Expert Systems with Applications 38 (2011) 11891–11900

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A novel clustering method on time series data

Xiaohang Zhang [*], Jiaqi Liu, Yu Du, Tingjie Lv

School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing 100876, China

ARTICLE INFO

Keywords:
Time series
Clustering
Dynamic time warping
Nearest neighbor network

ABSTRACT

Time series is a very popular type of data which exists in many domains. Clustering time series data has a wide range of applications and has attracted researchers from a wide range of discipline. In this paper a novel algorithm for shape based time series clustering is proposed. It can reduce the size of data, improve the efficiency and not reduce the effects by using the principle of complex network. Firstly, one-nearest neighbor network is built based on the similarity of time series objects. In this step, triangle distance is used to measure the similarity. Of the neighbor network each node represents one time series object and each link denotes neighbor relationship between nodes. Secondly, the nodes with high degrees are chosen and used to cluster. In clustering process, dynamic time warping distance function and hierarchical clustering algorithm are applied. Thirdly, some experiments are executed on synthetic and real data. The results show that the proposed algorithm has good performance on efficiency and effectiveness.

1. Introduction

Time series is a very popular type of data which exists in many domains. Clustering time series data has a wide range of applications and has attracted researchers from a wide range of discipline. For example, many researchers cluster the time series data which come from brain activity (Golay et al., 1998; Wismüller et al., 2002), commercial consumption (Košmelj & Batagelj, 1990), retail pattern (Kumar, Patel, & Woo, 2002), gene expression (Möller-Levet, Klawonn, Cho, & Wolkenhauer, 2003), earthquake (Shumway, 2003), financial data (Guan & Jiang, 2007), robot sensor data (Ramoni, Sebastiani, & Cohen, 2000) and speaker verification (Tran & Wagner, 2002) and so on. There are three main objectives in clustering time series, each of which requires different approaches (Bagnall & Janacek, 2005).

1.1. Type 1 objective: similarity in time

The first possible objective is to cluster together series that vary in a similar way on each time step. For example, one may want to cluster share prices of companies to discover which shares change in price together. Fig. 1 shows two clusters in each of which two time series change in a similar way on each time step.

1.2. Type 2 objective: similarity in shape

The second possible objective is to cluster series with common shape features together. This may constitute identifying common trends occurring at different times or similar sub patterns in the data. For example, the stock analyst may be interested in grouping shares that have exhibited similar patterns of change independent of when they occurred. The two broad approaches to achieving this objective are to either transform the data using techniques such as dynamic time warping, or to develop specific algorithms for matching subsequence patterns. Fig. 2 gives three clusters of time series each of which has typical characteristics in shapes.

1.3. Type 3 objective: similarity in change

The third objective is to cluster series by the similarity in how they vary from time step to time step. For example, a stock analyst may wish to cluster together shares that tend to follow a rise in share price with a fall the next day. The popular approach for this type of objective is to assume some form of underlying model such as a hidden Markov model or an ARMA process, fit a model to the series and cluster based on similarity of fitted parameters. Fig. 3 shows two time series clusters. The left one produced by AR(2) model is different from the right one produced by ARMA(2, 2). Although the series belonging to same cluster have different shapes, they vary in similar way on each time step which means that each time point depends on the value of specified earlier time steps in similar way.

In this paper a novel algorithm for shape based time series clustering is proposed. It can reduce the size of data, improve the efficiency and not reduce the effects by using the principle of complex network. Firstly, one-nearest neighbor network is built based on the similarity between any pair of time series. In this step, triangle distance is used to measure the similarity. Of the neighbor network each node represents one time series and each link

[*] Corresponding author. Tel.: +86 10 62283651; fax: +86 10 62282039.
E-mail address: zhangxiaohang@bupt.edu.cn (X. Zhang).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.03.081

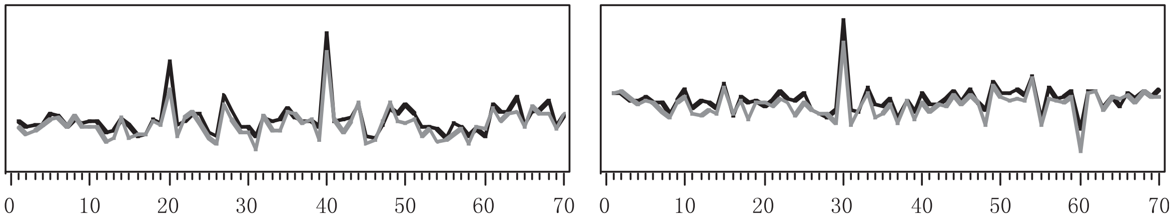

Fig. 1. Two clusters of time series for similarity in time.

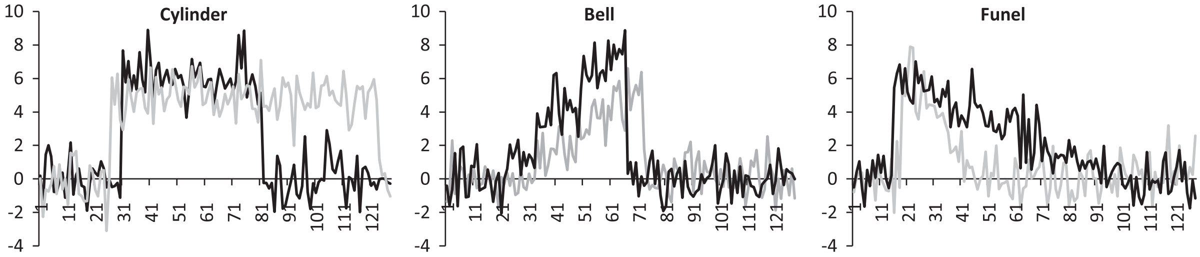

Fig. 2. Three clusters of time series each of which has same shape.

denotes nearest neighbor relationship between nodes. Secondly, the nodes with high number of neighbors are chosen as candidate objects and used to cluster. In clustering process, dynamic time warping distance function and hierarchical clustering algorithm are applied to these chosen candidate objects. Some experiments are executed on synthetic and real data. The results show that the proposed algorithm has good performance on efficiency and effects.

The rest of this paper is organized as follows. Background of the underlying theory and a survey of previous works in this area are given in Section 2. In Section 3 the key points of shape based time series clustering and triangle similarity and dynamic time warping measures are given. Also the experimental data used in this paper are described. In Section 4 the process of building nearest neighbor network and the algorithm of choosing candidates are proposed. And the effects of choosing candidates are discussed. The hierarchical clustering method and the experimental results are discussed in Section 5. The conclusions are presented in Section 6.

2. Literature

The two time series being compared are normally sampled at the same interval, but their length might or might not be the same. Clustering algorithms can work directly with raw data. However, it is usual that the time series data are preprocessed before clustering. There are many methods can be used to transform the raw data including principle component analysis (Gavrilov, Anguelov, Indyk, & Motwani, 2000), piecewise aggregate approximation (Yeh, Dai, & Chen, 2007), discrete Fourier transformation (Agrawal, Faloutsos, & Swami, 1993; Janacek, Bagnall, & Powell, 2005), discrete wavelet transformation (Struzik & Siebes, 1999; Yin & Gaber, 2008), clipping (Bagnall & Janacek, 2005; Bagnall, Ratanamahatana, Keogh, Lonardi, & Janacek, 2006). Then the transformed data serve as the inputs to clustering algorithm. Usually transformation of raw data can improve the efficiency by reducing the dimensions of data or improve the clustering effects by smoothing the trend and giving prominence to the typical features.

One key component of clustering is the function used to measure the similarity between data being compared. In practices and researches of clustering time series many measures were employed, such as Euclidean distance, Pearson's correlation coefficient, short time series distance (Möller-Levet et al., 2003), dynamic time warping (Hu, Ray, & Han, 2006; Yu, Dong, Chen, Jiang, & Zeng, 2007), probability-based distance (Kumar et al., 2002), KL distance (Dahlhaus, 1996) and J divergence (Shumway, 2003). In one survey of time series clustering (Liao, 2005), the formulas of various measures were given.

A lot of algorithms have been developed to cluster different types of time series data. In spirit they try to modify the existing algorithms for clustering static data in such a way that time series data can be handled or to convert time series data into the form of the static data so that the existing algorithms for clustering static data can be directly used. The popular clustering algorithms, generally also used in clustering time series data, include K-means (Beringer & Hüllermeier, 2006; Lin, Vlachos, Keogh, & Gunopulos, 2004), hierarchical clustering, density-based clustering (Chandrakala & Sekhar, 2008) and model-based clustering. Among them model-based clustering composes of polynomial (Bagnall & Janacek, 2005), ARIMA (Corduas & Piccoloa, 2008; Kalpakis, Gada, & Puttagunta, 2001), Hidden Markov models (Bicego, Murino, & Figueiredo, 2003; Hu et al., 2006), Gaussian mixed models

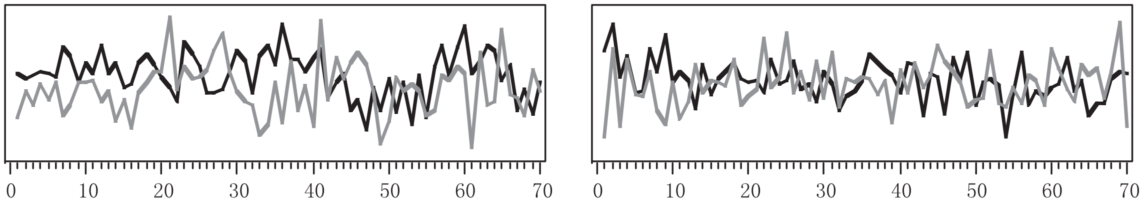

Fig. 3. Two clusters of time series data. The left is produced by AR(2) and the right is produced by ARMA(2, 2).

(Biernacki, Celeux, & Govaert, 2000), Markov chain (Ramoni et al., 2000).

3. Notations and experimental data

3.1. Time series clustering

For concreteness, we define the data type of interest, time series. Time series dataset U consists of n time series objects, $U = \{o_1, o_2, \ldots, o_n\}$. And each time series object is an ordered set of t real values. The goal of clustering is to identify structure in an unlabeled dataset by objectively organizing data into homogeneous groups where within-group-object similarity is minimized and the between-group-object dissimilarity is maximized. In order to improve performance on time series data, a lot of problems must be taken into account including high dimensionality, very high feature correlation, and large amounts of noise. Importantly for shape based time series clustering some of the various properties can be often encountered, which include noise, amplitude scaling, offset translation, longitudinal scaling, linear drift, discontinuities and temporal drift. These properties are also mentioned and used by other researchers in this domain and are generally accepted in verifying the validation of distance measures and algorithms (Agrawal, Lin, Sawhney, & Shim, 1995; Bagnall & Janacek, 2005; Chu & Wong, 1999; Kalpakis et al., 2001; Keogh & Pazzani, 1998; Man & Wong, 2001; Perng, Wang, Zhang, & Parker, 2000; Rafiei & Mendelzon, 1997). For clarity these properties are illustrated in a visual way (shown in Fig. 4) by reference to Keogh's work (Keogh & Pazzani, 1998) based on which we add the subfigure of temporal drift.

3.2. Similarity/distance measures

The function used to measure the similarity or distance between two time series objects is one key component in the time series clustering algorithm. In this paper, two measures are employed which are triangle similarity and dynamic time warping (DTW) distance.

3.2.1. Triangle similarity

Let o_i be a t-dimensional time series object, $o_i = \{o_{i1}, o_{i2}, \ldots, o_{it}\}$. The standardized time series object $\widehat{o_i} = \{\hat{o}_{i1}, \hat{o}_{i2}, \ldots, \hat{o}_{it}\}$, where

$$\hat{o}_{ij} = \frac{o_{ij}}{\left(\sum_{k=1}^{t} o_{ik}^2\right)^{1/2}}, \quad j = 1, 2, \ldots, t. \tag{1}$$

The triangle similarity measure between o_i and o_j is defined as

$$d_T(o_i) = \frac{\sum_{k=1}^{t} o_{ik} o_{jk}}{\left(\sum_{k=1}^{t} o_{ik}^2\right)^{1/2} \left(\sum_{k=1}^{t} o_{jk}^2\right)^{1/2}} = \sum_{k=1}^{t} \hat{o}_{ik} \hat{o}_{jk}. \tag{2}$$

Each time series object can be treated as a vector in t-dimensional space. Triangle distance measure is the cosine of triangle between two vectors, so the range of value of triangle distance is from -1 to 1. When the two vectors are overlapping and have same directions, the value is 1 and two time series objects are most similar to each other. Otherwise when the two vectors are overlapping and have opposite directions and are most different to each other, the value is -1. Triangle similarity measure can deal with noise, amplitude scaling very well and deal with offset translation, linear drift well in some situations (Zhang, Wu, Yang, Ou, & Lv, 2009).

3.2.2. Dynamic time warping distance (DTW)

DTW is a generalization of classical algorithms for comparing discrete sequences to sequences of continuous values. Given two time series, $o_i = \{o_{i1}, o_{i2}, \ldots, o_{in}\}$ and $o_j = \{o_{j1}, o_{j2}, \ldots, o_{jm}\}$, DTW aligns the two series so that their differences is minimized. We can get a $n \times m$ matrix where the (s, k) element of the matrix is the distance $d(o_{is}, o_{jk})$ between two time points o_{is} and o_{jk}. The Euclidean distance is used. A warping path, $W = w_1, w_1, \ldots, w_K$ where $\max(m, n) \leqslant K \leqslant m + n - 1$, is a set of matrix elements that satisfies three constraints: boundary condition, continuity and monotonicity. The boundary condition constraint requires the warping path to start and finish in diagonally opposite corner cells of the matrix. That is $w_1 = (1, 1)$ and $w_k = (m, n)$. The continuity constraint restricts the allowable steps to adjacent cells. The monotonicity constraint forces the points in the warping path to be monotonically spaced in time. The warping path that has the minimum distance between the two series is of interest. Mathematically,

$$d_{DTW} = \min\left(\sum_{k=1}^{K} w_K / K\right). \tag{3}$$

Dynamic programming can be used to effectively find this path by evaluating the following recurrence, which defines the cumulative distance as the sum of the distance of the current element and the minimum of the cumulative distance of the adjacent elements:

$$\begin{aligned} d_{cum}(s, k) = d(o_{is}, o_{jk}) + \min\{&d_{cum}(s-1, k-1), \\ &d_{cum}(s-1, k), d_{cum}(s, k-1)\}. \end{aligned} \tag{4}$$

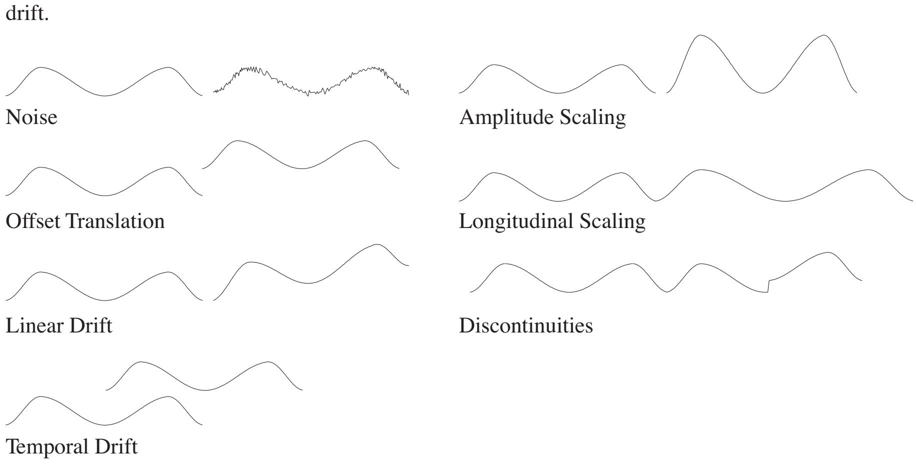

Fig. 4. Some of the difficulties encountered in defining a distance measure for time series.

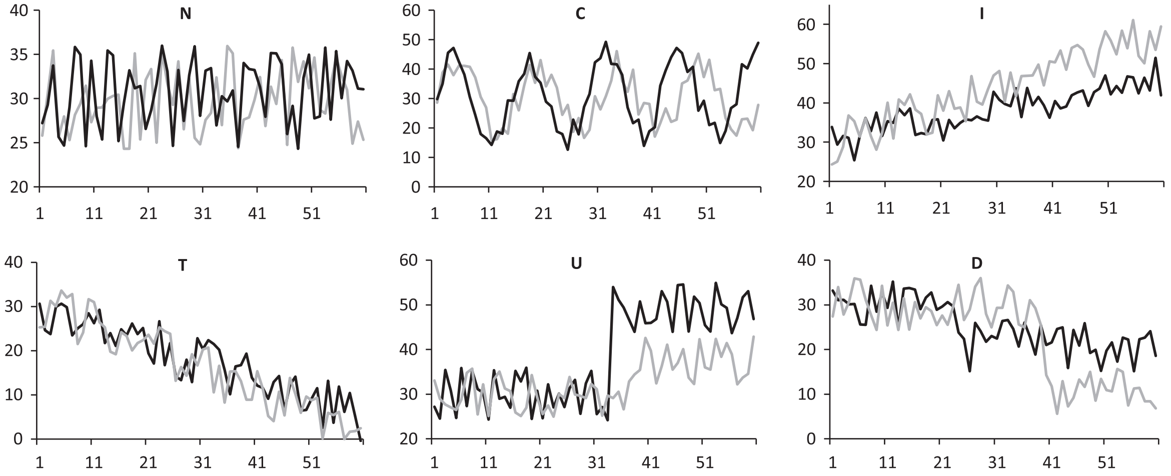

Fig. 5. Two examples of each class from CC dataset.

The superiority of DTW over Euclidean distance has been demonstrated by several authors (Aach & Church, 2001; Chu, Keogh, Hart, & Pazzani, 2002; Vlachos, Gunopoulos, & Kollios, 2002; Yi & Faloutsos, 2000). Keogh shows a classification experiment on time series data, which verifies that classification error rate of Euclidean distance is an order of magnitude higher than DTW's (Keogh & Ratanamahatana, 2005). DTW's advantage is that it can deal with temporal drift very well. In this point DTW is superior over Euclidean distance and triangle similarity. However the performance of DTW on very large database may be a limitation (Berndt & Clifford, 1994). Computation cost of DTW is much higher than Euclidean distance and triangle similarity.

3.3. Experimental data

In our experiments, we use three datasets which are also used by several other researchers in the same domain.

Control chart (CC): This dataset was proposed in Alcock and Manolopoulos (1999) to validate clustering techniques and used in Geurts (2001) for classification. Each series is an order set of sixty real values and classified into one of six possible classes. Each series is produced by

$$a(o,t)=\begin{cases} m+rs, & \textit{if}\ c(o)=\textit{Normal}\\ m+rs+a\sin(2\pi t/T), & \text{if}\ c(o)=\textit{Cyclic}\\ m+rs+gt, & \text{if}\ c(o)=\textit{Increasing}\\ m+rs-gt, & \text{if}\ c(o)=\textit{Decreasing}\\ m+rs+kx, & \text{if}\ c(o)=\textit{Upward}\\ m+rs-kx, & \text{if}\ c(o)=\textit{Downward} \end{cases} \tag{5}$$

Fig. 5 shows two examples of each class from which we can find that these time series have such characteristics as shown in Fig. 4.

Cylinder–Bell–Funnel (CBF): This problem was first introduced in Kadous (1999) and then used in Gonzalez and Diez (2000) for validation. Each series is separated into one of three classes: Cylinder (C), Bell (B) and Funnel (F).

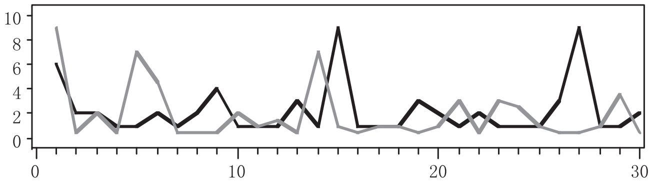

Fig. 6. Two series of telecom customers' daily calls in one month.

Each series is described by one temporal attribute given by

$$a(o,t)=\begin{cases} (6+\eta)\cdot\chi_{[a+b]}(t)+\epsilon(t), & \text{if}\ c(o)=C\\ (6+\eta)\cdot\chi_{[a+b]}(t)\cdot(t-a)/(b-a)+\epsilon(t), & \text{if}\ c(o)=B\\ (6+\eta)\cdot\chi_{[a+b]}(t)\cdot(b-t)/(b-a)+\epsilon(t), & \text{if}\ c(o)=F \end{cases} \tag{6}$$

where $t\in[1,128]$ and $\chi_{[a+b]}(t)=1$ if $a\leqslant t\leqslant b$, 0 otherwise. In the original problem, η and $\epsilon(t)$ are drawn from a standard normal distribution $N(0,1)$, a is an integer drawn uniformly from [16, 32] and $b-a$ is an integer drawn uniformly from [32, 96]. Each object is created based on different parameters, which means that before one object is created, all parameters including η, a, $b-a$ must be randomly redrawn according to their probability distributions. Then this series can be created by formula. Fig. 2 shows two examples of each class from which we can find that these time series have such characteristics as shown in Fig. 4.

Telecom customers' consumption data (TCCD): This dataset comes from one mobile company in China mainland. Each series is composed of 30 time points each of which denotes one customer's daily call counts in one month. There are 86,802 series in this dataset. Not like CBF and CC datasets, the series of this dataset are not clustered in advance, so it can not be used to verify the final clustering results. Fig. 6 shows the series of two customers' daily calls in one month.

In the experiments CC and CBF are all synthetic datasets and each series object in them belongs to one of several classes, so they can be used to verify the clustering effects by comparing each object's class with its cluster.

4. One-nearest neighbor network

In order to reduce the size of data and improve the efficiency but not reduce the effects, some ideas of complex network (Albert & Barabási, 2002; Boccalettia, Latorab, Morenod, Chavezf, & Hwanga, 2006; Watts, 2004) are borrowed to build one-nearest neighbor network. The network is built based on the similarity between any pair of time series, of which each node represents one time series and each link denotes one-nearest neighbor relationship between nodes. Based on some statistical properties of one-nearest network, some series are selected as candidates for further clustering. In this section, one-nearest neighbor network and its' statistical properties are defined and the process of selecting candidates is given. The following experiments verify that this process can reduce the size of data by approximate ten percent, but not reduce the effects greatly.

4.1. One-nearest neighbor network and its degree distribution

4.1.1. Definitions

A weighted and directed graph $G^W = (N, L, W)$ consists of a set $N = \{n_1, n_2, \ldots, n_N\}$ of nodes, a set $L = \{l_1, l_2, \ldots, l_m\}$ of links, and a set of weights $W = \{w_1, w_2, \ldots, w_m\}$. Each of links is defined by a couple of nodes i and j, and is denoted as l_{ij} which stands for a link from i to j, and $l_{ij} \neq l_{ji}$. The degree k_i of a node i is the number of links incident with the node. Each of weights is a real number and attached to one link. In a directed graph, the degree of the node has two components: the number of outgoing links k_i^{out} (referred to as the out-degree of the node), and the number of ingoing links k_i^{in} (referred to as the in-degree of the node). The most basic topological characterization of a graph can be contained in terms of the degree distribution $P(k)$, defined as the probability that a node chosen uniformly at random has degree k or, equivalently, as the fraction of nodes in the graph having degree k. Node strength s_i of a node i is defined as

$$s_i = \sum_{j \in N} w_{ij} \tag{7}$$

Neighbor network can be represented as a weighted and directed graph. In one-nearest neighbor network, each node represents one time series and each link denotes one-nearest neighbor relationship between nodes. In other words l_{ij} denotes that node j is the nearest neighbor of node i in terms of similarity defined in formula (2). The weight w_{ij} of link l_{ij} is defined as the similarity measure between node i and node j, so $w_{ij} \in [-1, 1]$. The out-degree of any node is 1 and the in-degree can be different. The nodes with high in-degree have more neighbors and locate in the local center of the network.

4.1.2. Degree distribution of one-nearest neighbor network

Some statistical properties of one-nearest neighbor networks of all experimental datasets, including degree distribution and correlation between node strength and degree, are computed to understand the networks. The size of CC dataset is specified by different numbers from 500 to 9000 and the size of CBF dataset ranges from 300 to 9000. And each class of the dataset has the same number of objects. The experimental results show that in all neighbor networks with different dataset size, the correlation between strength and degree approaches to 1.

The in-degree distributions of one-nearest neighbor network of all the three datasets are shown in Fig. 7. Note that all the figures are drawn in logarithm style. It can be seen that for different sizes of networks the in-degree follow the power-law distribution described by

$$P(k) \sim k^{-\alpha} \tag{8}$$

where k is in-degree and α is referred as degree exponent. The power-law distribution means that the in-degrees of the nodes for all networks are highly right skewed, in other words, their distribution has a long right tail of values that are far above the mean. So a little part of nodes has much bigger in-degree than other nodes and has central roles in terms of similarity, which implies that except for themselves they can also represent the typical characters of

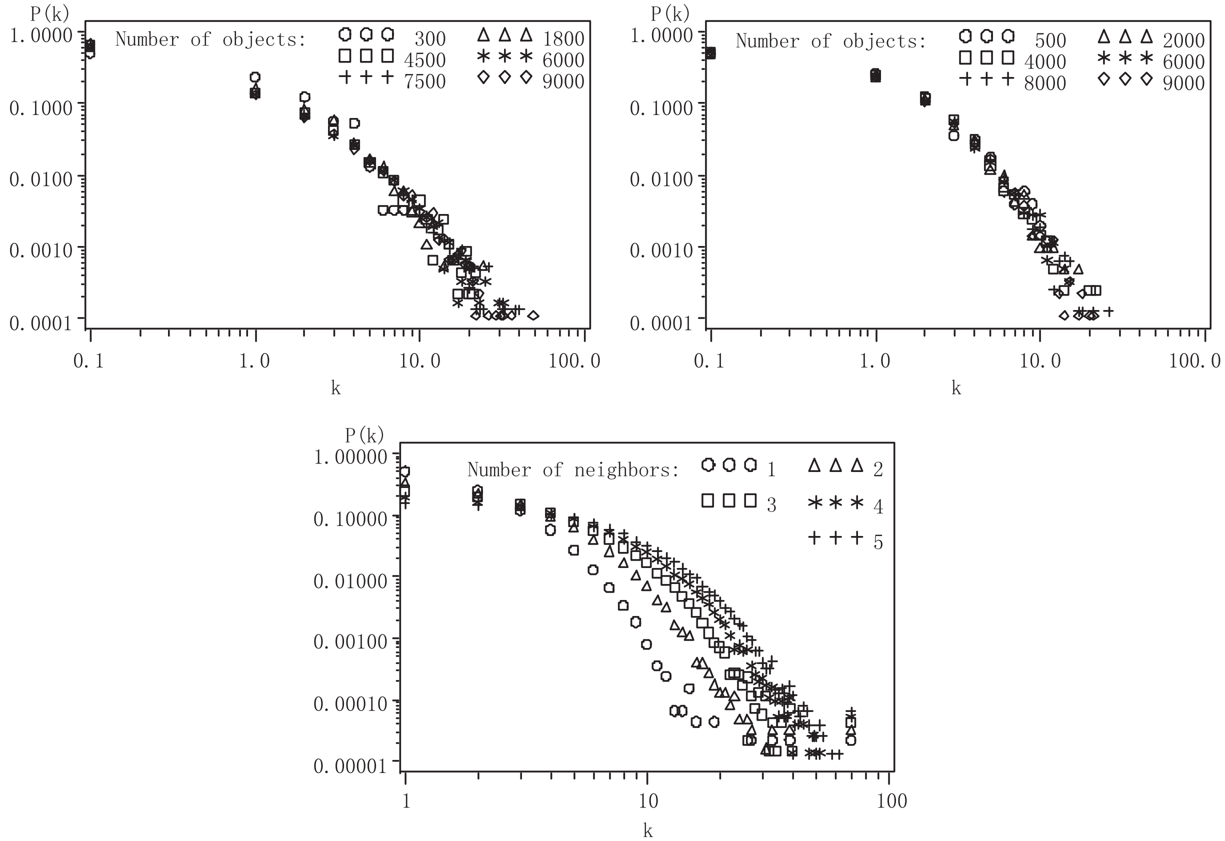

Fig. 7. In-degree distributions of the neighbor networks. The left top figure describes the in-degree distribution of CBF networks with different sizes. The right top figure describes the in-degree distribution of CC networks with different sizes. The bottom figure describes the in-degree distribution of TCCD networks with different number of nearest neighbors.

their neighbors. For TTCD dataset, k-nearest neighbor networks (k, number of nearest neighbors) with different k values are built and the in-degree distributions are drawn and can be also found to follow power-law distribution.

In order to further confirm the power-law distribution of in-degree, some experiments are carried out. For CC dataset and CBF dataset the degree exponent and R-square fit goodness are computed. And for each specified size of network, 50 networks are built based on formulas (5) and (6) and their statistical measures are computed. In Fig. 8 the box chart of degree exponent and fitted R-square for each network are given. The top two figures are for CBF dataset. It can be seen that as the size of network increases, the degree exponent increases firstly and become stable around 2.6 and its variance decreases. Except for some outliers of network with size of 300, R-square of any network is above 0.9 which means that power-law distribution is a good fit. CC networks' degree exponent approximately stabilizes at 3.0 and R-squares of most networks are above 0.9.

In most applications of the real world, the number of objects is so great that it is very time-spending to find the nearest neighbor. The complexity of finding nearest neighbor globally is °(n^2), n is number of objects. In order to reduce the costs of finding nearest neighbor, the objects can be firstly clustered into some clusters by method of k-means. Number of clusters k is assigned by $\sqrt{n/2f}$ where f is number of scanning dataset in k-means process. And then the task of finding nearest neighbor for each object is limited into its cluster, which reduces the searching area greatly. The complexity is °($n^{3/2}$) which is approved in Appendix A. Because the results of k-means depend on the initial seeds, 50 different networks of TTCD are built with different initial seeds. We can see that

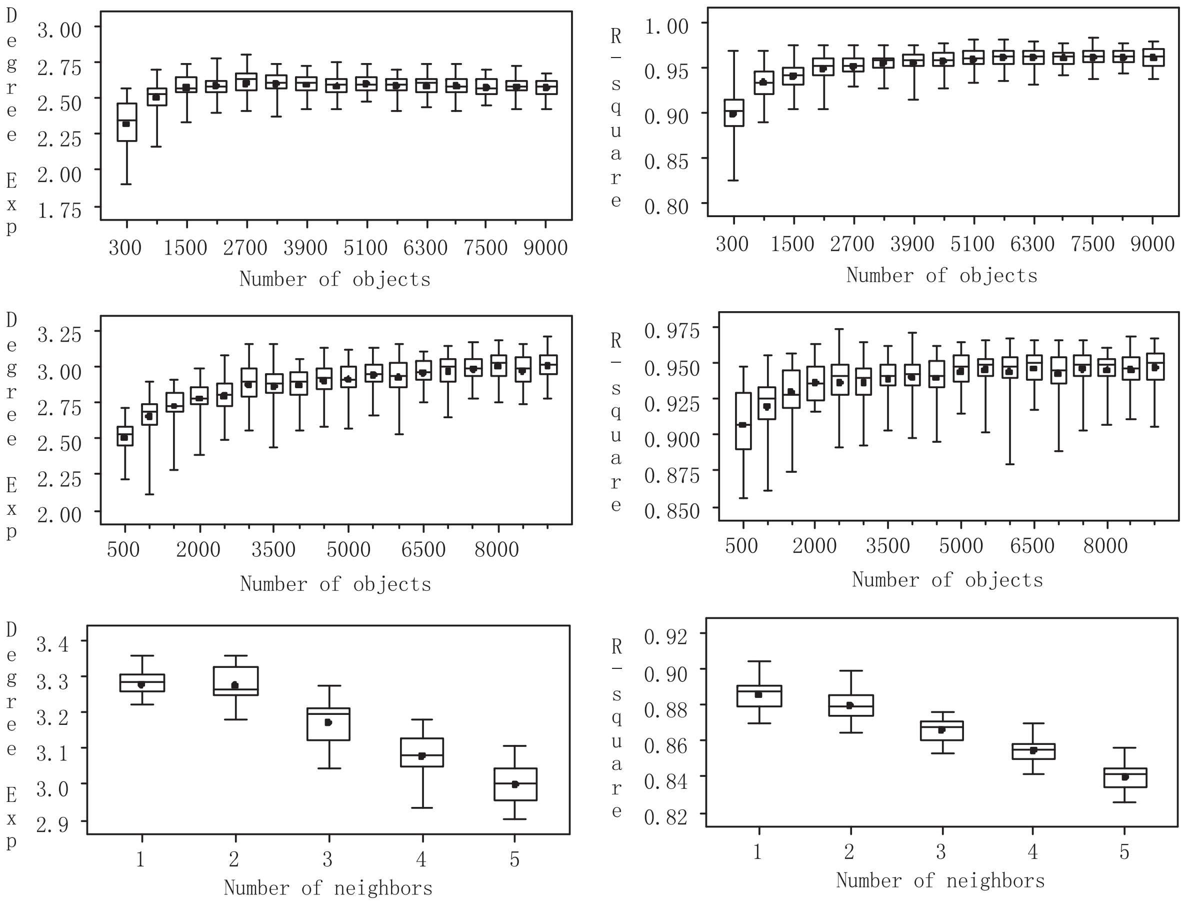

Fig. 8. Degree exponents of neighbor networks. The top two figures are box charts of degree exponents and R-square fit goodness of degree distribution for CBF datasets with different sizes. The middle two figures are for CC datasets. The bottom figures are for TCCD dataset with different number of neighbors.

```
Choosing candidates( k) //k is the maximum order of neighbors, which is specified in advance
{
    Sorting all nodes into a node set N by their in-degree descending ;
    While N is not empty
        Choosing one node n which belongs to N and has biggest in-degree;
        Adding n into candidate set C and removing n from N;
        Choosing all neighbors of n from first order to k^th order in N, then removing these neighbors from N and adding them into
            neighbor set NS;
    Return candidate set C;
}
```

Fig. 9. The algorithm of choosing candidates.

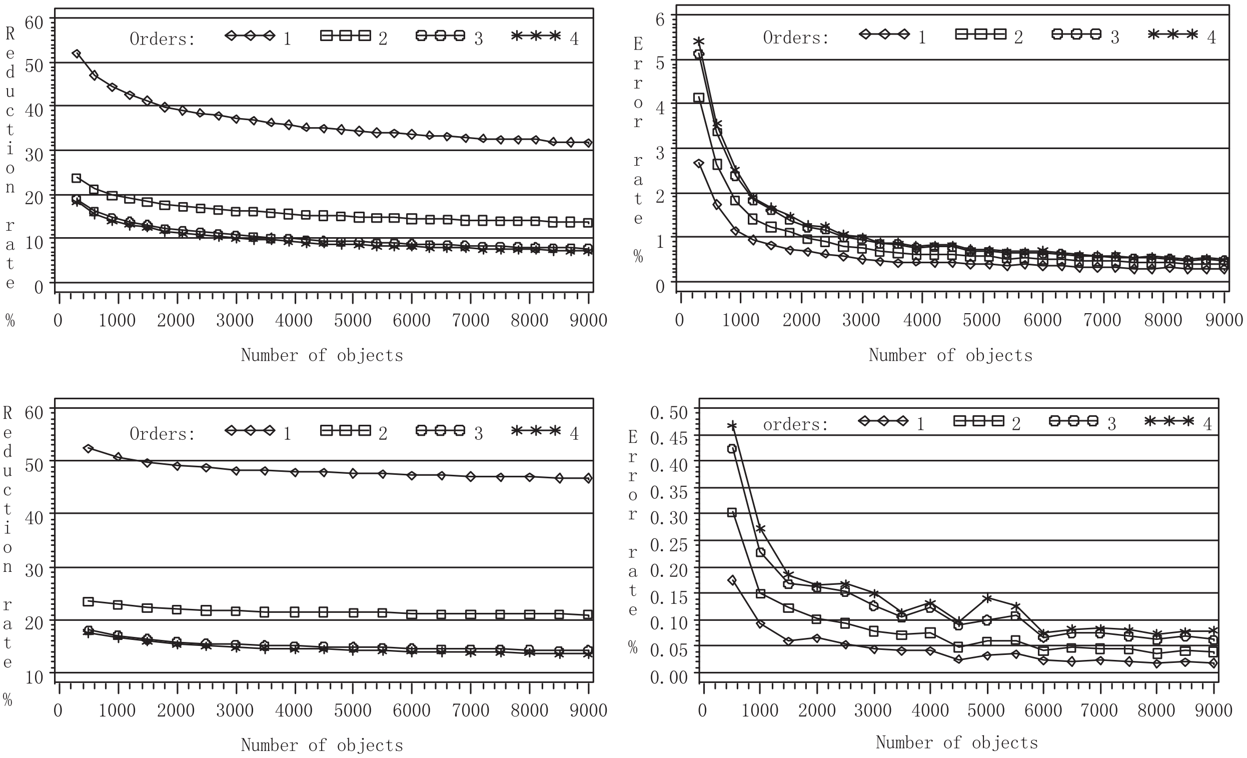

Fig. 10. Reduction rates and error rates of choosing candidate objects. The top two figures are for CBF dataset. The bottom figures are for CC dataset.

as the number of neighbors increases, the degree exponent and R-square for TTCD all decrease. Degree exponent ranges from 2.9 to 3.3 and R-square are all above 0.82.

4.2. Choosing candidates for clustering

4.2.1. Algorithm

Because in-degree follows the power-law distribution, some nodes with bigger in-degree can be chosen as candidates who can reflect the typical characters of their neighbors. So the task of clustering the whole dataset is transformed to clustering the candidates. There is no doubt that clustering candidates can reduce the cost of computation. Of course we need to find out whether clustering candidates will affect the effects greatly. The process of choosing candidates is based on nearest neighbor's order which is defined as the distance in the nearest neighbor network. For example, if series A is nearest neighbor of series B and series C is nearest neighbor of series B, then B is first order nearest neighbor of A, C is first order nearest neighbor of B and is second order nearest neighbor of A. The algorithm of choosing candidates is shown in Fig. 9. The chosen candidates are used as the inputs of further hierarchical clustering. All neighbors of each candidate will belong to the same cluster with the candidate.

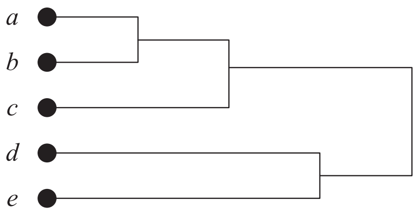

Fig. 11. A sample of dendrogram representation for hierarchical clustering of data objects {*a*, *b*, *c*, *d*, *e*}.

4.2.2. Analysis

Whether clustering candidates has effects on final clustering results can be evaluated from two aspects. One is the reduction rate which is defined as ratio of the size of candidate set to the size of dataset. Smaller it is, smaller the computation costs of hierarchical clustering are. The other is the error rate which is defined as ratio of number of misclassified objects, which belong to neighbors set *NS* (described in Fig. 9) and has different class with their candidate, to the size of dataset. If the error rate is high, the final clustering results are poor.

Fig. 10 shows the results of choosing candidates for CC and CBF datasets. It can be seen that the reduction rate decreases greatly from first order to second order and not distinctly from third order to fourth order. For CBF dataset, the reduction rate is below 0.1 when the maximum order is assigned with three or four, which means that the size of candidate set is less than 10% of whole dataset size. For CC dataset, the reduction rate is about 0.15. For all datasets in some order, the changes of error rates are not distinct when the size of dataset is large enough. Once the number of objects is greater than 3000, the error rate of CBF dataset is below 0.01. For CC dataset, the error rates of all experiments are below 0.005. So it can be concluded that the method of clustering candidates is effective.

5. Hierarchical clustering

5.1. Method

After the candidates are chosen, hierarchical clustering method is applied to them. A hierarchical clustering method works by grouping data objects into a tree of clusters. There are generally two types of hierarchical clustering methods: agglomerative and divisive. Agglomerative method starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all objects are in a single cluster or until certain termination conditions such as the desired number of clusters are satisfied. Divisive method does just the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster. It subdivides the cluster into smaller and smaller pieces (Han & Kamber, 2006). A tree structure called a dendrogram is commonly used to represent the process of hierarchical clustering. It shows how objects are grouped together step by step. Fig. 11 shows a dendrogram for five objects. In this paper, agglomerative method is adopted to group the candidate objects into clusters and any time series object in the neighbor set will also belong to the cluster of its neighbor candidate.

The distance between any one pair of single objects are computed by dynamic time warping defined in formula (3). During the agglomerative process, the distance between one pair of clusters, C and C', is measured based on intra-cluster distance function given by

$$d_{intra} = \frac{1}{|C||C'|} \sum_{o \in C} \sum_{o' \in C'} d_{DTW}(o, o'), \tag{9}$$

where $d_{DTW}(o, o')$ refers to dynamic time warping distance between time series objects o and o', $|C|$ means the number of objects which belong to cluster C.

5.2. Complexity

Suppose that time series dataset consists of n time series objects and each object is an ordered set of t real values. If all objects are clustered directly, the number of computation steps of DTW distance between any two objects is $t(t-1)/2$ and the computation complexity of hierarchical of clustering is $n(n-1)/2$, so the total cost is

$$C_{direct} = \frac{n(n-1)}{2} \cdot \frac{t(t-1)}{2}. \tag{10}$$

Suppose that selecting candidates can reduce the size by the reduction rate γ. Then the total computation complexity of hierarchical clustering of candidates is $\gamma n(\gamma n - 1)/2 \cdot t(t-1)/2$ and the complexity of selecting candidates is $(2f)^{1/2} n^{3/2} - n/2$ (shown in Appendix A) where f denotes the number of scans of dataset. The total complexity of our method is

$$C_{candidate} = \frac{\gamma n(\gamma n - 1)}{2} \cdot \frac{t(t-1)}{2} + [(2f)^{1/2} n^{3/2} - n/2]t. \tag{11}$$

Though generally $f \ll n$, here let $f = n/8$ and we can get

$$\begin{aligned} C_{canditate} &= \frac{\gamma n(\gamma n - 1)}{2} \cdot \frac{t(t-1)}{2} + \frac{n(n-1)}{2} t \\ &< \frac{\gamma^2 n(n-1)}{2} \cdot \frac{t(t-1)}{2} + \frac{n(n-1)}{2} t. \end{aligned} \tag{12}$$

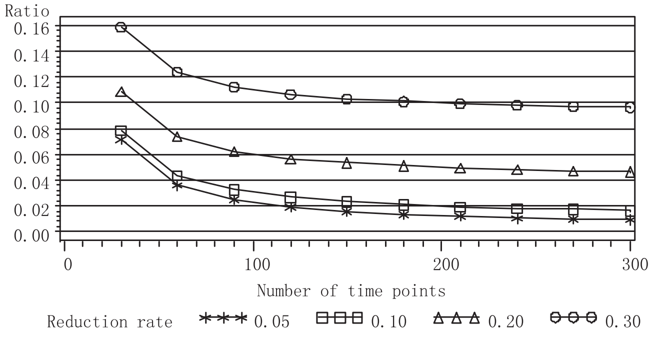

Fig. 12. The ratio of complexity of clustering candidates on clustering whole dataset directly.

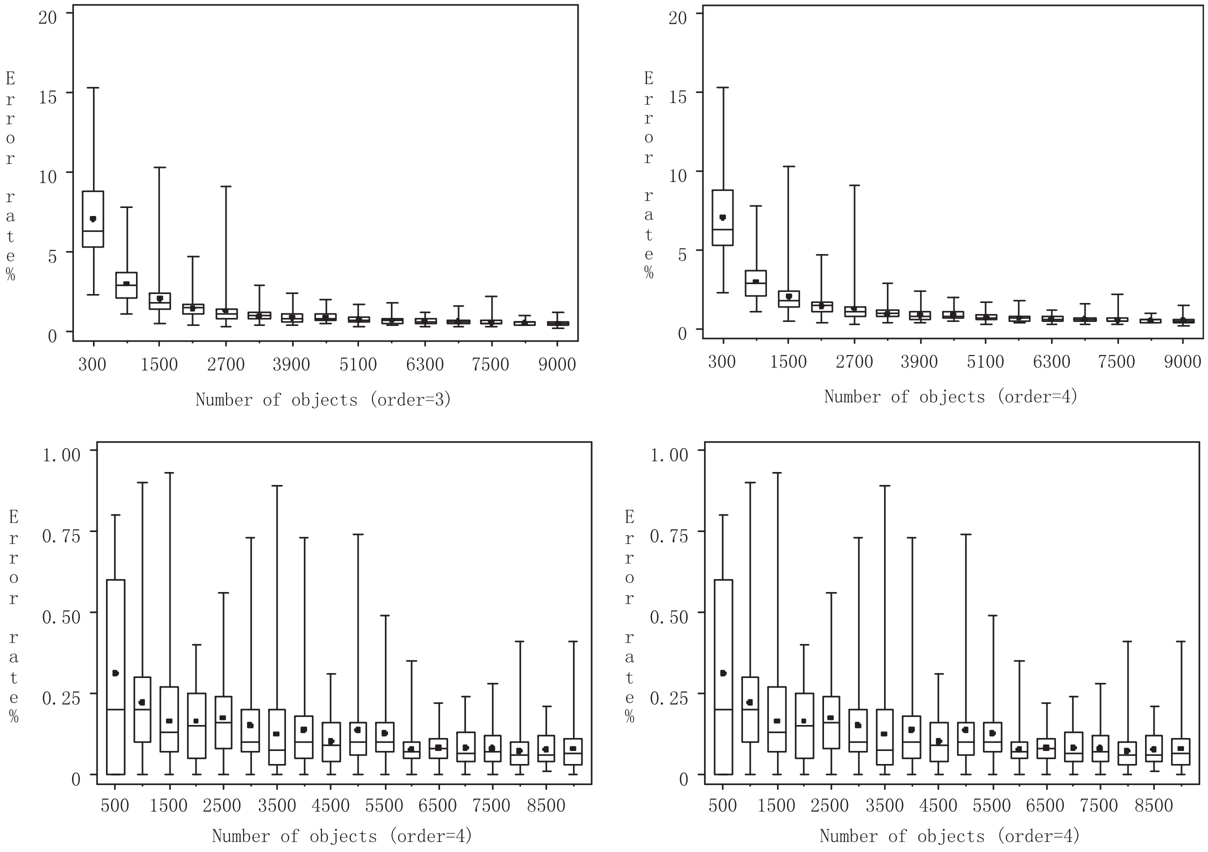

Fig. 13. The box plot of error rates of clustering results. The top row is for CBF dataset and the bottom is for CC dataset.

So the ratio of complexity of our method on complexity of clustering whole dataset directly is

$$Ratio = C_{candidate}/C_{direct} = \gamma^2 + \frac{2}{t-1}. \qquad (13)$$

Based on formula (13), we analyze the change of ratio with number of time points and reduction rate in a visual way as shown in Fig. 12. It can be seen that ratio decreases with the increase of number of points, which illustrates that our method become more effective when time series is longer. Ratio is positively correlated with reduction rate γ and decreases rapidly as reduction rate decreases.

5.3. Clustering results

The class of one cluster is defined as what the largest portion of objects in this cluster belong to. For example if in one cluster the largest portion of objects belong to class '*C*', then the cluster's class is '*C*'. If one object's class is inconsistent with its cluster's class, then it is called as misclassified object. The error rate of clustering results is defined as ratio of number of misclassified objects in all clusters to the size of dataset. In this paper the clustering effects are evaluated through error rate. The final clustering results are shown in Fig. 13. The top figures are for CBF dataset and the bottom figures are for CC dataset. The left ones are under condition that order *k* of choosing candidates equals to 3 and the right ones are that order equals to 4. For each value of parameters (number of objects and order) 50 different datasets are produced and clustered. From the results we can see that error rate decreases as number of objects decreases. In this case the error rates of different orders are not distinct. Once number of objects in CBF dataset is greater than 2700, average of error rates is less than 2%. For CC dataset all the error rates are less than 1%. The clustering effects of our algorithm are acceptable.

6. Conclusion

In this paper we propose a novel algorithm for shape based time series clustering. It can select representative candidate objects from the time series dataset based on the objects' degree in their neighbor network. The neighbor network is built based on the similarity of time series objects which is measured by triangle distance. Of the neighbor network each node represents one time series object and each link denotes neighbor relationship between nodes. Once the candidates are chosen, the task of clustering all objects in dataset is transferred to cluster the candidates, which can reduce the size of data, improve the efficiency and not reduce the effects. In clustering process, dynamic time warping distance function and hierarchical clustering algorithm are applied. Some experiments are executed on synthetic and real data. The results show that the proposed algorithm has good performance on efficiency and effectiveness.

Acknowledgement

This work is supported by Project 70901009 of National Science Foundation for Distinguished Young Scholars, BUPT_IBM 2009 Joint Project, and Youth Research and Innovation Program 2009RC1027 in Beijing University of Posts and Telecommunications.

Appendix A. The complexity of finding nearest neighbor

Suppose that there are *n* time series objects. In order to find the nearest neighbor, the distance of any pair of series must be computed, so the complexity is given by

$$n(n-1)/2.$$

The *k*-means method is applied to these objects and groups them into *k* clusters. Suppose that during the clustering, *f* (in most cases $f \gg n$) scans of dataset are carried out and each cluster has the same number of objects. The complexity is given by

$$\frac{n/k(n/k-1)}{2}k + nkf.$$

It can be easily verified that the above formula is minimized when $k = \sqrt{n/2f}$. Replacing formula with $k = \sqrt{n/2f}$, the complexity is given by

$$(2f)^{1/2}n^{3/2} - n/2.$$

References

Aach, J., & Church, G. M. (2001). Aligning gene expression time series with time warping algorithms. *Bioinformatics, 17*(6), 495–508.

Agrawal, R., Faloutsos, C., & Swami, A. (1993). Efficient similarity search in sequence databases. In *Lecture notes in computer science* (Vol. 730, pp. 69–84).

Agrawal, R., Lin, K. I., Sawhney, H. S., & Shim, K. (1995). Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceedings of the 21th international conference on very large data bases* (pp. 490–501).

Albert, R., & Barabási, A. L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics, 74*, 47–97.

Alcock, R. J., & Manolopoulos, Y. (1999). Time-series similarity queries employing a feature-based approach. In *Proceeding of the seventh Hellenic conference on informatics*.

Bagnall, A., & Janacek, G. (2005). Clustering time series with clipped data. *Machine Learning, 58*(2–3), 151–178.

Bagnall, A., Ratanamahatana, C. A., Keogh, E., Lonardi, S., & Janacek, G. (2006). A bit level representation for time series data mining with shape based similarity. *Data Mining and Knowledge Discovery, 13*(1), 11–40.

Beringer, J., & Hüllermeier, E. (2006). Online clustering of parallel data streams. *Data and Knowledge Engineering, 58*(2), 180–204.

Berndt, D. J., & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *Proceedings of AAAI workshop on knowledge discovery in databases* (pp. 229–248).

Bicego, M., Murino, V., & Figueiredo, M. A. T. (2003). Similarity-based clustering of sequences using hidden Markov models. In *Lecture Notes in Computer Science* (Vol. 2734, pp. 95–104).

Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(7), 719–725.

Boccalettia, S., Latorab, V., Morenod, Y., Chavezf, M., & Hwanga, D. U. (2006). Complex networks: structure and dynamics. *Physics Reports, 424*(4–5), 175–308.

Chandrakala, S., & Sekhar, C. C. (2008). A density based method for multivariate time series clustering in kernel feature space. In *Proceedings of IEEE international joint conference on neural networks* (pp. 1885–1890).

Chu, K. K. W., & Wong, M. H. (1999). Fast time-series searching with scaling and shifting. In: *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems* (pp. 237–248).

Chu, S., Keogh, E., Hart, D., & Pazzani, M. (2002). Iterative deepening dynamic time warping for time series. In: *Proceeding of SIAM international conference on data mining* (pp. 195–212).

Corduas, M., & Piccoloa, D. (2008). Time series clustering and classification by the autoregressive metric. *Computational Statistics and Data Analysis, 52*(4), 1860–1872.

Dahlhaus, R. (1996). On the Kullback–Leibler information divergence of locally stationary processes. *Stochastic Processes and Their Applications, 62*(1), 139–168.

Gavrilov, M., Anguelov, D., Indyk, P., & Motwani, R. (2000). Mining the stock market: which measure is best. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 487–496).

Geurts, P. (2001). Pattern extraction for time series classification. In *Lecture notes in computer science* (pp. 115–127).

Golay, X., Kollias, S., Stoll, G., Meier, D., Valavanis, A., & Boesiger, P. (1998). A new correlation-based fuzzy logic clustering algorithm for fMRI. *Magnetic Resonance in Medicine, 40*(2), 249–260.

Gonzalez, C. J. A., & Diez, J. J. R. (2000). Time series classification by boosting interval based literals. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial, 11*, 2–11.

Guan, H. S., & Jiang, Q. S. (2007). Cluster financial time series for portfolio. In *Proceedings of international conference on wavelet analysis and pattern recognition* (pp. 851–856).

Han, J. W., & Kamber, M. (2006). *Data mining: Concepts and techniques*. CA: Morgan Kaufman Publishers.

Hu, J. Y., Ray, B., & Han, L. (2006). An interweaved HMM/DTW approach to robust time series clustering. In *Proceedings of eighteenth international conference on pattern recognition* (pp. 145–148).

Janacek, G. J., Bagnall, A. J., & Powell, M. (2005). A likelihood ratio distance measure for the similarity between the Fourier transform of time series. In *Lecture notes in computer science* (Vol. 3518, pp. 737–743).

Kadous, M. W. (1999). Learning comprehensible descriptions of multivariate time series. In *Proceedings of the sixteenth international conference on machine learning* (pp. 454–463).

Kalpakis, K., Gada, D., & Puttagunta, V. (2001). Distance measures for effective clustering of ARIMA time-series. In *Proceedings of the IEEE international conference on data mining* (pp. 273–280).

Keogh, E. J., & Pazzani, M. J. (1998). An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Proceedings of the fourth international conference on knowledge discovery and data mining* (pp. 239–241).

Keogh, E., & Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. *Knowledge and Information Systems, 7*(3), 358–386.

Košmelj, K., & Batagelj, V. (1990). Cross-sectional approach for clustering time varying data. *Journal of Classification, 7*(1), 99–109.

Kumar, M., Patel, N. R., & Woo, J. (2002). Clustering seasonality patterns in the presence of errors. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 557–563).

Liao, T. W. (2005). Clustering of time series data—A survey. *Pattern Recognition, 38*(11), 1857–1874.

Lin, J., Vlachos, M., Keogh, E., & Gunopulos, D. (2004). Iterative incremental clustering of time series. In *Lecture notes in computer science* (Vol. 2992, pp. 521–522).

Man, P. W. P., & Wong, M. H. (2001). Efficient and robust feature extraction and pattern matching of time series by a lattice structure. In *Proceedings of the tenth international conference on information and knowledge management* (pp. 271–278).

Möller-Levet, C. S., Klawonn, F., Cho, K., & Wolkenhauer, O. (2003). Fuzzy clustering of short time-series and unevenly distributed sampling points. In *Lecture notes in computer science* (Vol. 2811, pp. 330–340).

Perng, C. S., Wang, H., Zhang, S. R., & Parker, D. S. (2000). Landmarks: A new model for similarity-based pattern querying in time series databases. In *Proceedings of 16th international conference on data engineering* (pp. 33–42).

Rafiei, D., & Mendelzon, A. (1997). Similarity-based queries for time series data. In *Proceedings of the ACM SIGMOD international conference on management of data* (pp. 13–25).

Ramoni, M., Sebastiani, P., & Cohen, P. R. (2000). Multivariate clustering by dynamics. In *Proceedings of the seventeenth national conference on artificial intelligence* (pp. 633–638).

Shumway, R. H. (2003). Time-frequency clustering and discriminant analysis. *Statistics and Probability Letters, 63*(3), 307–314.

Struzik, Z. R., & Siebes, A. (1999). Measuring time series' similarity through large singular features revealed with wavelet transformation. In *Proceedings of tenth international workshop on database & expert systems applications* (pp. 162–166).

Tran, D., & Wagner, M. (2002). Fuzzy c-means clustering-based speaker verification. In *Lecture notes in computer science* (Vol. 2275, pp. 363–369).

Vlachos, M., Gunopoulos, D., & Kollios, G. (2002). Discovering similar multidimensional trajectories. In *Proceedings of the 18th international conference on data engineering* (pp. 673–684).

Watts, D. J. (2004). The "New" science of networks. *Annual Review of Sociology, 30*, 243–270.

Wismüller, A., Lange, O., Dersch, D. R., Leinsinger, G. L., Hahn, K., Pütz, B., et al. (2002). Cluster analysis of biomedical image time-series. *International Journal of Computer Vision, 46*(2), 103–128.

Yeh, M. Y., Dai, B. R., & Chen, M. S. (2007). Clustering over multiple evolving streams by events and correlations. *IEEE Transactions on Knowledge and Data Engineering, 19*(10), 1349–1362.

Yi, B. K., & Faloutsos, C. (2000). Fast time sequence indexing for arbitrary Lp norms. In *Proceedings of the 26th international conference on very large data bases* (pp. 385–394).

Yin, J., & Gaber, M. M. (2008). Clustering distributed time series in sensor networks. In *Proceedings of the eighth IEEE international conference on data mining* (pp. 678–687).

Yu, F., Dong, K., Chen, F., Jiang, Y., & Zeng, W. (2007). Clustering time series with granular dynamic time warping method. In *Proceedings of IEEE international conference on granular computing* (pp. 393–393).

Zhang, X. H., Wu, J., Yang, X. C., Ou, H. Y., & Lv, T. J. (2009). A novel pattern extraction method for time series classification. *Optimization and Engineering, 10*(2), 253–271.