Information Sciences 374 (2016) 32–50

Contents lists available at ScienceDirect

Information Sciences

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

# Piecewise two-dimensional normal cloud representation for time-series data mining

Weihui Deng, Guoyin Wang*, Ji Xu

*Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China*

ARTICLE INFO

*Article history:*
Received 5 December 2015
Revised 6 September 2016
Accepted 10 September 2016
Available online 14 September 2016

*Keywords:*
Two-dimensional normal cloud
Representation
Dimensionality reduction
Similarity measure
Cloud model
Time-series data mining

ABSTRACT

Many high-level dimensionality reduction approaches for mining time series have been proposed, e.g., SAX, PWCA , and Feature-based. Due to the rapid performance degradation of time-series data mining in much lower dimensionality and the continuously increasing amount of time series data with uncertainty, there remains a burning need to develop new time-series representations that can retain good performance in much lower reduced space and address uncertainty efficiently. In this work, we propose a novel time series representation, namely Two-dimensional Normal Cloud Representation (2D-NCR), based on cloud model theory. The representation achieves dimensionality reduction by transforming the raw time series into a sequence of two-dimensional normal cloud models. Moreover, a new similarity measure between the transformed time series is presented. The proposed method can reflect the characteristic data distribution of the time series and capture the variation with time. We validate the performance of our representation on the various data mining tasks of classification, clustering, and query by content. The experimental results demonstrate that 2D-NCR is an effective and competitive representation for time-series data mining.



## 1. Introduction

A time series is a collection of numerical values obtained from sequential measurements over time. It is generally used to describe the current status and future variations of objects over time lags. Enormous amounts of time-series data are being continually generated and collected from various domains, including the financial industry, the ecological environment, aviation, telecommunication, the medical field, and so on.

Time-series data mining (TSDM), as a research issue currently receiving significant amounts of attention, has been extensively studied to discover the knowledge and information hiding in time-series data. Various tasks are applied to mine time series, which can be summarized by 7 aspects [13]: classification [1,6,18], clustering [15,28], query of content [23,27,37], prediction [10,21,42], segmentation [16,43], anomaly detection [38] and motif discovery [3]. These tasks have been successfully applied to various fields, including power systems, smart grid, stock market time-series prediction and biomedicine. With the rapid growth of digital sources of information, all of the tasks of TSDM are unveiling various facets of complexity. The most prominent problems stem from the high dimensionality of time-series data and the difficulty of defining a form

* Corresponding author.
*E-mail addresses:* dengweihui@cigit.ac.cn (W. Deng), wanggy@ieee.org (G. Wang).

http://dx.doi.org/10.1016/j.ins.2016.09.027
0020-0255/

of similarity measure based on human cognition. Therefore, the following two major issues must be addressed to perform TSDM tasks efficiently [13]:

- Time-series representation. The motivation of time-series representation is to gain additional benefits such as acceleration of processing, accuracy improvement and noise removal, with a simultaneous emphasis on the essential and concise characteristics of the raw time series. Thus, a representation technique should have the properties of significant dimensionality reduction, emphasis on fundamental data distribution and variation, good reconstruction quality from the reduced space, and insensitivity to noise.
- Similarity measure. How to distinguish or match any pair of time series and how to formalize an intuitive similarity or distance between two time series based on human cognition are the two core issues of similarity measure. Therefore, a similarity measure should have the following properties: recognition of perceptually similar time series, consistency with human intuition, emphasis on the most salient features on both local and global scales, and capability to identify or distinguish arbitrary objects without any assumptions.

In this work, we primarily focus on the problem of time-series representation and the corresponding similarity measure. The formalized problem of time-series representation can be described as follows:

Given a time series $T = (t_1, t_2, \ldots, t_n)$ of length $n$, a time-series representation of $T$ is a model $\overline{T}$ of reduced dimensionality $w (w \ll n)$ such that $\overline{T}$ closely approximates $T$ or extracts the essential features of $T$.

Over the past several decades, numerous time-series representations have been developed and successfully applied to various time-series data mining tasks. The well-known representations include the transformation domain methods (like discrete Fourier transform (DFT) [14], discrete wavelet transform (DWT) [4]), piecewise aggregate approximation (PAA) [22], piecewise linear approximation (PLA) [32], singular value decomposition (SVD) [30], symbolic aggregate approximation (SAX) [26], symbolic aggregate approximation based on trend distance (SAX-TD) [33], derivative segment approximation (DSA) [19], piecewise cloud approximation (PWCA) [25], learned pattern similarity (LPS) [2], and non-parametric symbolic approximate representation (NSAR) [20]. All of the methods mentioned above have in common the capability of representing time series. Additionally, particular time-series representations have been proposed for the specific TSDM tasks [8,17,31]. For instance, feature-based representations have been developed for time-series classification [17].

However, most of the existing dimension-reduction representations mentioned above have a major limitation; their performance degrades rapidly in much lower dimensionality, although they are the better approximate representations for mining time-series data in the appropriate reduced space. That is, a larger compression ratio means worse performance. PWCA was proposed to overcome this limitation by representing raw time series as a series of cloud models. However, because it only emphasizes the distribution of raw time series without considering variation with time, its performance still must improve. In addition, the prominent state-of-the-art methods for time-series dimension-reduction representation cannot address the vagueness and uncertainty inherent in certain time-series data because of inaccuracies in measurements, incomplete sets of observations, or difficulties in obtaining the measurements.

The motivation of this paper is to develop a novel time-series representation and the corresponding similarity measure in a reduced space to tackle the three problems mentioned above. First, the new representation must not only be able to approximate time series well for the TSDM tasks but also guarantee its performance in much lower dimensionality. In other words, the new method must guarantee its efficiency and accuracy in both high and much lower reduced space. Second, the proposed approach should consider the distribution and variation of the time series. Finally, because increasing numbers of time-series datasets with uncertainty are being employed, the new approach must be capable of handling uncertainty so that it can be successfully applied to various TSDM tasks.

In this paper, we propose a novel time-series representation to reduce the dimensionality of time series based on cloud model theory. The proposed method first calculates the first-order difference of the raw time series to obtain the variation series over time. Second, we simultaneously partition the raw series and variation series into equal-length "subsequence pairs". Each "subsequence pair" is then transformed into one two-dimensional normal cloud model. Thus, a two-dimensional normal cloud model sequence for a time series is obtained. Finally, we propose a new similarity measure between two two-dimensional normal cloud models and the corresponding similarity measure for the transformed time series.

To compare with prominent state-of-the-art methods for time-series representation and dimensionality reduction, our approach has the following advantages:

- Using six numerical characteristics of a two-dimensional normal cloud model to represent a subsequence in the reduced space can retain more characteristics of the raw time series than state-of-the-art methods such as PAA, SAX, or transformation domain methods (DFT, DWT), which determines that our approach will have better performance in much lower reduced space.
- The proposed approach considers the distribution of raw time series and the variation with time rather than only the distribution of the time series (PWCA); thus, the proposed approach outperform PWCA. In addition, the similarity measure between the two cloud models in this paper is more simple and efficient than that of is PWCA.
- The normal cloud model can effectively integrate the randomness and fuzziness of concepts via its numerical characteristics, which will be further discussed in Section 2.2. This effectiveness guarantees that the proposed method can efficiently address the uncertainty inherent in time-series data. In addition, 2D-NCR is insensitive to abnormal data or noise because the cloud model emphasizes the whole distribution rather than a single point.

To validate the performance of the proposed method, we performed our method on 20 time-series datasets (including 4 synthetic datasets, 9 real world datasets, and 7 shape datasets) for three time-series data mining tasks (classification, clustering, and query of content). The experimental results demonstrate that 2D-NCR is an effective and competitive representation method for time-series data mining.

The remainder of this paper is organized as follows. Section 2 briefly discusses state-of-the-art methods for time-series representation and reviews some basic concepts of the cloud model. In Section 3, we present our piecewise two-dimensional normal cloud representation and the corresponding similarity measure. Section 4 validates the performance of our method by implementing three time-series data mining tasks. The last section summarizes the conclusions.

## 2. Related work

### 2.1. Time-series representation

In the past two decades, numerous high-level time-series representations have been proposed for time-series data mining. According to the taxonomy of [26], these representations can be divided into two categories, namely *Non-Data Adaptive* and *Data Adaptive*.

The *Non-Data Adaptive* methods used parameters of the transformation to represent the raw time series regardless of its nature. One of the popular transformation techniques is DFT, which was first proposed for efficient similarity search in sequence databases by Agrawal et al. [14]. The basic idea of DFT contains two phases: using DFT to map time sequences to the frequency domain, and selecting $w$ Fourier coefficients to represent raw time series of length $n$, where $w \ll n$. Unlike DFT, Chan and Fu used scaled and shifted versions of a mother wavelet function instead of a fixed set of basis functions, namely DWT [4]. The main difference between DFT and DWT is that DFT always captures the global features of the time series, whereas some wavelet coefficients represent the local subsection time series. Another well-known *Non-Data Adaptive* method is PAA introduced by Keogh et al. [22], which transforms a time series of length $n$ into $w$ segments ($w \ll n$) represented by the mean values of data points falling within the segment. In [11], D'Urso proposed to consider the geometric representations of time arrays in the "object space", and analyzed three different dissimilarity measures between multivariate time-series trajectories. Thereafter, this representation is applied into implement fuzzy clustering [12] and classification [7] for multivariate time series.

In the *Data Adaptive* representations, the parameters of a transformation are modified depending upon the data available. The most widely used time-series representation for time-series data mining is SAX, proposed by Lin and Keogh [26]. It first transforms a raw time series into the PAA representation and then symbolizes the PAA representation into a discrete string with equiprobability. The distance measure between two symbolic strings in SAX representation space has a lower bound; thus, the SAX representation accelerates the time-series data mining process while maintaining the quality of the mining results. However, the SAX representation only emphasizes the average values of segments without considering their variation with time. Consequently, different segments with similar average values can be transformed into the same symbols with the SAX distance equal to 0. To overcome this limitation, many researchers have performed various studies. Sun et al. designed a measure to compute the distance of trends using the starting and the ending points of segments and proposed a modified distance measure by integrating the SAX distance with a weighted trend distance, namely SAX-TD [33]

To address the vagueness and uncertainty inherent in certain time-series data, Li and Guo used cloud model to represent time series and proposed a novel approach called piecewise cloud approximation [25]. They first partitioned a time series into equal length "frames" for approximation and then transformed each "frame" into a cloud model. After obtaining a set of cloud sequences, they proposed a new similarity measure between two cloud sequences based on expectation curves of cloud models. The PWCA representation can achieve a better approximation to mine time series in the much lower reduced space than can other techniques such as SAX, DWT. Nevertheless, because it only emphasizes the distribution of data points within the corresponding "frame" without considering variation with time, its performance still must improve. In addition, the similarity measure of PWCA has high time complexity.

To support accurate and fast similarity detection, Gullo et al. presented a derivative time-series segment approximation (DSA) representation model, which originally featured derivative estimation and segment approximation to provide both high sensitivity in capturing the variation and data compression [19]. DSA yielded a concise yet feature-rich time-series representation and achieved effective and efficient similarity detection of time series by using DTW as the similarity measure.

Other *Data Adaptive* representations have been proposed. In PLA [32], a time series is divided into several segments represented by linear functions. It is often used for the time series segmentation task. Singular value decomposition, proposed by Korn et al. [30], considered all of the time series in the dataset instead of one time series at a time. Moreover, almost all *Non-Data Adaptive* approaches can be extended into *Data Adaptive* methods by adding an extra data-adaptive selection step; e.g., Keogh et al. proposed an extended version of PAA, namely adaptive piecewise constant approximation, in which the length of each segment is adaptive to the shape of the series rather than fixed.

### 2.2. Cloud model

The cloud model, as a new cognition model of uncertainty, was proposed by Li [24] based on probability theory and fuzzy set theory. It is a significant approach to realizing bidirectional cognitive transformation between qualitative concepts

and a quantitative description. In cloud model theory, it is possible to measure the deviation of a random phenomenon from a normal distribution when the random phenomenon does not strictly satisfy a normal distribution [35]. Additionally, a cloud model can formally describe the inherent relationship between randomness and fuzziness.

Normal cloud models based on a normal distribution function and a Gaussian membership function are the most important cloud models. They have been successfully applied with universality to many fields [9,29,40]. The definition of a normal cloud model is as follows.

**Definition 1** ([24]). : Let $U$ be a universal set described by precise numbers and $C(Ex, En, He)$ be a qualitative concept related to $U$. If a quantitative number $x(x \in U)$ is a random realization of the concept $C$, and $x$ is subjected to the normal distribution $x \sim N(Ex, En'^2)$, where $En'$ is a random realization of the normal distribution $En' \sim N(En, He^2)$, and the certainty degree of $x$ on $U$ is $\mu(x) = \exp\{-\frac{(x-Ex)^2}{2(En')^2}\}$, then the distribution of $x$ on $U$ is a normal cloud, and each $x$ is defined as a cloud drop.

The normal cloud has the following properties:

- The universal set $U$ can be either one-dimensional or multi-dimensional, and the one-dimensional normal cloud model has the "expectation curve" $y = \exp\{-\frac{(x-Ex)^2}{2(En)^2}\}$.
- For any $x \in U$, the mapping from $x$ to [0, 1] is multiple. The certainty degree of $x$ on $C$ is a normal distribution rather than a fixed number.
- The cloud is composed of cloud drops. The more cloud drops there are, the better the overall features of this concept are represented.
- The more probable cloud drop $x$ appears, the higher the certainty degree is and hence the greater is the contribution cloud drop $x$ makes to the concept.

These features of normal clouds mean that it is reasonable to use the two-dimensional normal cloud to represent time series. The normal cloud can effectively integrate the randomness and fuzziness of concepts via the three following numerical parameters:

- *Ex* (Expectation). The mathematical expectation of the cloud drops distributed in the universal set. In other words, *Ex* is the point that is most representative of the qualitative concept.
- *En* (Entropy). The uncertainty measurement of the qualitative concept. *En* is determined by both the randomness and the fuzziness of the concept.
- *He* (Hyper-entropy). *He* is the uncertainty measurement of the entropy.

The two-dimensional normal cloud model is an extension of the normal cloud model, which can be defined as follows:

**Definition 2** ([24]). : Let $U\{X, Y\}$ be a two-dimensional universal set described by precise numbers and $C(Ex, Ey, Enx, Eny, Hex, Hey)$ be a qualitative concept related to $U$. If a quantitative number $(x, y)$, $(x \in X, y \in Y)$ is a random realization of the concept $C$, and $(x, y)$ is subjected to the two-dimensional normal distribution $(x, y) \sim N((Ex, Ey), (Enx'^2, Eny'^2))$, where $(Enx'^2, Eny'^2)$ is a random realization of two-dimensional normal distribution $(Enx'^2, Eny'^2) \sim N((Enx, Eny), (Hex^2, Hey^2))$, and the certainty degree of $(x, y)$ on $U$ is

$$\mu(x, y) = \exp\left\{-\frac{(x-Ex)^2}{2(Enx')^2} - \frac{(y-Ey)^2}{2(Eny')^2}\right\} \tag{1}$$

then the distribution of $(x, y)$ on $U$ is a two-dimensional normal cloud, and each $(x, y)$ is defined as a cloud drop.

Two cloud transformations, namely forward cloud transformation (*FCT*) and backward cloud transformation (*BCT*), are used to realize the bidirectional cognitive transformation between the intension and extension of a concept [24]. *FCT* is used to perform the transformation from intension to extension of a concept, whereas *BCT* implements the transformation from extension to intension. The two-dimensional forward normal cloud transformation algorithm (*T-FNCT*) and the two-dimensional backward normal cloud transformation algorithm (*T-BNCT*) are described as Algorithms 1 and 2 [41].

## 3. 2D-NCR: two-dimensional normal cloud representation

In this section, we first propose a new time-series representation, namely Two-dimensional Normal Cloud Representation (2D-NCR), and then, we present the corresponding similarity measure for 2D-NCR in the reduced space.

### 3.1. Two-dimensional normal cloud representation

Let $T = (t_1, t_2, \ldots, t_n)$ be the raw time-series and $w$ be the reduced dimension; then, the Two-dimensional Normal Cloud Representation can be divided into three stages, as shown in Fig. 1.

*Stage 1*: Calculate the first-order difference of the raw time series $T$ to obtain the variation sequence $\Delta T = (e_1, e_2, \ldots, e_n)$, where

$$e_i = \begin{cases} 0, & \text{if} \quad i = 1 \\ t_i - t_{i-1}, & \text{othewise} \end{cases} \tag{2}$$

**Algorithm 1.** T-FNCT.

**Input:** Numerical characteristics $(Ex, Ey, Enx, Eny, Hex, Hey)$, number of cloud drop $n$
**Output:** Cloud drops $(x_i, y_i)$ with certainty degree $\mu(x_i, y_i)(i = 1, 2, \ldots, n)$
*Step 1*: Generate a two-dimensional normal distributed random vector $\left(Enx_i', Eny_i'\right)$ with expectation $(Enx, Eny)$ and variance $\left(Hex^2, Hey^2\right)$.
*Step 2*: Generate a two-dimensional normally distributed random vector $(x_i, y_i)$ with expectation $(Ex, Ey)$ and variance $\left(Enx_i'^2, Eny_i'^2\right)$.
*Step 3*: Calculate the certainty degree

$$\mu(x_i, y_i) = \exp\left\{-\frac{(x_i - Ex)^2}{2(Enx'_i)^2} - \frac{(y_i - Ey)^2}{2(Eny'_i)^2}\right\}$$

*Step 4*: $(x_i, y_i)$ is a two-dimensional cloud drop with certainty degree $\mu(x_i, y_i)$.
*Step 5*: Repeat *Steps 1* to *4* until $n$ two-dimensional cloud drops are generated.

**Algorithm 2.** T-BNCT.

**Input:** Samples $(x_i, y_i)(i = 1, 2, \ldots, n)$.
**Output:** Numerical characteristics $(Ex, Ey, Enx, Eny, Hex, Hey)$.
*Step 1*: Calculate the mean and variance for each dimension of the sample, i.e.,

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i, Sx^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \bar{X}\right)$$

$$\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} y_i, Sy^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(y_i - \bar{Y}\right)$$

*Step 2*: Calculate expectation $Ex$ and $Ey$:

$$Ex = \bar{X}, Ey = \bar{Y}$$

*Step 3*: Calculate entropy $Enx$ and $Eny$:

$$Enx = \sqrt{\frac{\pi}{2}} \times \frac{1}{n}\sum_{i=1}^{n}|x_i - Ex|, Eny = \sqrt{\frac{\pi}{2}} \times \frac{1}{n}\sum_{i=1}^{n}|y_i - Ey|$$

*Step 4*: Calculate hyper-entropy $Hex$ and $Hey$:

$$Hex = \sqrt{Sx^2 - Enx^2}, Hey = \sqrt{Sy^2 - Eny^2}$$

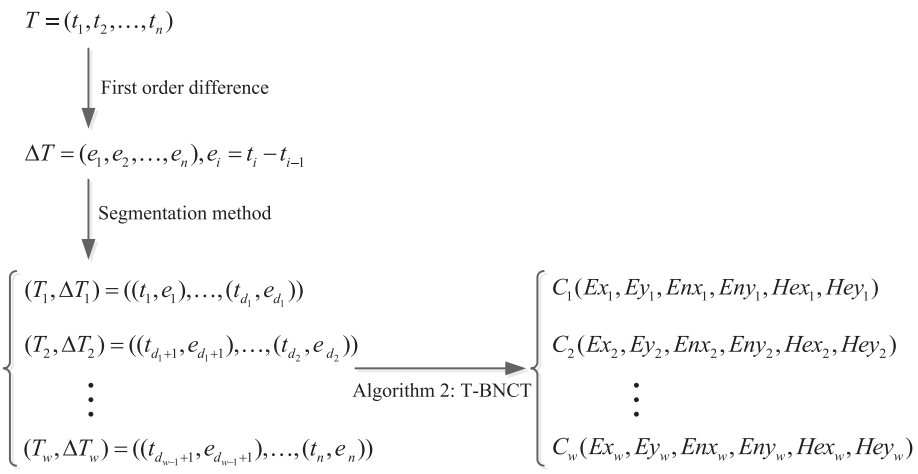


**Fig. 1.** The framework of 2D-NCR.

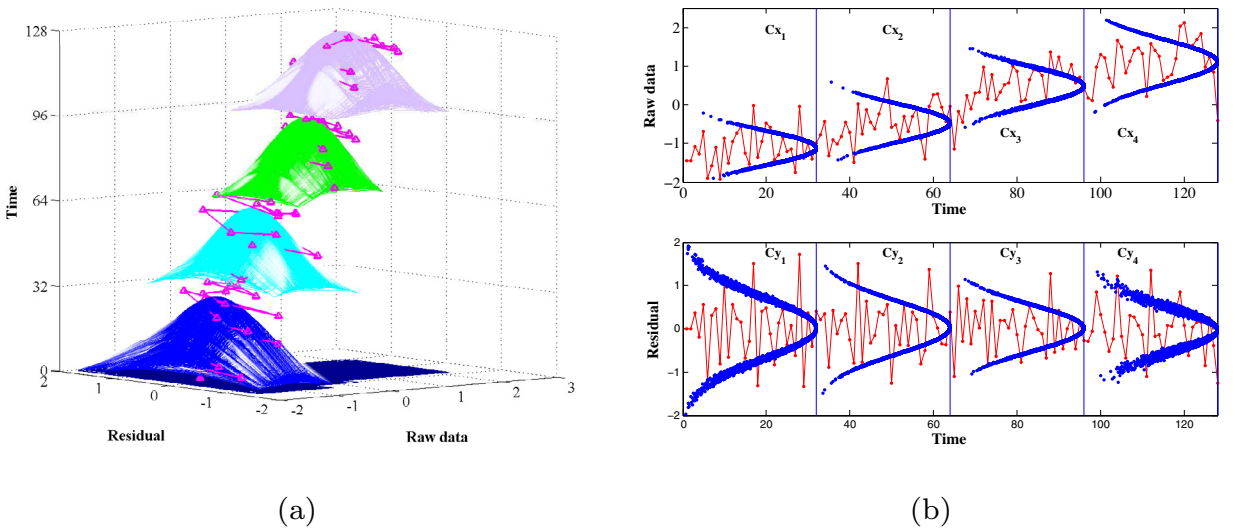


**Fig. 2.** (a) 2D-NCR of a time series with length 128. (b) The normal cloud representation of the two dimensions: "Raw data-time" and "Residual-time", respectively.

*Stage 2*: Simultaneously segment $T$ and $\Delta T$ into $w$ segments using segmentation methods, and then group them into a sequence of "subsequence pairs" $(T_i, \Delta T_i), i = 1, \cdots, w$. In this paper, we use the equal-length method to segment the series into $w$ equal-length segments. That is, the length of the "subsequence pair" $(T_i, \Delta T_i), i = 1, \ldots, w$ is $\lceil n/w \rceil$. Thus, $d_i$ in Fig. 1 represents the following

$$d_i = \frac{n}{w} \times i, i = 1, \ldots, w - 1 \tag{3}$$

*Stage 3*: Perform the two-dimensional backward normal cloud transformation algorithm (*T-BNCT*) to transform each "subsequence pair" into a two-dimensional normal cloud. Finally, we obtain a two-dimensional normal cloud sequence $(C_1, C_2, \ldots, C_w)$ for a time series.

The data distribution and variation of time series are the two most common and important characteristics for a time-series dataset. However, the existing representations either employ one of them to reduce the time series or partially make use of the two characteristics to perform the dimension reduction. For example, PWCA only uses the data distribution of time series to reduce the dimensionality, whereas SAX-TD utilizes the starting point, the average point, and the ending point of the segment to capture the trends and measure the distance. In this paper, 2D-NCR employs the two-dimensional normal cloud model; one dimension can capture the data distribution of a raw time series, and the other dimension is the representative of variation with time, to represent the time series in the reduced space. Thus, 2D-NCR can retain more characteristics of the raw time series than can competing dimension-reduction methods, e.g., PWCA, SAX, and PAA. Therefore, the performance of 2D-NCR for time series data mining is better than state-of-the-art methods in much lower reduced space.

Furthermore, 2D-NCR transforms the time series into a sequence of two-dimensional normal cloud models, and then uses the numeric characteristics of the cloud models to represent the time series. In contrast, we can use these cloud models to reconstruct the raw time series with some randomness and fuzziness. One numeric characteristic of cloud model *En*, determined by the randomness and fuzziness of the concept, is the uncertainty measurement of the qualitative concept. Thus, 2D-NCR can address the vagueness and uncertainty inherent in certain time-series datasets effectively. Moreover, 2D-NCR is insensitive to abnormal data because the cloud model emphasizes the whole distribution rather than a single point. In other words, 2D-NCR satisfies the two properties of "good reconstruction quality from the reduced space" and "insensitivity to noise" described in Section 1. All of these characteristics of 2D-NCR determine that 2D-NCR will be an effective time-series representation for time-series data mining.

Fig. 2 illustrates the two-dimensional normal cloud sequence representation for a raw time series from the CBF (Cylinder-Bell- Funnel) dataset, which will be described further in Section 4.1. 2D-NCR transforms the raw time series of length 128 into 4 two-dimensional normal clouds. The new representation significantly reduces the dimensionality while emphasising the characteristics of data distribution and variation of the raw time series. Fig. 2a shows the 2D-NCR in the three-dimensional coordinate system, consisting of raw data, first-order difference sequence (namely residual), and time stamp. Fig. 2b shows the Normal Cloud Representation of the two dimensions: "Raw data-time" and "Residual-time". Fig. 2b shows that the whole tendency of the raw time series is *upward*, because the entropy *En* of $Cx_{i+1}$ is greater than $Cx_i$. However, the local variation in each segment remains almost unchanged because the entropy *En* of $Cy_i (i = 1, \ldots, 4)$ is nearly equal to 0. Furthermore, the expectation *Ex* of the normal cloud reflects the data distribution of each segment, whereas the entropy *En*

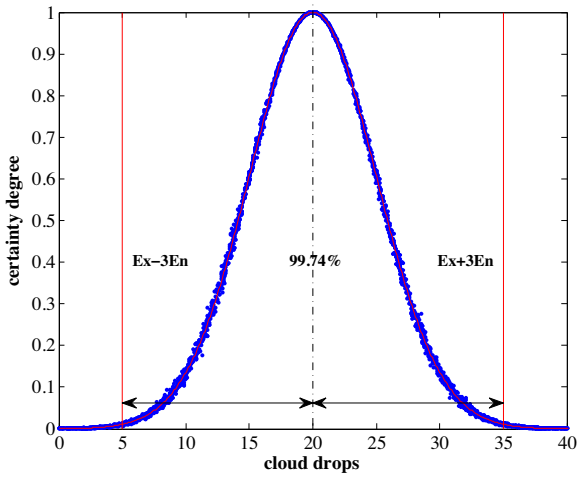


**Fig. 3.** Contributions to the concept $C(20, 5, 0.3)$ by cloud drops [24].

and the hyper-entropy *He* of the normal cloud can emphasize the dispersion degree. Overall, 2D-NCR can not only capture the feature of data distribution but also reflect the global and local variation of the raw time series.

### 3.2. Similarity measure

Having introduced the 2D-NCR of a time series, we now present a new similarity measure on it in the reduced space. Many approaches of distance measure or similarity measure for time series have been proposed and successfully applied in various domains. Wang et al. [39] presented a detailed survey and conducted an extensive experimental study re-implementing the similarity measures and their variants and testing their effectiveness on 38 time-series datasets from a wide variety of application domains. Nevertheless, none of them is applicable to the new representation in the reduced space, because the numerical characteristics of cloud model have their specific meaning, e.g., the expectation *Ex* is the point that is most representative of the qualitative concept.

Our new approach to similarity measures is based on the "3*En* rule", which is similar to the "3$\sigma$ rule" of a normal distribution. Thus, the contributive cloud drops on a concept $C$ in the universal domain $U$ lie in the interval $[Ex - 3En, Ex + 3En]$ with probability 99.74% [24]. In other words, we can neglect the contribution to concept $C$ by the cloud drops out of the interval $[Ex - 3En, Ex + 3En]$. E.g., for the concept $C(20, 5, 0.3)$, we can neglect the cloud drops out of the interval [5, 35], as shown in Fig. 3.

Fig. 4 is the framework of the new similarity measure, which clearly shows that the similarity between two time series in the reduced space can be calculated by the "***divide-calculate-combine***" strategy. The idea of the new approach derives from the perspective of multi-granular computing (MGrC), which emphasizes jointly utilizing multiple levels of information granules in problem solving. The approach first considers dividing the complex problem at a coarse level into simple sub-problems at finer levels and then integrates the solution components at each layer to form the correct solution to the entire problem [36]. In addition, having been represented by 2D-NCR, the raw time series is transformed into a two-dimensional normal cloud model sequence in which each two-dimensional normal cloud model is independent of the others. Therefore, it is reasonable to divide the primitive task into smaller independent subtasks and then combine the solution in the reduced space to form the final solution of the primitive problem.

Given two time series, $T = (t_1, t_2, \ldots, t_n)$ and $\bar{T} = (\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_n)$, after dimensionality reduction by 2D-NCR, $T$ and $\bar{T}$ can be represented as a two-dimensional normal cloud sequence $(C_1, C_2, \ldots, C_w)$ and $(\bar{C}_1, \bar{C}_2, \ldots, \bar{C}_w)$, respectively. Then, we can describe the "***divide-calculate-combine***" strategy as follows.

***Divide***. Fig. 4 shows that the new approach first divides the task of calculating the similarity between two raw time series into $w$ smaller subtasks that calculate the similarity between two two-dimensional normal cloud models in the reduced space. For example, the $i^{th}$ subtask in the reduced space is to calculate the similarity between $C_i(Ex_i, Ey_i, Enx_i, Eny_i, Hex_i, Hey_i)$ and $\bar{C}_i(\bar{E}x_i, \bar{E}y_i, \bar{E}nx_i, \bar{E}ny_i, \bar{H}ex_i, \bar{H}ey_i)$. Then, each subtask calculating the similarity between two two-dimensional normal cloud models is further divided into two much smaller subtasks by projecting the two-dimensional normal cloud model into the plane "raw data-time" and the plane "residual-time" (For convenience, we use XOZ and YOZ to replace "raw data-time" and "residual-time", respectively). Thus, the new subtask calculates the similarity between two one-dimensional normal cloud models.

***Calculate***. After the "***divide***" stage, the primitive problem is transformed to calculate the similarity between two one-dimensional normal cloud models. In this stage, we present a new similarity measure between two cloud models based on the overlap degree and intersection's certainty degree of their expectation curves in the interval $[Ex - 3En, Ex + 3En]$.

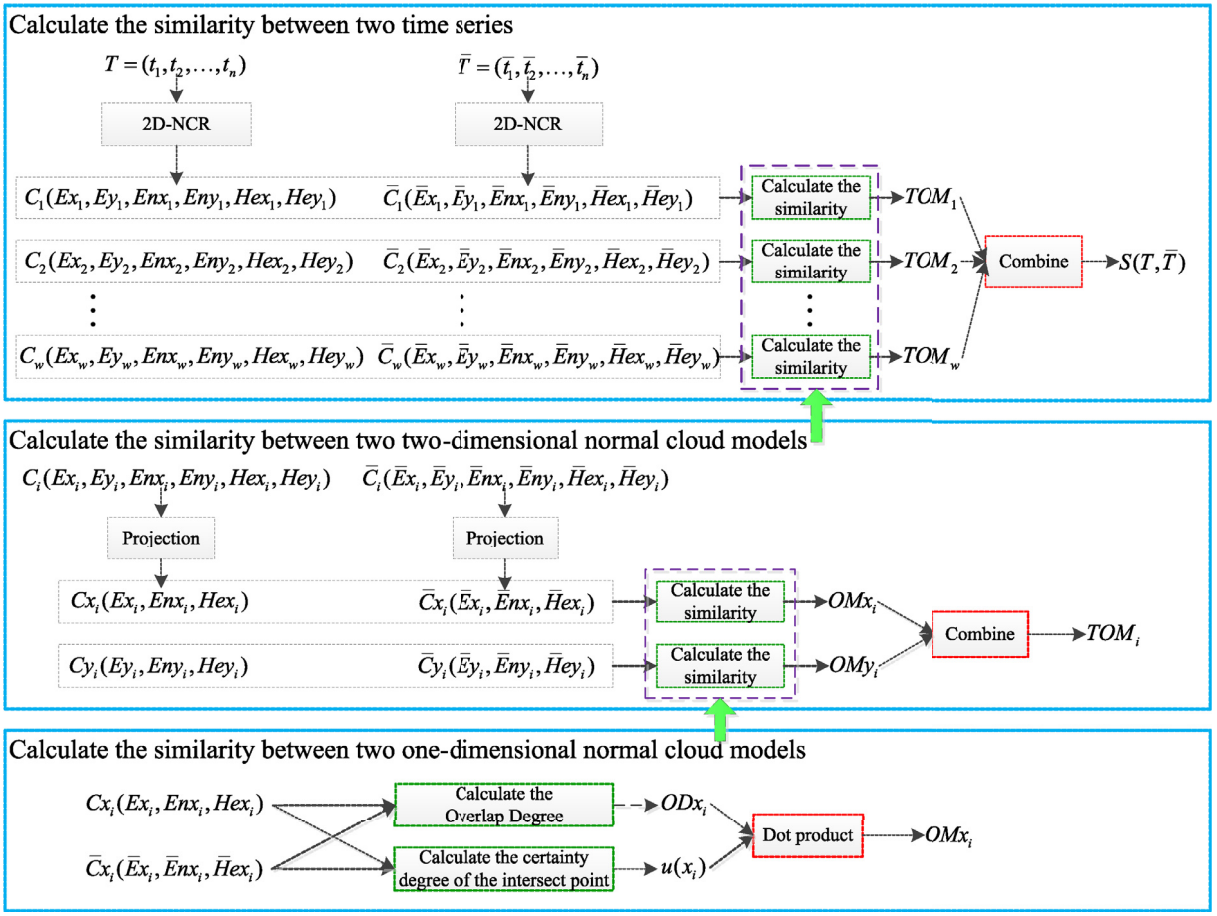


**Fig. 4.** The framework of the proposed similarity measure.

Given two cloud models, $Cx_i(Ex_i, Enx_i, Hex_i)$ and $\bar{C}x_i(\bar{E}x_i, \bar{E}nx_i, \bar{H}ex_i)$, the expectation curves can be expressed as follows:

$$y=\exp\left\{-\frac{(x-Ex_i)^2}{2(Enx_i)^2}\right\},\quad \bar{y}=\exp\left\{-\frac{\left(x-\bar{E}x_i\right)^2}{2\left(\bar{E}nx_i\right)^2}\right\} \tag{4}$$

Then, the approach can be divided into the following three steps.

***Step 1***: Calculate the overlap degree ($ODx_i$) of the expectation curves $y$ and $\bar{y}$. The overlap degree is the measure of two cloud models' overlaps in the horizontal direction. For convenience, we let $Ex_i < \bar{E}x_i$, implying that $Cx_i$ lies to the left of $\bar{C}x_i$ in the horizontal direction, as shown in Fig. 5. Then, the overlap degree $ODx_i$ can be calculated as the following expression.

$$ODx_i=\frac{2\times\left(\min\{Sup\{Cx_i\},Sup\{\bar{C}x_i\}\}-\max\{Inf\{Cx_i\},Inf\{\bar{C}x_i\}\}\right)}{(Sup\{Cx_i\}-Inf\{Cx_i\})+\left(Sup\{\bar{C}x_i\}-Inf\{\bar{C}x_i\}\right)} \tag{5}$$

where $Sup\{Cx_i\}$ and $Inf\{Cx_i\}$ are the upper bound and lower bound of cloud drops belonging to cloud model $Cx_i$ and satisfying the "3*En* rule". That is, $Sup\{Cx_i\}=Ex_i+3Enx_i$ and $Inf\{Cx_i\}=Ex_i-3Enx_i$, respectively. $Sup\{\bar{C}x_i\}$ and $Inf\{\bar{C}x_i\}$ are the upper bound and lower bound of cloud drops belonging to cloud model $\bar{C}x_i$ and satisfying the "3*En* rule". That is, $Sup\{\bar{C}x_i\}=\bar{E}x_i+3\bar{E}nx_i$ and $Inf\{\bar{C}x_i\}=\bar{E}x_i-3\bar{E}nx_i$, respectively. Note that if there is no overlapping area satisfying the "3*En* rule" between $Cx_i$ and $\bar{C}x_i$, then the overlap degree $ODx_i=0$, and if the two cloud models are the same, then the overlap degree $ODx_i=1$.

***Step 2***: Calculate the certainty degree ($u(x_i)$) of the two expectation curves' intersections. Assume that $x_0$ is the intersection of two expectation curves, and $y(x)=\bar{y}(x)$ . Solve and obtain

$$x_1=\frac{\bar{E}x_i\times Enx_i-Ex_i\times\bar{E}nx_i}{Enx_i-\bar{E}nx_i}\quad or\quad x_2=\frac{Ex_i\times\bar{E}nx_i+\bar{E}x_i\times Enx_i}{Enx_i+\bar{E}nx_i} \tag{6}$$

There thus are three cases:

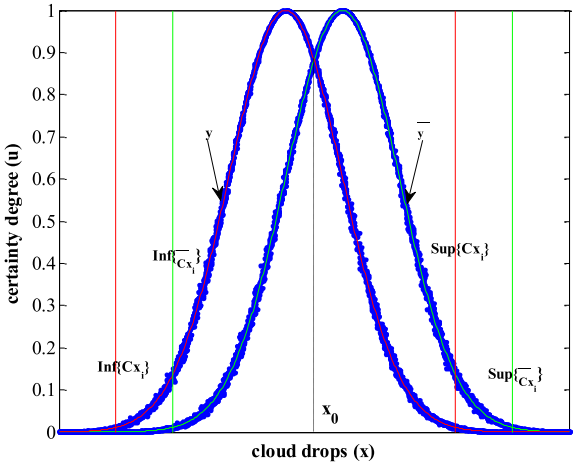


**Fig. 5.** The upper and lower bound of cloud drops satisfying the "3*En* rule".

*Case 1*: if $x_1$ and $x_2$ all lie outside the interval $[Ex-3En, Ex+3En]$, then $u(x_i)=0$;
*Case 2*: if $x_1$ or $x_2$ lies within the interval $[Ex-3En, Ex+3En]$ (Assume the point is $x_1$), then $u(x_i)=u(x_1)$.
*Case 3*: if $x_1$ and $x_2$ all lie within the interval $[Ex-3En, Ex+3En]$, then $u(x_i)=\max\{u(x_1), u(x_2)\}$.

***Step 3***: Combine $ODx_i$ and $u(x_i)$ to obtain the similarity ($OMx_i$) between cloud models $Cx_i(Ex_i, Enx_i, Hex_i)$ and $\bar{C}x_i(\bar{E}x_i, \bar{E}nx_i, \bar{H}ex_i)$. From Step 1 and Step 2, we can see that the overlap degree $ODx_i$ is the similarity measure of the two cloud models in the horizontal direction, and the certainty degree $u(x_i)$ is the similarity measure of the two cloud models in the vertical direction. Therefore, the similarity $OMx_i$ between $Cx_i$ and $\bar{C}x_i$ can be calculated by the following equation.

$$OMx_i = ODx_i \times u(x_i) \tag{7}$$

Notes that both the similarity measures in the horizontal direction and vertical direction play a significant role in calculating the similarity $OMx_i$. E.g., given three cloud models $Cx_i(20, 5, 0.3)$, $\bar{C}x_{i1}(25, 5, 0.3)$, and $\bar{C}x_{i2}(20, 25/7, 0.3)$, we determine that the overlap degree $ODx_{i1}$ between $Cx_i$ and $\bar{C}x_{i1}$ is equal to the overlap degree $ODx_{i2}$ between $Cx_i$ and $\bar{C}x_{i2}$ by Eq. (5), in which $ODx_{i1}=ODx_{i2}=0.8333$. We can calculate the similarities in the vertical direction, where $u(x_{i1})=0.8825$ and $u(x_{i2})=1$. Thus, the similarity between $Cx_i$ and $\bar{C}x_{i1}$ is $OMx_{i1}=0.8333\times0.8825=0.7354$, whereas the similarity between $Cx_i$ and $\bar{C}x_{i2}$ is $OMx_{i2}=0.8333\times1=0.8333$. Thus, the certainty degree $u(x_i)$ affects the similarity $OMx_i$ independently of the overlap degree $ODx_i$. Similarly, we can demonstrate that the overlap degree also has an independent effect on calculating the similarity $OMx_i$. Therefore, it is reasonable to calculate the similarity between two cloud models by combining the similarities in the horizontal and vertical directions.

***Combine***. As described in "***Divide***" stage, the primitive task of calculating the similarity between two raw time series has been divided twice. Thus, to obtain the final result, we should also perform the "combine operation" twice.

First, we combine the similarity measures of two one-dimensional cloud models on plane XOZ and plane YOZ to obtain the similarity measure $TOM_i$ of two two-dimensional cloud models, shown as follows.

$$TOM_i = OMx_i \times OMy_i \tag{8}$$

where $OMx_i$ is the similarity measure of two cloud models on plane XOZ, and $OMy_i$ is the similarity measure of two cloud models on plane YOZ.

Then, the similarity $S(T,\bar{T})$ of the two raw time series can be calculated by combining the $w$ similarity measures corresponding to the "two-dimensional cloud model pairs" obtained in the "***Divide***" stage. The similarity $S(T,\bar{T})$ of time series $T=(t_1,t_2,\ldots,t_n)$ and $\bar{T}=(\bar{t}_1,\bar{t}_2,\ldots,\bar{t}_n)$ in the reduced space is

$$S(T,\bar{T}) = \sqrt{\frac{1}{w}\sum_{i=1}^{w} TOM_i} \tag{9}$$

Fig. 5 and Eq. (5) show that the overlap degree represents how similar the two clouds' "3*En* intervals" are, and the certainty degree of the two expectation curves' intersections (Eq. (6)) reflects the similarity of the two clouds' shape. Thus, the composite multiplicative operation (Eq. (7)) considers the similarity of the two clouds' location and shape. Moreover, Eq. (8) shows that the proposed method simultaneously employs the two dimensions (raw time series and residual time series) to calculate the similarity between the two time series, in which the raw time series considers data distribution and the residual time series reflects the variation with time. These characteristics determine that the proposed method has good ability to identify the dissimilarity between the two time series. For example, let $T_1$ and $T_2$ be two time-series segments. Assume that the data within $T_1$ and $T_2$ has identical distribution with expectation $Ex$, and assume that the variation with

time within $T_1$ is not the same as that within $T_2$. Thus, if we use PAA, SAX, or PWCA to represent two time series, then the similarity between $T_1$ and $T_2$ will be equal to 1. By using the proposed method, the similarity between $T_1$ and $T_2$ will not be equal to 1 because the variation with time within the segments will help to identify the dissimilarity between the two time series. Furthermore, Fig. 4 shows that the proposed similarity measure divides the primitive complicated task into simple subtasks, and then calculates the similarity between two cloud models by combining the similarities in the horizontal and vertical directions. Both of the two steps are consistent with the process of human cognition. In other words, the proposed similarity measure is consistent with human intuition. Finally, there is no assumed restriction on time series implying that the new similarity measure is universal in the sense that it allows identifying arbitrary time series.

### 3.3. Computational complexity analysis

In this subsection, the computational complexity of the proposed approach is analyzed. First, we consider the time complexity of two-dimensional normal cloud representation. For stage 1, obtaining the difference sequence $\Delta T$ of raw time series $T$ with length $n$ consumes $O(n)$ time. According to Algorithm 2, the time complexity of transforming one "subsequence pair" $(T_i, \Delta T_i)$ of length $\lceil n/w \rceil$ into a two-dimensional normal cloud is $O(3 \cdot \lceil n/w \rceil)$. Thus, the overall complexity of two-dimensional normal cloud representation is

$$O(n) + w \cdot O(3 \cdot \lceil n/w \rceil) \approx O(n) \tag{10}$$

Then, we analyze the time complexity of the proposed similarity measure using a bottom-up approach. At the bottom in Fig. 4, the time complexity of calculating the similarity between two one-dimensional normal cloud models is $O(1) + O(1) = 2 \cdot O(1)$. Consequently, the time complexity of calculating the similarity between two two-dimensional normal cloud models is $2 \cdot O(1) + 2 \cdot O(1) = 4 \cdot O(1)$. Then, calculating the similarity of two two-dimensional normal cloud sequences has complexity $w \cdot 4 \cdot O(1)$. Therefore, the overall complexity of the proposed similarity measure is

$$O(n) + 4w \cdot O(1) \approx O(n) + O(w) \tag{11}$$

Representing time series $T$ by a sequence of two-dimensional normal cloud model with length $w$ requires $6w$ memory-space units. From Fig. 4, 6 units are required to obtain similarity $TOM_i$ between the two two-dimensional normal cloud models $C_i$ and $\bar{C}_i$. Thus, the total space needed to calculate the similarity between two time series is $6w + 6w = 12w$. That is, the space complexity of the proposed similarity measure is $O(w)$.

## 4. Experimental validation

In this section, we perform three of the most common data mining tasks on 20 time-series datasets using 2D-NCR and compare the results with state-of-the-art methods. Subsection 4.1 briefly introduces the datasets used in this paper. We validate the performance of the proposed method in terms of classification, clustering, and query of content in Subsections 4.2, 4.3, and 4.4, respectively.

### 4.1. Datasets

The 20 time-series datasets analyzed in this study are collected from *The UCR Time Series Classification/Clustering Homepage* [5], which is derived from a broad range of applications including yoga poses (Yoga), monitoring of fish migration (Fish), the shapes of Swedish leaves (Swedish Leaf), and so on. A summary of the datasets is shown in Table 1. In this paper, we validate the classification accuracy of our proposed method using all 20 datasets, as with other methods such as SAX [26], SAX-TD [33], and Feature-based [17]. To compare the clustering performance with PWCA [25] and DSA [19], four datasets including Synthetic Control, Gun-Point, CBF and Trace are used in the clustering task. In addition, we select the Synthetic Control and Two Patterns datasets to perform the query of content and compare its performance with PWCA [25].

### 4.2. Classification

Classification of time series is one of the common time-series data mining tasks. Many classifiers have been proposed for special purposes. In this paper, we choose the one nearest neighbor (1-NN) classifier on labeled data to validate the classification accuracy of the proposed approach. One main advantage is that the time-series representation and the corresponding underlying similarity metric are critical to the performance of the 1-NN classifier; thus, the accuracy of the 1-NN classifier directly reflects the effectiveness of the time-series representation and the similarity measure. The other main advantage is that the 1-NN classifier is straightforward to implement and is parameter free, which allows other researchers to reproduce our experiment results easily. Furthermore, most of the existing time-series dimensionality reduction approaches and similarity measures adopted the 1-NN classifier to evaluate their performance, facilitating comparing our approach with state-of-the-art methods objectively. The error rate is used to measure the classification accuracy. That is, the smaller the error rate is, the better the performance will be.

First, we evaluate the classification performance of the 2D-NCR on different reduced dimensions $w$ and compare it with that of existing common time-series representations such as PWCA, PAA, PLA, and SAX. The experiments are performed

**Table 1**
Fundamental characteristics of the datasets used in this paper.

| No. | Name | Classes | Size of training set | Size of testing set | Length of time series | Type |
|---|---|---|---|---|---|---|
| 1 | Synthetic Control | 6 | 300 | 300 | 60 | Synthetic |
| 2 | Gun-Point | 2 | 50 | 150 | 150 | Real |
| 3 | CBF | 3 | 30 | 900 | 128 | Synthetic |
| 4 | Face (all) | 14 | 560 | 1690 | 131 | Shape |
| 5 | OSU Leaf | 6 | 200 | 242 | 427 | Shape |
| 6 | Swedish Leaf | 15 | 500 | 625 | 128 | Shape |
| 7 | 50Words | 50 | 450 | 455 | 270 | Real |
| 8 | Trace | 4 | 100 | 100 | 275 | Synthetic |
| 9 | Two Patterns | 4 | 1000 | 4000 | 128 | Synthetic |
| 10 | Wafer | 2 | 1000 | 6174 | 152 | Real |
| 11 | Face (four) | 4 | 24 | 88 | 350 | Shape |
| 12 | Lightning-2 | 2 | 60 | 61 | 637 | Real |
| 13 | Lightning-7 | 7 | 70 | 73 | 319 | Real |
| 14 | ECG | 2 | 100 | 100 | 96 | Real |
| 15 | Adiac | 37 | 390 | 391 | 176 | Shape |
| 16 | Yoga | 2 | 300 | 3000 | 426 | Shape |
| 17 | Fish | 7 | 175 | 175 | 463 | Shape |
| 18 | Beef | 5 | 30 | 30 | 470 | Real |
| 19 | Coffee | 2 | 28 | 28 | 286 | Real |
| 20 | OliveOil | 4 | 30 | 30 | 570 | Real |

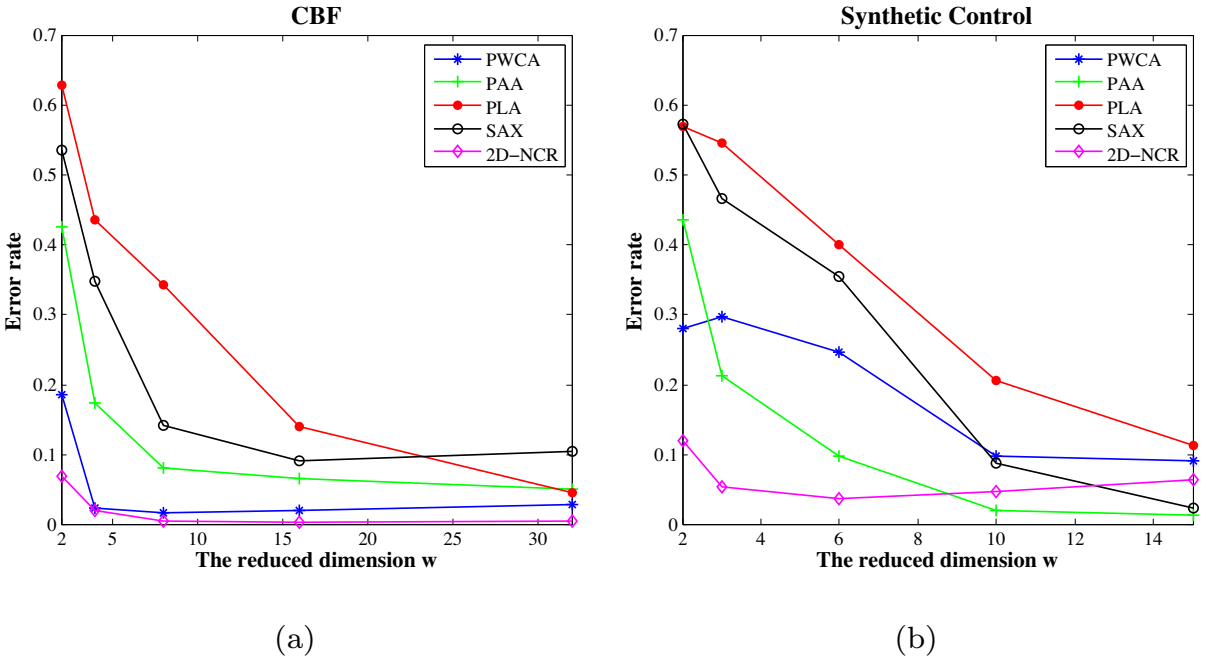


**Fig. 6.** Classification results of different approaches in the two datasets according to linear spline interpolation. (a) CBF; (b) Synthetic Control.

on the CBF and Synthetic Control datasets. Both datasets are used exactly as obtained from *The UCR Time Series Classification/Clustering Homepage* [5], without any preprocessing and using the specified method to partition each dataset into training and testing portions. For the CBF dataset, we set the reduced dimension $w = (2, 4, 8, 16, 32)$ successively for five experiments and analyzed the relationship using the linear spline interpolation. The smaller reduced dimension greater compression ratio. Fig. 6 (a) shows the interpolation results, which indicate that 2D-NCR obtained higher classification accuracy than did PWCA, PAA, PLA, or SAX both in the high space of dimensionality reduction and the much lower reduced space on the CBF dataset. Fig. 6 (a) also illustrates that the performances of PAA, PLA, and SAX degraded rapidly with the decrease of the reduced dimension, whereas 2D-NCR and PWCA can maintain their performance in much lower reduced space. Because 2D-NCR considers the variation with time of the raw time series whereas PWCA not, 2D-NCR obtains better performance than does PWCA in the lower reduced space, which can be further demonstrated in the experiment with the Synthetic Control dataset.

**Table 2**
Error rates of different approaches on the 20 datasets.

| No. | Name | 2D-NCR | SAX | ED | SAX-TD | Feature-based |
|---|---|---|---|---|---|---|
| 1 | Synthetic Control | 0.0433 | **0.02** | 0.12 | 0.77 | 0.037 |
| 2 | Gun-Point | **0.0467** | 0.18 | 0.087 | 0.073 | 0.073 |
| 3 | CBF | **0.0022** | 0.104 | 0.148 | 0.088 | 0.289 |
| 4 | Face (all) | **0.2142** | 0.33 | 0.286 | 0.215 | 0.292 |
| 5 | OSU Leaf | 0.3719 | 0.467 | 0.483 | 0.446 | **0.165** |
| 6 | Swedish Leaf | **0.0912** | 0.483 | 0.211 | 0.213 | 0.227 |
| 7 | 50Words | **0.2506** | 0.341 | 0.369 | 0.338 | 0.453 |
| 8 | Trace | **0** | 0.46 | 0.24 | 0.21 | 0.01 |
| 9 | Two Patterns | 0.1447 | 0.081 | 0.093 | **0.071** | 0.074 |
| 10 | Wafer | 0.0126 | 0.0034 | 0.0045 | 0.0042 | **0** |
| 11 | Face (four) | **0.0227** | 0.17 | 0.216 | 0.181 | 0.261 |
| 12 | Lightning-2 | 0.229 | 0.213 | 0.246 | 0.229 | **0.197** |
| 13 | Lightning-7 | 0.3424 | 0.397 | 0.425 | **0.329** | 0.438 |
| 14 | ECG | 0.11 | 0.12 | 0.12 | 0.09 | **0.01** |
| 15 | Adiac | **0.2583** | 0.89 | 0.389 | 0.273 | 0.355 |
| 16 | Yoga | **0.138** | 0.195 | 0.17 | 0.179 | 0.226 |
| 17 | Fish | **0.12** | 0.474 | 0.217 | 0.154 | 0.171 |
| 18 | Beef | 0.433 | 0.567 | 0.467 | **0.2** | 0.433 |
| 19 | Coffee | 0.1428 | 0.464 | 0.25 | **0** | **0** |
| 20 | OliveOil | 0.2667 | 0.833 | 0.133 | 0.067 | **0.01** |

**Table 3**
Rankings of different approaches for classification on the 20 datasets.

| No. | Name | 2D-NCR | SAX | ED | SAX-TD | Feature-based |
|---|---|---|---|---|---|---|
| 1 | Synthetic Control | 3 | **1** | 4 | 5 | 2 |
| 2 | Gun-Point | **1** | 5 | 4 | 2.5 | 2.5 |
| 3 | CBF | **1** | 3 | 4 | 2 | 5 |
| 4 | Face (all) | **1** | 5 | 3 | 2 | 4 |
| 5 | OSU Leaf | 2 | 4 | 5 | 3 | **1** |
| 6 | Swedish Leaf | **1** | 5 | 2 | 3 | 4 |
| 7 | 50Words | **1** | 3 | 4 | 2 | 5 |
| 8 | Trace | **1** | 5 | 4 | 3 | 2 |
| 9 | Two Patterns | 5 | 3 | 4 | **1** | 2 |
| 10 | Wafer | 5 | 2 | 4 | 3 | **1** |
| 11 | Face (four) | **1** | 2 | 4 | 3 | 5 |
| 12 | Lightning-2 | 3.5 | 2 | 5 | 3.5 | **1** |
| 13 | Lightning-7 | 2 | 3 | 4 | **1** | 5 |
| 14 | ECG | 3 | 4.5 | 4.5 | 2 | **1** |
| 15 | Adiac | **1** | 5 | 4 | 2 | 3 |
| 16 | Yoga | **1** | 4 | 2 | 3 | 5 |
| 17 | Fish | **1** | 5 | 4 | 2 | 3 |
| 18 | Beef | 2.5 | 5 | 4 | **1** | 2.5 |
| 19 | Coffee | 3 | 5 | 4 | **1** | **1** |
| 20 | OliveOil | 4 | 5 | 3 | 5 | **1** |
| Average ranking | | **2.15** | 3.825 | 3.825 | 2.37 | 2.875 |
| Standard deviation | | 1.39 | 1.33 | **0.78** | 0.97 | 1.58 |

Note: If some approaches have the same error rate, then we let their rankings be equal to the average of their rankings. E.g., for the Gun-Point dataset, the ascending order of error rates is "2D-NCR"<"SAX-TD" = "Feature-based"<"ED"<"SAX", then their rankings are 1, 2.5, 2.5, 4, and 5, respectively, because (2+3)/2 = 2.5.

For the Synthetic Control dataset, we also conduct the 1-NN classifier five times by setting the reduced dimension $w = (2, 3, 6, 10, 15)$, and then compare the results based on linear spline interpolation. The experimental results are shown in Fig. 6 (b). The figure clearly shows that 2D-NCR significantly outperforms the others methods when the reduced dimension $w \leq 9$ and that the error rate of 2D-NCR is greater than are those of PAA ($w \geq 10$) and SAX ($w \geq 12$) but remains less than are those of other representations. These results validate that 2D-NCR has better performance for classification in much lower reduced space than have state-of-the-art methods.

Then, to examine more closely how 2D-NCR is a highly effective time-series representation for classification of time series, we ran an extensive experiment on 20 time-series datasets, as mentioned in Section 4.1, and compared the error rate with some existing results achieved by SAX [26], Euclidean Distance [5], SAX-TD [33], and Feature-based [17]. All of the datasets are used exactly as obtained from *The UCR Time Series Classification/Clustering Homepage* [5], without any pre-processing and using the specified method to partition each dataset into training and testing portions. The experimental results are presented in Table 2. Table 3 shows the rankings of different approaches for classification on the 20 time-series

**Table 4**
Result of Friedman test for the ranking in Table 3.

| | Sum of Squares | Degree of Freedom | Mean Square | $\chi^2$ | $p$-value |
|---|---|---|---|---|---|
| Columns | 50.1 | 4 | 12.525 | 20.29 | 0.0004 |
| Error | 147.4 | 76 | 1.9395 | | |
| Total | 197.5 | 99 | | | |

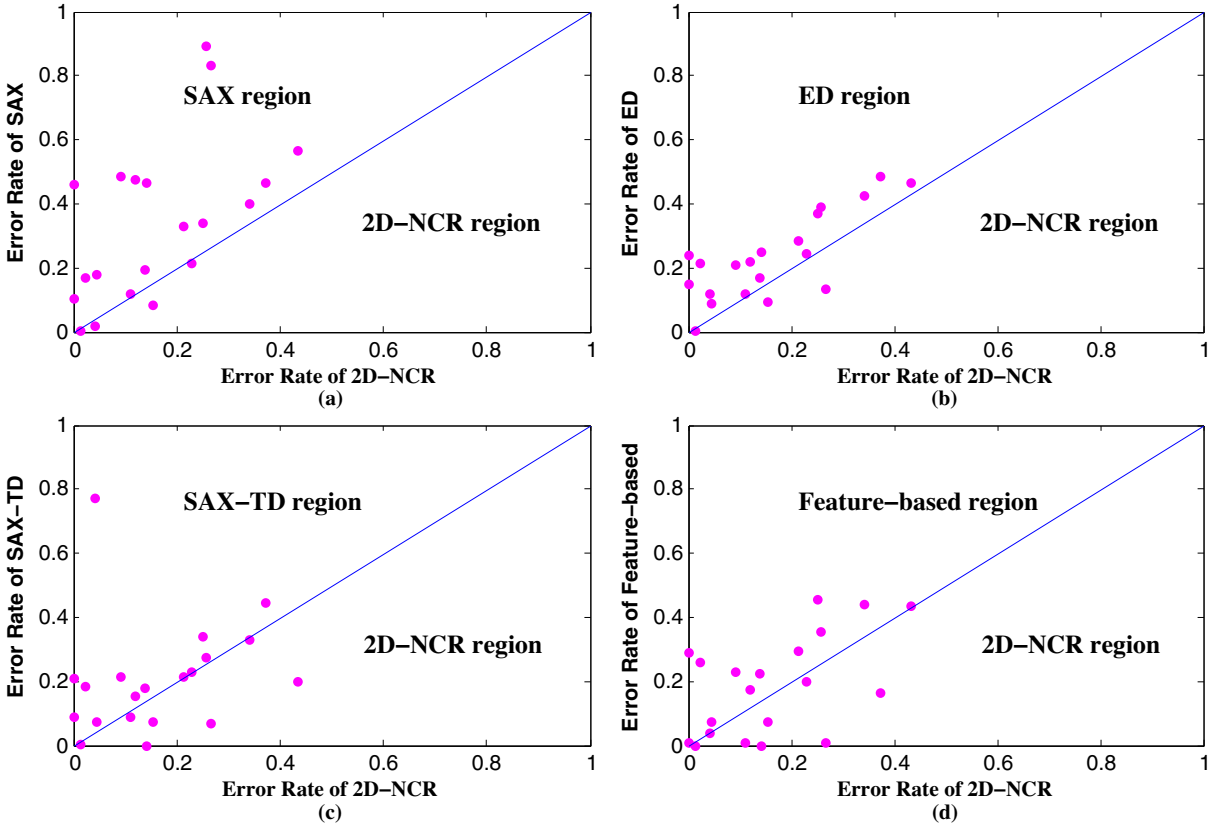


**Fig. 7.** A pairwise comparison of classification error rates for SAX, Euclidean distance (ED), SAX-TD, Feature-based, and 2D-NCR. (a) SAX vs. 2D-NCR. (b) ED vs. 2D-NCR. (c) SAX-TD vs. 2D-NCR. (d) Feature-based vs. 2D-NCR.

datasets. From Tables 2 and 3, we can see that 2D-NCR ranked first 10 times, whereas the other methods altogether ranked first 10 times in total. Furthermore, we employ the Friedman test to verify the statistical significance of the results. Table 4 is the Friedman test's ANOVA table, which shows that the $p$-value is 0.0004, far smaller than the significance level of 0.05. This measure implies a statistically significant difference among these classification results. Moreover, the average ranking of 2D-NCR is 2.15, less than that of any other method. Therefore, overall, 2D-NCR achieves higher classification accuracy on the 20 datasets than the other four methods. Note that the standard deviation of 2D-NCR is slightly greater than that of ED and SAX-TD. This difference is caused by the datasets Two Patterns and Wafer, in which all methods perform well and the classification accuracy of 2D-NCR is slightly less than that of the others.

To provide a more intuitive illustration of the performance of the approaches compared in Table 2, we use scatter plots to conduct pair-wise comparisons. On a scatter plot, the error rates of the two methods under comparison are used as the $x$ and $y$ coordinates of a dot, where a dot represents a particular dataset. A dot fallings in a region means that the corresponding method of the region has a worse performance than that of the other method. Moreover, the further a dot is from the line, the greater the margin of accuracy improvement. Fewer dots in a region indicates the better method.

The scatter plots of the comparison results are shown in Fig. 7. Fig. 7(a)-(b) show clearly that 2D-NCR performs significantly better than SAX and Euclidean Distance do because almost all of the dots fall into the "SAX region" and "ED region", respectively. Fig. 7(c) shows that the number of dots that fall in the "SAX-TD region" versus the "2D-NCR region" is 10 versus 4, and 6 dots are on the line or almost on the line. In other words, 2D-NCR outperform SAX-TD for time-series classification on the 20 datasets. Finally, we illustrate the performance of Feature-based against 2D-NCR. Fig. 7(d) shows that 2D-NCR is superior to Feature-based on the datasets we tested, because more dots fall in the "Feature-based region" than in the

**Table 5**
Summary of the detailed error rate for each class on 10 datasets.

| Dataset | Classes | Method | Mean | Standard deviation | Max | Min |
|---|---|---|---|---|---|---|
| Synthetic Control | 6 | 2D-NCR | 0.0433 | **0.0345** | 0.08 | **0** |
| | | SAX | **0.02** | 0.049 | 0.12 | **0** |
| | | ED | 0.12 | 0.218 | 0.56 | **0** |
| CBF | 3 | 2D-NCR | **0.0022** | **0.0038** | **0.0066** | **0** |
| | | SAX | 0.104 | 0.0678 | 0.1633 | 0.0302 |
| | | ED | 0.148 | 0.1863 | 0.3609 | 0.02 |
| OSU Leaf | 6 | 2D-NCR | **0.3719** | 0.2796 | **0.6739** | **0.0227** |
| | | SAX | 0.467 | **0.2111** | 0.7174 | 0.1818 |
| | | ED | 0.483 | 0.243 | 0.7609 | 0.1818 |
| Trace | 4 | 2D-NCR | **0** | **0** | **0** | **0** |
| | | SAX | 0.46 | 0.1784 | 0.5862 | 0.2105 |
| | | ED | 0.24 | 0.2671 | 0.6071 | **0** |
| Two Patterns | 4 | 2D-NCR | 0.1447 | 0.0273 | 0.1721 | 0.1092 |
| | | SAX | **0.081** | **0.0233** | **0.1009** | **0.0483** |
| | | ED | 0.093 | 0.0264 | 0.1167 | 0.0599 |
| Face (four) | 4 | 2D-NCR | **0.0227** | **0.0376** | **0.0769** | **0** |
| | | SAX | 0.17 | 0.067 | 0.2273 | 0.0714 |
| | | ED | 0.216 | 0.0982 | 0.3077 | 0.0714 |
| Lightning-7 | 7 | 2D-NCR | **0.3424** | **0.2528** | **0.7778** | **0** |
| | | SAX | 0.397 | 0.3556 | 1 | **0** |
| | | ED | 0.425 | 0.3327 | 0.8889 | 0.1579 |
| Fish | 7 | 2D-NCR | **0.12** | **0.0929** | **0.2917** | **0** |
| | | SAX | 0.474 | 0.1881 | 0.6667 | 0.2273 |
| | | ED | 0.217 | 0.1204 | 0.375 | 0.0455 |
| Beef | 5 | 2D-NCR | **0.433** | **0.2789** | **0.6667** | **0** |
| | | SAX | 0.567 | 0.4183 | 1 | 0.1667 |
| | | ED | 0.467 | 0.3613 | 0.8333 | **0** |
| OliveOil | 4 | 2D-NCR | 0.2667 | **0.2398** | **0.6** | 0.0833 |
| | | SAX | 0.833 | 0.5 | 1 | **0** |
| | | ED | **0.133** | 0.3632 | 0.75 | **0** |

"2D-NCR region". Fig. 7(a)-(d) demonstrate that 2D-NCR obtains better performance than the other four methods do by a large margin, both in the number of dots and the distance of these dots from the line.

To investigate the detailed classification performance for each class on the datasets, we re-implement 2D-NCR, SAX, and Euclidean Distance on ten datasets, for which the numbers of classes range from 3 to 7, and summarize the detailed error rate. The source codes of time-series classification for SAX and Euclidean are downloaded from the available URL: http://www.cs.ucr.edu/~eamonn/SAX.htm. Table 5 shows the summary results including the mean, standard deviation, maximum, and minimum of error rate for each class on the datasets. 2D-NCR obtained the smallest value of the mean, standard deviation, maximum, and minimum error rate on 7, 9, 9, and 8 datasets, respectively. These measures suggest that 2D-NCR achieves good classification performance for each class with universality and stability.

From the experimental results above, we can state that 2D-NCR significantly outperforms state-of-the-art representations for time-series classification in much lower reduced space. Moreover, 2D-NCR also has better overall performance. Overall, 2D-NCR is an effective and competitive time-series representation for classification.

### *4.3. Clustering*

Time-series clustering can provides underpinning techniques for discovering the intrinsic structure conveyed in time-series datasets. Hierarchical clustering, a common clustering technique, is often chosen to contrast similarity measures. In this paper, we use the Unweighted Pair Group Method using arithmetic Averages (UPGMA) algorithm, which is based on group-average-linkage to compute the distance between any two clusters.

We cluster nine time series arbitrarily chosen from the Synthetic Control dataset, three each from the normal (1, 2, 3), cyclic (4, 5, 6), and upward shift (7, 8, 9) classes. To compare the clustering performance between 2D-NCR and other state-of-the-art methods (PWCA, PAA, and SAX) both in high and much lower reduced space, we conduct all of the methods on the nine time series by using the UPGMA algorithm and setting the reduced dimension $w = (3, 10)$. The comparisons of the four approaches' clustering results according to different reduced dimensions are shown in Fig. 8–9, which clearly show that 2D-NCR is superior, because it correctly assigns each class to its own subtree in the reduced spaces $w = (3, 10)$. Fig. 8 shows that the clustering qualities of PWCA and PAA are the same but better than that of SAX in the reduced space $w = 3$, whereas Fig. 9 shows that the clustering quality of PWCA is slightly better than those of SAX and PAA in the reduced space $w = 10$. These results are simply determined by the characteristics of the methods and inherent in the dataset. For example, the time series from the "upward shift" class has a growth trend overall, whereas the change of time series from the "cyclic class" is periodic. The "normal" class contains a great deal of noise, which makes the time series from the "normal" class have

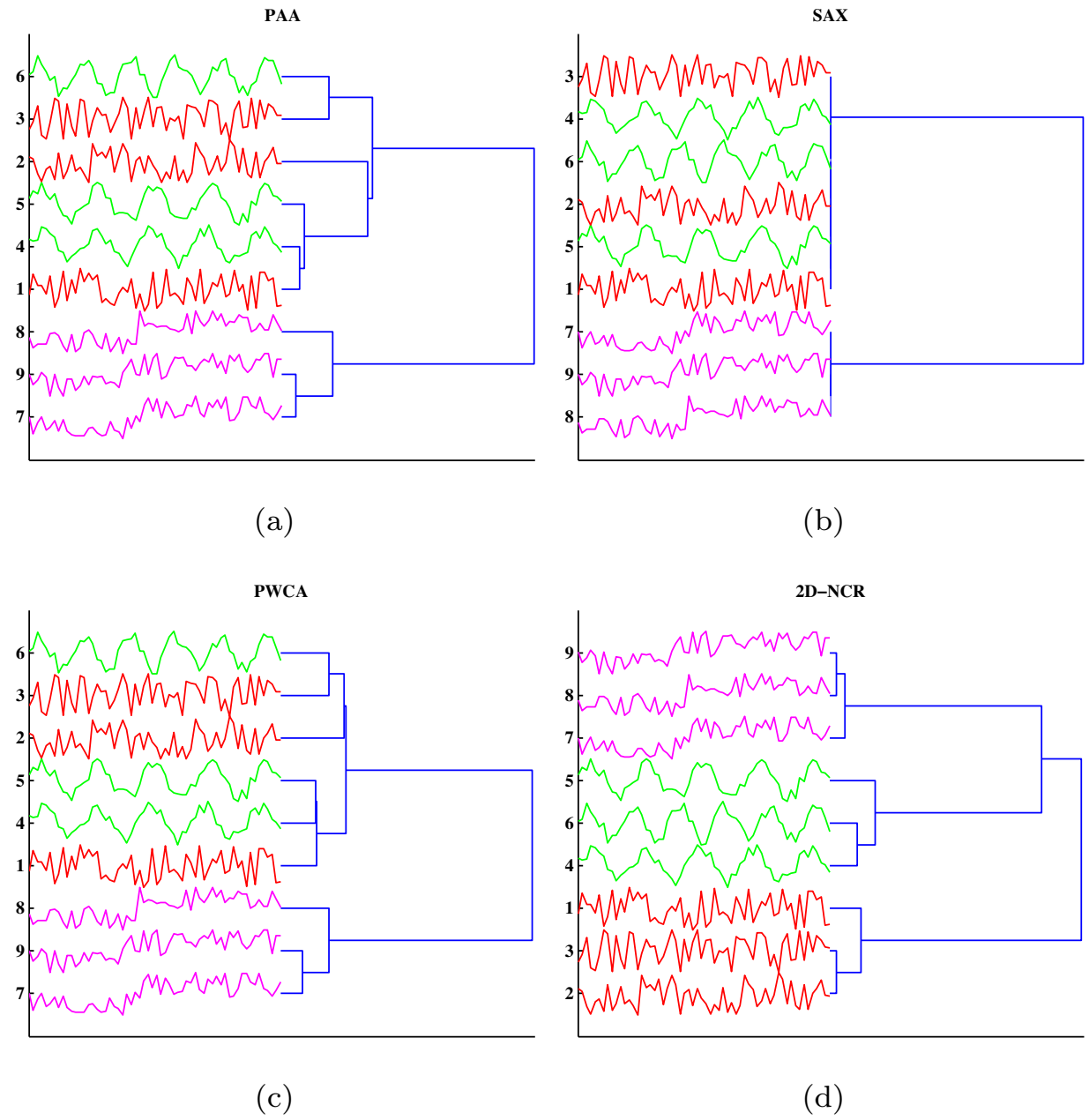


**Fig. 8.** A comparison of the four approaches' clustering results according to reduced dimension $w = 3$.

the characteristic of uncertainty. Consequently, 2D-NCR can address these characteristics effectively, whereas others cannot. Thus, we can objectively state that 2D-NCR outperforms the other three methods for clustering on the nine time series.

To validate further the clustering performance of 2D-NCR, we perform 2D-NCR on the Gun-Point dataset, Trace dataset, Synthetic Control dataset, and CBF dataset using the UPGMA algorithm, as DSA does in [19]. We also adopted the *F-measure*($F$) [34] as the evaluation criterion, which is defined as the harmonic mean between the information retrieval notions of precision ($P$) and recall ($R$), with the following expression:

$$F = \frac{2 \times P \times R}{P + R} \tag{12}$$

Given a set of time series $\mathcal{Z}$ , let $\Gamma = \{\Gamma_1, \ldots, \Gamma_H\}$ be the ground truth clusters of time series in $\mathcal{Z}$, and $\mathcal{C} = \{C_1, \ldots, C_k\}$ be the actual output of the clustering algorithm. The precision of $C_j$ with respect to $\Gamma_i$ is $P_{ij} = |C_j \cap \Gamma_i|/|C_j|$and the recall of

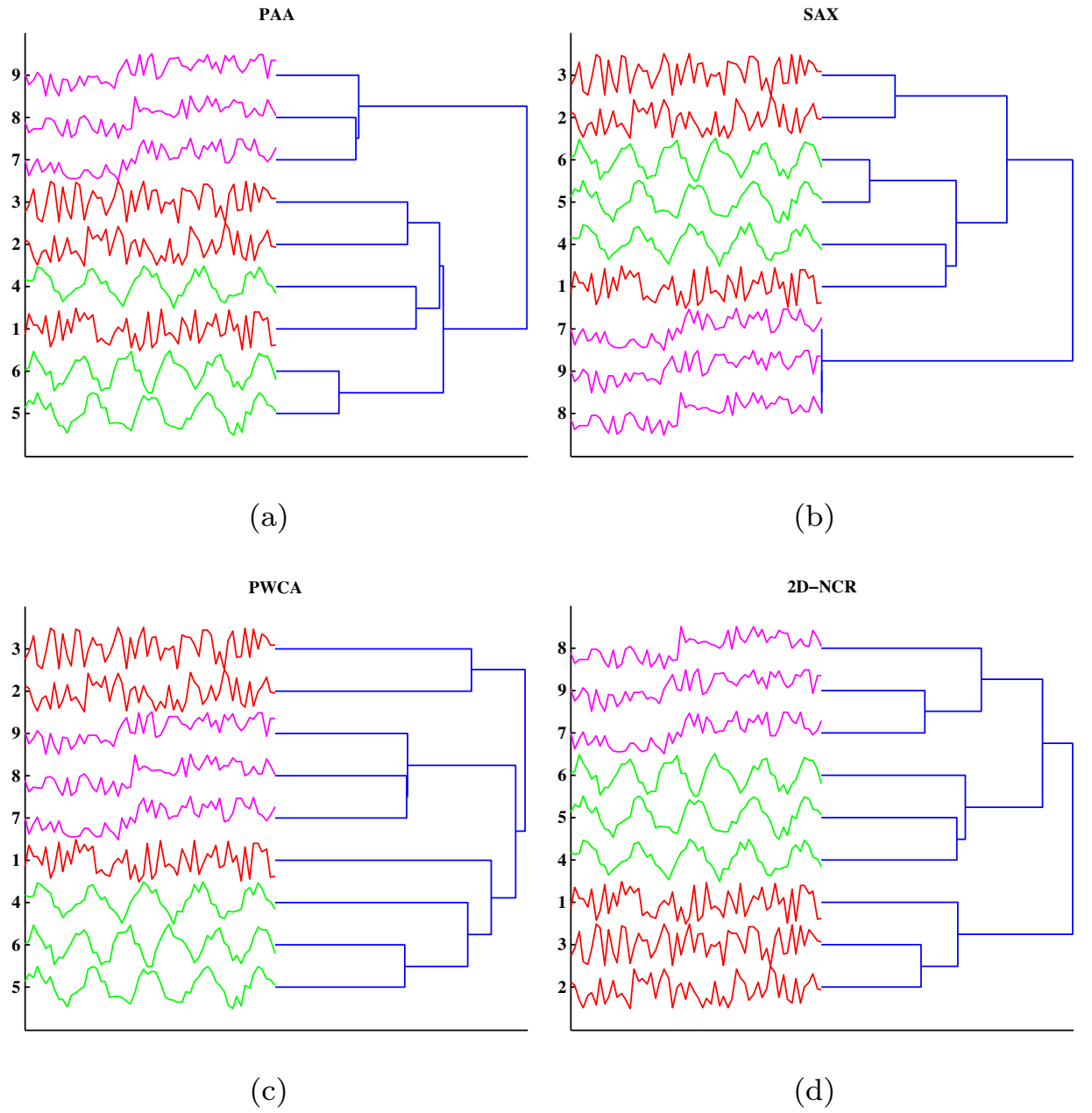


**Fig. 9.** A comparison of the four approaches' clustering results according to reduced dimension $w = 10$.

$C_j$ with respect to $\Gamma_i$ is $R_{ij} = |C_j \cap \Gamma_i|/|\Gamma_i|$. The overall precision and recall are defined as follows:

$$P = \frac{1}{H}\sum_{i=1}^{H} P_i, \quad R = \frac{1}{H}\sum_{i=1}^{H} R_i \tag{13}$$

where $P_i$ and $R_i$ are equal to $P_{ij*}$ and $R_{ij*}$, respectively, such that $j* \in \arg\max_{j=1,\ldots,K}\{P_{ij}, R_{ij}\}$.

Table 6 shows the experimental results, in which the results in the first five columns are derived from [19]. The *F-measure* score of 2D-NCR is much higher than that of other methods in all of the datasets (except Trace). However, in this dataset, 2D-NCR is only 0.06 less than that of DSA. To evaluate the clustering quality more intuitively, we rank the scores for all of the approaches shown in Table 7, which suggests that 2D-NCR obtains the best clustering quality in terms of average ranking with a value of 1.25. Moreover, the standard deviation of the ranking for 2D-NCR is the smallest with a value of 0.5, meaning that the clustering results have stability. We also employ the Friedman test to verify the statistical significance of the results. Table 8 shows the results of the Friedman test, showing that the $p$-value is 0.0009, far smaller than the significance level

**Table 6**
A comparison of quality results (*F-measure*) for UPGMA clustering.

| Dataset | DTW | PLA | PAA | SAX | DSA | PWCA | 2D-NCR |
|---|---|---|---|---|---|---|---|
| Gun-Point | 0.61 | 0.61 | 0.61 | 0.61 | 0.73 | 0.69 | **0.79** |
| Trace | 0.48 | 0.63 | 0.61 | 0.6 | **0.82** | 0.55 | 0.76 |
| Synthetic Control | 0.48 | 0.4 | 0.36 | 0.48 | 0.54 | 0.68 | **0.81** |
| CBF | 0.51 | 0.51 | 0.51 | 0.56 | 0.6 | 0.72 | **0.76** |

Note: the similarity measure used for PLA, PAA, SAX, and DSA is DTW.

**Table 7**
Ranking of different approaches for UPGMA clustering on the 4 datasets.

| Dataset | DTW | PLA | PAA | SAX | DSA | PWCA | 2D-NCR |
|---|---|---|---|---|---|---|---|
| Gun-Point | 5.5 | 5.5 | 5.5 | 5.5 | 2 | 3 | **1** |
| Trace | 7 | 3 | 4 | 5 | **1** | 6 | 2 |
| Synthetic Control | 4.5 | 6 | 7 | 4.5 | 3 | 2 | **1** |
| CBF | 6 | 6 | 6 | 4 | 3 | 2 | **1** |
| Average ranking | 5.75 | 5.125 | 5.125 | 4.75 | 2.25 | 3.25 | 1.25 |
| Standard deviation | 1.04 | 1.44 | 1.25 | 0.65 | 0.96 | 1.89 | 0.5 |

**Table 8**
Result of Friedman test for the rankings in Table 7.

| | Sum of Squares | Degree of Freedom | Mean Square | $\chi^2$ | *p*-value |
|---|---|---|---|---|---|
| Columns | 100.4 | 6 | 16.7333 | 22.82 | 0.0009 |
| Error | 31.6 | 24 | 1.3167 | | |
| Total | 132 | 34 | | | |

of 0.05. This measure implies that there is a statistically significant difference among these clustering results. Therefore, we can conclude that 2D-NCR is superior to state-of-the-art representations for time-series clustering on the datasets used in this study.

### 4.4. Query by content

Query by content is based on retrieving a set of time series that are most to the query series in the dataset. The approach has attracted much interest from the time-series data mining community. According to [13], query by content can be divided into two types of query: (i) *ε-range query*, returning the set of time series that are within distance $\varepsilon$ of the query series; (ii) *K-Nearest Neighbors query*, returning the top $K$ series closest to the query series.

In this study, we choose the "*K-Nearest Neighbors query*" as the experimental task to validate the performance of 2D-NCR. To compare with PWCA, PLA, and SAX, we perform the experiment on two datasets, Synthetic Control and Two Patterns, as in [25]. For Synthetic Control dataset, we arbitrarily choose $n(n = 66)$ as query times whereas the remaining ones are used as queried time series datasets. The *K-Nearest Neighbors query* experiments are conducted five times by setting the reduced dimensions $w = (2, 3, 6, 10, 15, 20)$, and then the relationship of results for different methods are analyzed using linear spline interpolation. At the same time, we set $K = 60$. In other words, every query operation of every query time series returns $K$ time series that nearest to the query series in the remaining group. Similarity, we set $n = 30$, $K = 100$, and $w = (1, 2, 4, 8, 16, 32)$ for Two Patterns dataset.

To evaluate the query precision, we calculate the rate of right query for every query operation by counting how many objects in the $K$ time series of which class label are the same to that of the corresponding query series, as the following expression.

$$fr_i = \frac{rs_i}{K} \tag{14}$$

where $fr_i$ denotes the rate of right query for the *ith* query series, and $rs_i$ is the number of time series in the $K$ time series of which class label are the same to that of the *ith* query series. Then, the precision ($P$) used to evaluate the performance for the *K-Nearest Neighbors query* can be calculated:

$$P = \frac{1}{n}\sum_{i=1}^{n} fr_i \tag{15}$$

Fig. 10 shows the comparison results of query of content by four approaches (2D-NCR, PWCA, PLA, and SAX) according to linear spline interpolation. 2D-NCR clearly, significantly outperforms the other three methods in Synthetic Control dataset. In Two Patterns dataset, 2D-NCR obtains higher precision than other methods in much lower reduced space ($w < 15$), but

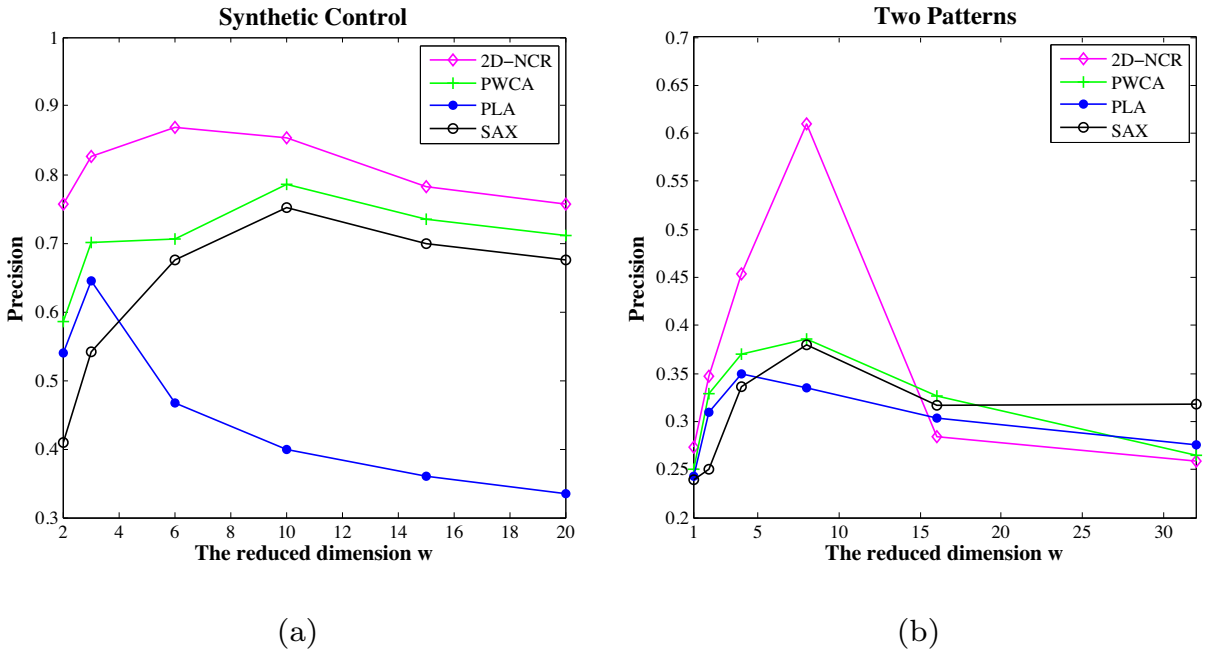


**Fig. 10.** A comparison of the results of *K-Nearest Neighbors query* according to linear spline interpolation. (a) Synthetic Control. (b) Two Patterns.

its performance is slightly worse than that of other methods with reduced dimension $w \geq 15$. Overall, 2D-NCR has a better performance of query of content than other methods particularly in much lower reduced space, which is consistent with [25], but our approach outperforms PWCA.

## 5. Conclusions

This paper focuses on time-series representation and similarity measure. We proposed a novel time-series representation, namely piecewise Two-dimensional Normal Cloud Representation (2D-NCR). The representation transformed the raw time series into a sequence of two-dimensional normal cloud model. Then a new similarity measure between the transformed time series has been presented. The proposed method can not only reflect the distribution of each segment but also provide the variation with time. In addition, it can address the vagueness and uncertainty inherent in certain time-series data. We experimentally validated 2D-NCR in three common tasks (classification, clustering, and query by content) of time-series data mining, and compared it to state-of-the-art dimensionality reduction methods. The contributions of this paper are: (i) the successful development of a new time-series representation; (ii) the successful development of a new similarity measure for the transformed time series; (iii) the successful implementation of the proposed method and comparing its performance with the competitive methods.

Although the proposed method shows better performance, the method itself suggests a host of future directions. A novel adaptive segmentation method based on the features of normal distribution can be developed to improve 2D-NCR. In addition, it might create a similarity measure using the six numerical characteristics of two-dimensional cloud model rather than four numerical characteristics used in this study. Finally, 2D-NCR can be applied into other tasks of time-series data mining such as anomaly detection and prediction.

## Acknowledgements

This work has been supported by the National Science and Technology Major Project of China under grant 2014ZX07104006, the National Naturel Science Foundation of China under grant numbers of 61272060 and 61572091. We would like to express thanks to Prof. Eamonn Keogh for his datasets and suggestions.

## References

[1] J. Ares, J.A. Lara, D. Lizcano, S. Suárez, A soft computing framework for classifying time series based on fuzzy sets of events, Inf. Sci. 330 (2016) 125–144.
[2] M.G. Baydogan, G. Runger, Time series representation and similarity based on local autopatterns, Data Min. Knowl. Discov. 30 (2) (2016) 476–509.
[3] N. Begum, E. Keogh, Rare time series motif discovery from unbounded streams, Proc. VLDB Endowment 8 (2) (2014) 149–160.
[4] K. Chan, A.W. Fu, Efficient time series matching by wavelets, in: Data Engineering, 1999. Proceedings of the 15th International Conference on, IEEE, 1999, pp. 126–133.

[5] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, The UCR Time Series Classification Archive, 2015, www.cs.ucr.edu/~eamonn/time_series_data/.
[6] Z. Chen, W. Zuo, Q. Hu, L. Lin, Kernel sparse representation for time series classification, Inf. Sci. 292 (2015) 15–26.
[7] R. Coppi, P. D'Urso, Fuzzy unsupervised classification of multivariate time trajectories with the Shannon entropy regularization, Comput. Stat. Data Anal. 50 (6) (2006) 1452–1477.
[8] H. Deng, G. Runger, E. Tuv, M. Vladimir, A time series forest for classification and feature extraction, Inf. Sci. 239 (2013) 142–153.
[9] W. Deng, G. Wang, X. Zhang, A novel hybrid water quality time series prediction method based on cloud model and fuzzy forecasting, Chemom. Intell. Lab. Syst. 149 (2015) 39–49.
[10] W. Deng, G. Wang, X. Zhang, J. Xu, G. Li, A multi-granularity combined prediction model based on fuzzy trend forecasting and particle swarm techniques, Neurocomputing 173 (2016) 1671–1682.
[11] P. D'Urso, Dissimilarity measures for time trajectories, Stat. Methods Appl. 1 (3) (2000) 53–83.
[12] P. D'Urso, Fuzzy clustering for data time arrays with inlier and outlier time trajectories, Fuzzy Syst. IEEE Trans. 13 (5) (2005) 583–604.
[13] P. Esling, C. Agon, Time-series data mining, ACM Comput. Surv. (CSUR) 45 (1) (2012) 12.
[14] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, Fast Subsequence Matching in Time-series Databases, 23, ACM, 1994.
[15] L.N. Ferreira, L. Zhao, Time series clustering via community detection in networks, Inf. Sci. 326 (2016) 227–242.
[16] E. Fuchs, T. Gruber, J. Nitschke, B. Sick, Online segmentation of time series based on polynomial least-squares approximations, Pattern Anal. Mach. Intell. IEEE Trans. 32 (12) (2010) 2232–2245.
[17] B.D. Fulcher, N.S. Jones, Highly comparative feature-based time-series classification, Knowl. Data Eng. IEEE Trans. 26 (12) (2014) 3026–3037.
[18] J. Grabocka, M. Wistuba, L. Schmidt-Thieme, Scalable classification of repetitive time series through frequencies of local polynomials, Knowl. Data Eng. IEEE Trans. 27 (6) (2015) 1683–1695.
[19] F. Gullo, G. Ponti, A. Tagarelli, S. Greco, A time series representation model for accurate and fast similarity detection, Pattern Recognit. 42 (11) (2009) 2998–3014.
[20] X. He, C. Shao, Y. Xiong, A non-parametric symbolic approximate representation for long time series, Pattern Anal. Appl. 19 (1) (2016) 111–127.
[21] D.T. Ho, J.M. Garibaldi, Context-dependent fuzzy systems with application to time-series prediction, Fuzzy Syst. IEEE Trans. 22 (4) (2014) 778–790.
[22] E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrotra, Dimensionality reduction for fast similarity search in large time series databases, Knowl.Inf. Syst. 3 (3) (2001) 263–286.
[23] M. Krawczak, G. Szkatuła, An approach to dimensionality reduction in time series, Inf. Sci. 260 (2014) 15–36.
[24] D. Li, Y. Du, Artificial Intelligence with Uncertainty, CRC Press, 2007.
[25] H. Li, C. Guo, Piecewise cloud approximation for time series mining, Knowl.-Based Syst. 24 (4) (2011) 492–500.
[26] J. Lin, E. Keogh, L. Wei, S. Lonardi, Experiencing SAX: a novel symbolic representation of time series, Data Min. Knowl. Discov. 15 (2) (2007) 107–144.
[27] S. Miao, U. Vespier, R. Cachucho, et al., Predefined pattern detection in large time series, Inf. Sci. 329 (2016) 950–964.
[28] W. Pedrycz, A. Gacek, Clustering granular data and their characterization with information granules of higher type, Fuzzy Syst. IEEE Trans. 23 (4) (2015) 850–860.
[29] K. Qin, K. Xu, F. Liu, D. Li, Image segmentation based on histogram analysis utilizing the cloud model, Comput. Math. Appl. 62 (7) (2011) 2824–2833.
[30] K. Ravi Kanth, D. Agrawal, A. Singh, Dimensionality reduction for similarity searching in dynamic databases, in: ACM SIGMOD Record, 27, ACM, 1998, pp. 166–176.
[31] R. Rosas-Romero, A. Díaz-Torres, G. Etcheverry, Forecasting of stock return prices with sparse representation of financial time series over redundant dictionaries, Expert Syst. Appl. 57 (2016) 37–48.
[32] H. Shatkay, S.B. Zdonik, Approximate queries and representations for large data sequences, in: Data Engineering, 1996. Proceedings of the 12th International Conference on, IEEE, 1996, pp. 536–545.
[33] Y. Sun, J. Li, J. Liu, B. Sun, C. Chow, An improvement of symbolic aggregate approximation distance measure for time series, Neurocomputing 138 (2014) 189–198.
[34] C.J. Van Rijsbergen, Information Retrieval, Butterworths, 1976.
[35] G. Wang, C. Xu, D. Li, Generic normal cloud model, Inf. Sci. 280 (2014) 1–15.
[36] G. Wang, J. Xu, Q. Zhang, Y. Liu, Multi-granularity intelligent information processing, in: Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Springer, 2015, pp. 36–48.
[37] H. Wang, Y. Cai, Y. Yang, S. Zhang, N. Mamoulis, Durable queries over historical time series, Knowl. Data Eng. IEEE Trans. 26 (3) (2014) 595–607.
[38] H. Wang, M. Tang, Y.-S. Park, C.E. Priebe, Locality statistics for anomaly detection in time series of graphs, Signal Process. IEEE Trans. 62 (3) (2014) 703–717.
[39] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, E. Keogh, Experimental comparison of representation methods and distance measures for time series data, Data Min. Knowl. Discov. 26 (2) (2013) 275–309.
[40] C. Xu, G. Wang, Backward cloud transformation algorithm for realizing stability bidirectional cognitive mapping, Pattern Recognit. Artif. Intell. 26 (2013) 634–642.
[41] C. Yang, D. Li, Planar model and its application in prediction, Chin. J. Comput. 21 (11) (1998) 961–969.
[42] F. Ye, L. Zhang, D. Zhang, et al., A novel forecasting method based on multi-order fuzzy time series and technical analysis, Inf. Sci. 367 (2016) 41–57.
[43] H. Zhao, Z. Dong, T. Li, X. Wang, C. Pang, Segmenting time series with connected lines under maximum error bound, Inf. Sci. 345 (2016) 1–8.