

$$Precision = \frac{1}{n}\sum_{i=1}^{n} fr_i, \tag{9}$$

where fr_i denotes the rate of right search of the ith query time series used to similarity search.

The results of similarity search in the two datasets by the four approaches according to different reduced dimensions are shown in Fig. 13. In the experiment of CC dataset, PWCA is obvious better than the other methods. In the experiment of TP dataset, the performance of PWCA is worse than that of the other methods when the reduced dimensions w is quite large, but it has a good performance in the much lower dimension such as $w = [1,2,4]$. On the whole, the precision of similarity search using PWAC is higher than that using the other methods in the much lower space, so in this case we can objectively state that PWCA is superior.

7. Conclusions

In this paper we propose a novel technique of time series dimensionality reduction based on cloud model which is called piecewise cloud approximation (PWCA). It not only can validly reduce the dimensionality of time series as same as the other existing methods but also has more prominent performance to mine time series. Moreover, each cloud model can reflect the distribution of data points within the corresponding "frame". It also overcomes some disadvantages of PAA and SAX, such as the simple instance (sub Section 6.1) indicates. Through executing the common tasks of time series mining, the results demonstrate that PWCA is an effective approximation of time series and can obtain better results than other approaches in much lower space.

It is well known that SAX and ESAX are symbolic representation methods and have wide application recently. Therefore, one of the future works is to transform the cloud representations into symbolic strings so that PWCA is more perfect to mine time series. At the same time, we only use two characteristics to measure the similarity between two cloud models. If there is a way to measure the similarity between cloud models using three characters, the similarity measure between time series in reduced space may work better.

Acknowledgments

This work has been partly supported by the Natural Science Foundation of China under Grant Nos. 70871015 and 71031002. We also would like to acknowledge Prof. Eamonn Keogh for the dataset and his procedure source code.

References

[1] R.K. Lai, C.Y. Fan, W.H. Huang, et al., Evolving and clustering fuzzy decision tree for financial time series data forecasting, Expert Systems with Applications 36 (2) (2009) 3761–3773.
[2] J.T. Yao, J.P. Herbert, Financial time-series analysis with rough sets, Applied Soft Computing 9 (3) (2009) 1000–1007.
[3] Q. Song, B.S. Chissom, Fuzzy time series and its models, Fuzzy Sets and Systems 54 (3) (1993) 269–277.
[4] E. Hadavandi, H. Shavandi, A. Ghanbari, Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting, Knowledge-Based Systems 23 (8) (2010) 800–808.
[5] G.P. Zhang, M. Qi, Neural network forecasting for seasonal and trend time series, European Journal of Operational Research 160 (2) (2005) 501–514.

[6] J.B.I. Bulla, Stylized facts of financial time series and hidden semi-Markov models, Computational Statistics and Data Analysis 51 (4) (2006) 2192–2209.

[7] T.C. Fu, A review on time series data mining, Engineering Applications of Artificial Intelligence 24 (1) (2011) 164–181.

[8] Y.S. Lee, L.I. Tong, Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming, Knowledge-Based Systems 24 (1) (2011) 66–72.

[9] R. Agrawal, C. Faloutsos, A. Swami, Efficient similarity search in sequence databases, in: Proceedings of the 4th International Conferences on Foundations of Data Organization and Algorithms, Springer-Verlag, Chicago, 1993, pp. 69–84.

[10] K.P. Chan, A.W.C. Fu, Efficient time series matching by wavelets, in: Proceedings of the 15th IEEE International Conference on Data Engineering, 1999, pp. 117–126.

[11] F. Korn, H.V. Jagadish, C. Faloutsos, Efficiently supporting ad hoc queries in large dataset of time sequences, in: Special Interest Group on Management of Data (SIGMOD'97), 1997, pp. 289–300.

[12] N.Q. Hung, D.T. Anh, An improvement of paa for dimensionality reduction in large time series databases, in: Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence, 2008, pp. 698–707.

[13] E. Keogh, S. Chu, D. Hart, et al., An online algorithm segmenting time series, in: IEEE International Conference on Data Mining, 2001, pp. 289–296.

[14] J. Lin, E. Keogh, S. Lonardi, et al., A symbolic representation of time series, with implications for streaming algorithms, in: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003, pp. 2–11.

[15] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, Fast subsequence matching in time series databases, in: Proceedings of the Special Interest Group on Management of Data (SIGMOD'94), 1994, pp. 419–429.

[16] D.Y. Li, Y. Du, Artificial Intelligence with Uncertainty, Chapman & Hall/CRC, 2008.

[17] E.V. Bauman, A.A. Dorofeyuk, G.V. Kornilovm, Optimal piecewise-linear approximation algorithms for complex dependencies, Automation and Remote Control 65 (2004) 1667–1674.

[18] H. Zhang, S.N. Wan, Linearly constrained global optimization via piecewise-linear approximation, Journal of Computational and Applied Mathematics 214 (2008) 111–120.

[19] B. Lkhagva, Y. Suzuki, K. Kawagoe, Extended SAX: extension of symbolic aggregate approximation for financial time series data representation, in: DEWS2006 4A-i8, pp. 1–6.

[20] H. Hndre-Jonsson, D.Z. Badal, Using signature files for querying time-series data, in: Proceedings of Principles of Data Mining and Knowledge Discovery, 1997, pp. 211–220.

[21] Y.W. Huang, P.S. Yu, Adaptive query processing for time-series data, in: Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining, 1999, pp. 282–286.

[22] E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrotra, Dimensionality reduction for fast similarity search in large time series databases, Knowledge and Information Systems 3 (3) (2001) 263–286.

[23] D.Y. Li, Knowledge representation in KDD based on linguistic atoms, Journal of Computer Science and Technology 12 (6) (1997) 481–496.

[24] C. Çiflikli, E.K. Özyirmidokuz, Implementing a data mining solution for enhancing carpet manufacturing productivity, Knowledge-Based Systems 23 (8) (2010) 783–788.

[25] D.Y. Li, J.W. Han, X.M. Shi, et al., Knowledge representation and discovery based on linguistic atoms, Knowledge-Based systems 10 (7) (1998) 431–440.

[26] D.Y. Li, Uncertainty reasoning based on cloud models in controllers, Computers and Mathematics with Applications 35 (3) (1998) 99–123.

[27] B.X. Liu, H.L. Li, L.B. Yang, Cloud decision analysis method, Control and Decision 24 (6) (2009) 957–960.

[28] D.T. Pham, A.B. Chan, Control chart pattern recognition using a new type of self organizing neural network, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 212 (1) (1998) 115–127.

[29] N. Saito, Local feature extraction and its application using a library of bases, Ph.D. Thesis, Yale University, 1994.

[30] S. Manganaris, Supervised classification with temporal data, Ph.D. Thesis, Computer Science Department, School of Engineering, Vanderbilt University, 1997.

[31] P. Geurts, Contributions to decision tree induction: bias/variance tradeoff and time series classification, Ph.D. Thesis, Department of Electrical Engineering, University of Liege, Belgium, 2002.

[32] E. Keogh, X.P. Xi, L. Wei, et al., UCR time series classification & clustering page, 2003. Available from: <http://www.cs.ucr.edu/eamonn/time_series_data/>.