ISSN 1330-3651(Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20190103125702
Original scientific paper

Hierarchical Clustering of Time Series Based on Linear Information Granules

Hailan CHEN, Xuedong GAO, Yifan GUO

Abstract: Time series clustering is one of the main tasks in time series data mining. In this paper, a new time series clustering algorithm is proposed based on linear information granules. First, we improve the identification method of fluctuation points using threshold set, which represents the main trend information of the original time series. Then using fluctuation points as segmented nodes, we segment the original time series into several information granules, and linear function is used to represent the information granules. With information granulation, a granular time series consisting of several linear information granules replaces the original time series. In order to cluster time series, we then propose a linear information granules based segmented matching distance measurement (LIG_SMD) to calculate the distance between every two granular time series. In addition, hierarchical clustering method is applied based on the new distance (LIG_SMD_HC) to get clustering results. Finally, some public and real datasets about time series are experimented to examine the effectiveness of the proposed algorithm. Specifically, Euclidean distance based hierarchical clustering (ED_HC) and Dynamic Time Warping distance based hierarchical clustering (DTW_HC) are used as the compared algorithms. Our results show that LIG_SMD_HC is better than ED_HC and DTW_HC in terms of F-Measure and Accuracy.

Keywords: distance measurement; hierarchical clustering; information granules; time series

1 INTRODUCTION

Time series data arises in many areas, such as finance [1], medicine [2], environment [3] and traffic [4]. A time series is a collection of data points made chronologically. In recent years, researchers have paid more attention to time series data mining techniques, including clustering [5, 6], association rule discovery [7], classification [8, 9] and prediction [10, 11]. Such techniques tend to be used to extract meaningful knowledge and patterns of time series data. Clustering, among these techniques, is generally considered a method for data pre-processing, so it also plays an important role for the use of techniques in time series data mining.

Clustering is a process of finding natural groups in a dataset, also known as clusters. The objective is to find the most homogeneous clusters that are as distinct as possible from other clusters [12]. Any clustering technique need to rely on two concepts [13]: clustering algorithm and distance measurement. It is obvious that distance measurement is the basis of clustering algorithm, as it has a significant impact on clustering results.

The traditional time series similarity measurements are classified into five categories [14]: Euclidean distance, Dynamic Time Warping distance, Symbolic distance, Model distance and Compression distance. The most widely used distance measurement is Euclidean distance [15], but it performs unsatisfactory in terms of measuring time series trend features. In 1994, Berndt and Clifford [16] proposed Dynamic Time Warping (DTW) distance and applied it to discover patterns in time series. DTW can measure time series of different length, and it is also robust to time series offset and amplitude variation. However, the demerits of this measure include a high computational complexity, and its failure to satisfy the triangular inequality of distance. For the similarity measurement based on symbolic distance [17], time series are divided into intervals, and a short trend symbol sequence is determined by the trends judgments of intervals. Then the connectivity indexes of each trend symbol are calculated. Finally, the Mitani coefficients of the connectivity index for each trend symbol are calculated. However, this method entails the conversion of time series into the corresponding trend symbols, and its accuracy is not satisfying. Consequently, recent academic attention [18-21] has turned to information granules and granular computing for time series distance measurements.

The concept of information granules was firstly proposed by Zadeh [22]. It refers to decomposing a whole into small parts, and each part is considered as a granule. In other words, the information granules is a collection of elements, which are similar, indistinguishable, or functional [23]. In the field of time series data mining, there are various types of information granules, such as interval information granule [24], rough information granule [25] and fuzzy information granule [18]. In general, information granulation mainly includes two phases: information granule division and information granule representation. However, most researchers [26, 27] focus more on the methods of information granule representation, ignoring the study of information granule division. Previous literature usually uses the fixed time interval to segment the time series. Such operation does not take into consideration the trend features of time series, and hence the information granules cannot effectively represent the trend information. In recent years, fuzzy information granule (FIG) has been widely studied, and there are many fuzzy membership functions. However, this approach ignores the concept of time. In addition, a few researchers [21] have used variable time intervals to divide information granules. For example, l1 trend filtering [28] is introduced to divide and represent time series, which can extract the underlying linear trend and give a linear fitting of time series, but the selected parameter has a great impact on the results. Therefore, in this paper we propose a new method of information granulation including information granule division and information granule representation.

The rest of this paper is organized as follows. In Section 2, we identify the fluctuation points to segment the original time series into information granules, and use linear function to describe each information granule. In Section 3, we propose a new distance measurement—segmented matching distance based on linear information granule (LIG_SMD), which is used with hierarchical

clustering algorithm. In Section 4, we apply the proposed clustering algorithm—hierarchical clustering based on LIG_SMD (LIG_SMD_HC) to some public and real datasets, and present its performance. Finally, Section 5 summarizes our method and points out problems to be improved in future study.

2 INFORMATION GRANULATION OF TIME SERIES

Information granulation for time series mainly includes two phases: information granule division and information granule representation. Information granule division is that the time series is segmented into several small subsequences, which are then used as operation windows. For the information granule representation, the method is constructed on the divided window instead of the original window information. In this section, we propose the method of information granule division and information granule representation.

2.1 Information Granule Division

In this section, we introduce the method of fluctuation points identification [29] to divide the original time series into several information granules.

2.1.1 Identification of Extreme Points

Definition 1 (Extreme Point) Suppose there are three consecutive points of uniform sampling sequence x_{i-1}, x_i, x_{i+1}, if $(x_i - x_{i-1})(x_{i+1} - x_i) < 0$, then x_i is an Extreme Point. Where $i = 2, 3, \ldots, n - 1$.

Using the above definition to identify the extreme points of a time series, the result is shown in Fig. 1.

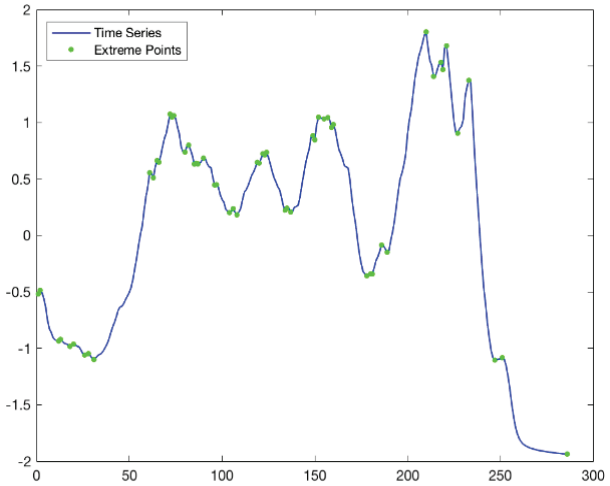

Figure 1 Identification of extreme points

2.1.2 Determination of Fluctuation Points

As shown in Fig. 1, there are many extreme points distributed on the curve segments, some of which are not obvious. Therefore, we need to set the threshold ε to select the points that its variety is larger from the extreme point. Meanwhile, we mark the attribute of the extreme points as 1 or −1.

After selecting extreme points, the remaining points that cannot represent the trend features of the original time series are called candidate fluctuation point. In order to obtain the fluctuation points, we perform the corresponding operations. The definition of fluctuation point is given as below.

Definition 2 (Fluctuation Point) Suppose there are two consecutive points of candidate fluctuation points x_{i-1}, x_i, if $|x_i - x_{i-1}| > \varepsilon$ and $Attr_{xi} * Attr_{xi-1} = -1$, then x_i is an Fluctuation Point.

Where ε is the threshold, $Attr_{xi}$ is the attribute of x_i, and $i = 2, 3, \ldots, n$.

The detailed operations are as follows. When the attributes of two consecutive points are 1, we delete the minimum one of two points. Conversely, when the attributes of two consecutive points are −1, we delete the maximum one of them. Repeating this operation until $Attr_{xi} * Attr_{xi-1} = -1$, then the remaining points are fluctuation points.

According to the above operation, fluctuation points can be identified (Fig. 3).

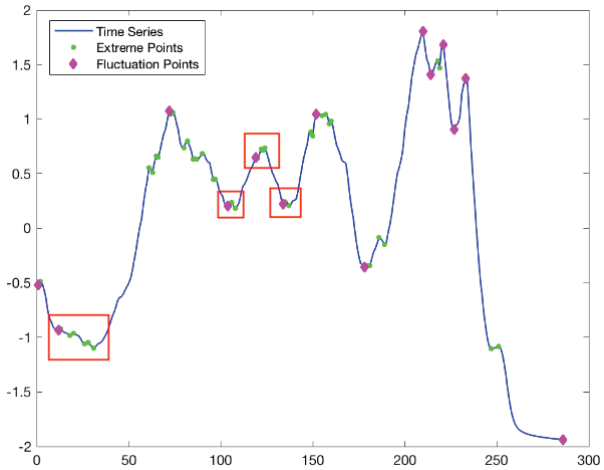

Figure 2 Fluctuation points of a time series using the single threshold

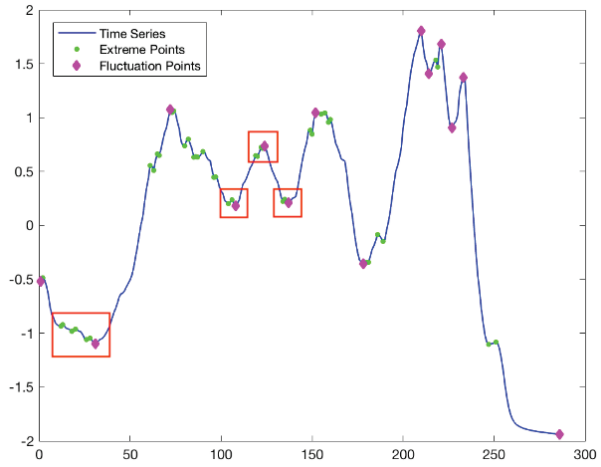

Figure 3 Fluctuation points of a time series using the threshold set

In [30] a single threshold method is used to filter the extreme points to identify the fluctuation points. Fig. 2 shows the fluctuation points that are obtained by the single threshold method. It can be seen that the method fails to identify some fluctuation points. In order to solve this problem, a method of setting a threshold set is proposed. By filtering each threshold in the threshold set, the fluctuation points are finally obtained, which are shown in

Fig. 3. It is obvious that this method is more accurate in identifying the fluctuation points.

To sum up, the process of fluctuation points identification is presented in Algorithm 1.

Algorithm 1 Fluctuation Points Identification

Input: a time series
Output: fluctuation points of the time series
Step 1: Identify the extreme points of the time series, and mark the attribute of them. Where $Attr_{xi}$=1 represent x_i is *a* maximum point, and $Attr_{xi}$=−1 represent x_i *is a* minimum point;
Step 2: Set the threshold set to filter extreme points;
Step 3: For each threshold of the threshold set, select the points from extreme points that are greater than the threshold. For $Attr_{xi}*Attr_{xi-1}$=1, when the attributes of two consecutive points are 1, deleting the minimum one of them; when the attributes of two consecutive points are −1, delete the maximum one of them; repeat this process until $Attr_{xi}*Attr_{xi-1}$=−1;
Step 4: CycleStep 3 from the first value to the last one of the threshold set.
Step 5: Finally, the remaining points are fluctuation points.

By identifying fluctuation points, the information granule division of time series is finished. In other words, an information granule is formed between two adjacent fluctuation points in the time series.

2.2 Information Granule Representation

Completing information granule division, subsequently we need to describe each information granule. The traditional information granule representation method adopts fuzzy membership function, which mainly includes triangular membership function, trapezoidal membership function, parabolic membership function, and Gaussian membership function. Nevertheless, these membership functions ignore the concept of time. Thus, in this section we use the linear function to describe the information granules, which is obviously time-dependent.

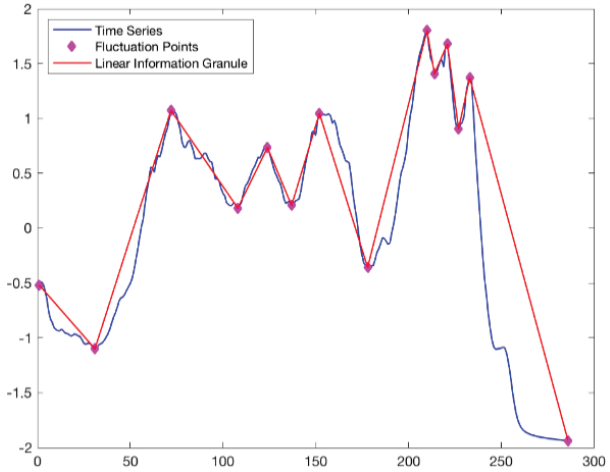

Figure 4 The information granule representation of a time series

Based on information granule division, the original series is divided into a set of subsequences. Given a subsequence $Y=\{Y_1, Y_2, \ldots, Y_T\}$, a $LIG(k, b)$ is generated through linear function $Y = kt + b$, where k, b can be calculated by two adjacent fluctuation points.

For the time series in the previous section, we use linear function to describe the divided information granules (Fig. 4).

In Fig. 4, there are 13 information granules, and each of them is represented in different linear functions (Tab. 1).

Table1 The information granular representation of a time series

Number	Time Interval	*LIG*
1	[1, 31]	$y = -0.019t - 0.499$
2	[31, 72]	$y = 0.053t - 2.743$
3	[72, 108]	$y = -0.025t + 2.857$
4	[108, 124]	$y = 0.035t - 3.548$
5	[124, 137]	$y = -0.041t + 5.773$
6	[137, 152]	$y = 0.056\ t - 7.461$
7	[152, 178]	$y = -0.054t + 9.256$
8	[178, 210]	$y = 0.068t - 12.370$
9	[210, 214]	$y = -0.099t + 22.503$
10	[214, 221]	$y = 0.039t - 6.892$
11	[221, 227]	$y = -0.129t + 30.199$
12	[227, 233]	$y = 0.078t - 16.808$
13	[233, 286]	$y = -0.062t + 15.922$

After information granulation, several linear information granules (*LIG*) form a granular time series to represent the original time series.

3 TIME SERIES CLUSTERING BASED ON LIG_SMD

The next step after information granulation is to calculate the distance of each two linear information granules. It is also the most important part for clustering. Since the time points and the time intervals of two linear information granules from any two time series are different, in this section we propose a new distance measurement—linear information granules based segmented matching distance (LIG_SMD). Afterwards, we apply hierarchical clustering method based on the new distance to give the clustering result.

3.1 Segmented Matching of *LIG*

Since the information granular division of each time series is different, when calculating the distance of two time series, firstly we need to match the linear information granules of each two time series. Fig. 5 shows the *LIG* of two time series.

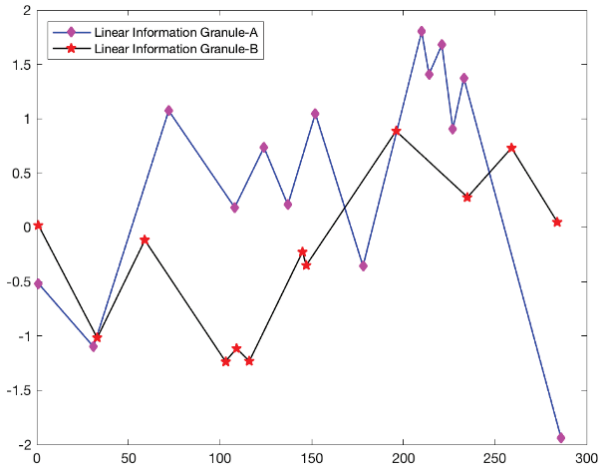

Figure 5 *LIG* of time series A and B

In order to match the linear information granules of each two time series, the following requirements must be met:

(1) The starting point and the end point of two *LIG*s from two time series should be as identical as possible;

(2) One *LIG* of a time series can be matched with multiple *LIG*s of another time series. That is, the matching relationship of *LIG* is 1:1 or 1:*n*.

The matching progress of *LIG* for any two time series is as follows (Algorithm 2).

Algorithm 2 Segmented Matching of *LIG*

Input: *LIG* of time series A, *LIG* of time series B

Output: the segmented matching *LIG* between time series A and B

Step 1: Extract all time points of *LIG* from time series A and B;

Step 2: Delete the repeated time points to obtain a chronologically non-repeating time points set *T*;

Step 3: According T to segment and match *LIG*.

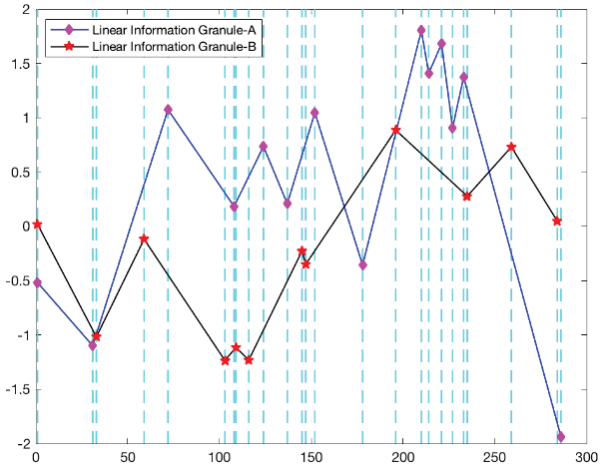

Figure 6 Segmented matching of LIG from two time series

Table 2 LIG matching of time series A and B

Number	time interval	*LIG* of Time Series A	*LIG* of Time Series B
1	[1, 31]	$y = -0.019t - 0.499$	$y = -0.032t + 0.050$
2	[31, 33]	$y = 0.053t - 2.743$	$y = -0.032t + 0.050$
3	[33, 59]	$y = 0.053t - 2.743$	$y = 0.035t - 2.160$
4	[59, 72]	$y = 0.053t - 2.743$	$y = -0.025t + 1.379$
5	[72, 103]	$y = -0.025t + 2.857$	$y = -0.025t + 1.379$
6	[103, 108]	$y = -0.025t + 2.857$	$y = 0.020t - 3.300$
7	[108, 109]	$y = 0.035t - 3.548$	$y = 0.020t - 3.300$
8	[109, 116]	$y = 0.035t - 3.548$	$y = -0.017t + 0.698$
9	[116, 124]	$y = 0.035t - 3.548$	$y = 0.035t - 5.249$
10	[124, 137]	$y = -0.041t + 5.773$	$y = 0.035t - 5.249$
11	[137, 145]	$y = 0.056t - 7.461$	$y = 0.035t - 5.249$
12	[145, 147]	$y = 0.056t - 7.461$	$y = -0.062t + 8.769$
13	[147, 152]	$y = 0.056t - 7.461$	$y = 0.025t - 4.062$
14	[152, 178]	$y = -0.054t + 9.256$	$y = 0.025t - 4.062$
15	[178, 196]	$y = 0.068t - 12.370$	$y = 0.025t - 4.062$
16	[196, 210]	$y = 0.068t - 12.370$	$y = -0.016t + 3.945$
17	[210, 214]	$y = -0.099t + 22.503$	$y = -0.016t + 3.945$
18	[214, 221]	$y = 0.039t - 6.892$	$y = -0.016t + 3.945$
19	[221, 227]	$y = -0.129t + 30.199$	$y = -0.016t + 3.945$
20	[227, 233]	$y = 0.078t - 16.808$	$y = -0.016t + 3.945$
21	[233, 235]	$y = -0.062t + 15.922$	$y = -0.016t + 3.945$
22	[235, 259]	$y = -0.062t + 15.922$	$y = 0.019t - 4.165$
23	[259, 284]	$y = -0.062t + 15.922$	$y = -0.027t + 7.801$
24	[284, 286]	$y = -0.062t + 15.922$	

According to Algorithm 2, we segment the *LIG* of two time series (Fig. 6), and then match them. As shown in Tab. 2, except for the first and last part, the whole middle parts are matched.

3.2 The New Distance Measurement Based on Segmented Matching *LIG*

The traditional distance measurements are mostly based on point-to-point calculation mode, such as Euclidean distance and Dynamic Time Warping (DTW) distance. It is apparent that this pattern of matching points is costly in time.

After matching *LIG*, the original time series is cut into subsequences. Next the essential step is to calculate the distance of each segmented matching *LIG* and sum them up to obtain the final distance. In this section, we use calculating integral to represent the distance between the corresponding segmented *LIG* of two time series, which is based on piece-to-piece calculation mode.

The segmented linear information granules are equal in length except for the first and last parts. According to the different morphological features of *LIG*, the distance is considered in four cases.

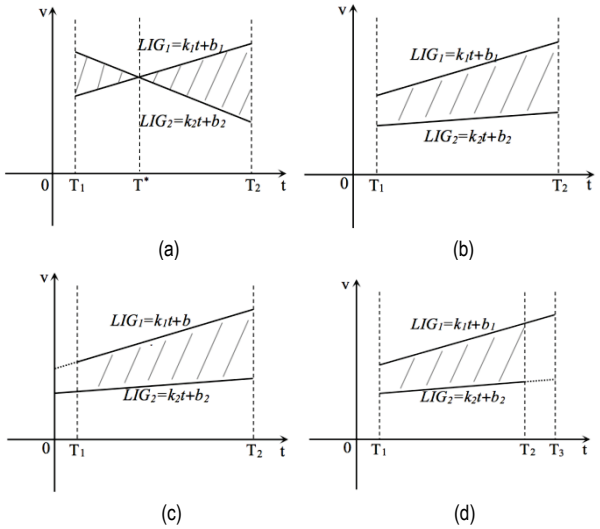

Figure 7 Four cases of distance calculation

Case 1: As shown in Fig. 7(a), the two linear information granules do not intersect in the time interval $[T_1, T_2]$, and the distance between LIG_1 and LIG_2 is:

$$D(LIG_1, LIG_2) = \int_{T_1}^{T_2} [(k_1 t + b_1) - (k_2 t + b_2)] \mathrm{d}t \tag{1}$$

Case 2: As shown in Fig. 7 (b), the two linear information granules intersect at T^* in the time interval $[T_1, T_2]$, and the distance between LIG_1 and LIG_2 is:

$$\begin{aligned} D(LIG_1, LIG_2) = &\int_{T_1}^{T^*} [(k_2 t + b_2) - (k_1 t + b_1)] \mathrm{d}t + \\ &+ \int_{T^*}^{T_2} [(k_1 t + b_1) - (k_2 t + b_2)] \mathrm{d}t \end{aligned} \tag{2}$$

where $T^* = -(b_1 - b_2) / (k_1 - k_2)$.

Case 3: As shown in Fig. 7 (c), time series A has no information granule in the time interval $[0, T_1]$, and in $[T_1,$

T_2] the linear information granule is k_1t+b_1; while for time series B, in the time interval [0, T_2] the linear information granule is k_2t+b_2, in that way, the distance between $LIG_1(T_1, T_2)$ and $LIG_2(0, T_2)$ is:

$$D(LIG_1(T_1,T_2),LIG_2(0,T_2)) = \\ = \int_0^{T_2}[(k_1t+b_1)-(k_2t+b_2)]\mathrm{d}t \quad (3)$$

Case 4: As shown in Fig. 7 (d), the linear information granule of time series A is k_1t+b_1 in the time interval [T_1, T_3]; while for time series B, the linear information granule is k_2t+b_2 in [T_1, T_2], and there is no information granule in the time interval [T_2, T_3]. Then the distance between $LIG_1(T_1, T_3)$ and $LIG_2(T_2, T_3)$ is:

$$D(LIG_1(T_1,T_3),LIG_2(T_2,T_3)) = \\ = \int_{T_1}^{T_3}[(k_1t+b_1)-(k_2t+b_2)]\mathrm{d}t \quad (4)$$

In summary, the distance between time series A and B is:

$$D(A,B) = \sum_{i=1}^{n} D(LIG_A^i, LIG_B^i) \quad (5)$$

where n is the number of segmented matching LIG, LIG_A^i is *i-LIG* of time series A, LIG_B^i is *i-LIG* of time series B.

The algorithm process of LIG_SMD is as follows:

Algorithm 3 Linear Information Granules based Segmented Matching Distance (LIG_SMD)

Input: the segmented matching LIG between time series A and B

Output: the distance between time series A and B

Step 1: Using Algorithm 2, we get the segmented matching *LIG* between time series A and B;

Step 2: Judge whether there are intersections between each pair of segmented *LIG* except for the first and last pairs; if they belong to (a), calculate the distance according to Eq. (1); if they belong to (b), find T^* and calculate the distance according to Eq. (2);

Step 3: Calculate the distance between the first pair of segmented LIG according to Eq. (3), and calculate the distance between the last pair of segmented *LIG* according to Eq. (4);

Step 4: Finally, sum up the distance of each pair to get the total distance between time series A and B.

3.3 Hierarchical Clustering Method based on The New Distance

The commonly used clustering methods include segmentation clustering method, hierarchical clustering method, density-based clustering method and grid-based clustering method [30].

Hierarchical clustering is a prototype-based clustering algorithm that attempts to divide datasets at different levels to form a tree-like clustering structure. This method can help us interpret the clustering results in a visual way by drawing a dendrogram. Moreover, we do not need to specify the number of classes before clustering.

According to Algorithm 3, the distance between every two time series is calculated. Then we apply hierarchical clustering method based on the new distance proposed in the previous section.

In conclusion, we summarize the frame of LIG_SMD based Hierarchical Clustering (LIG_SMD_HC) method as illustrated by Fig. 8.

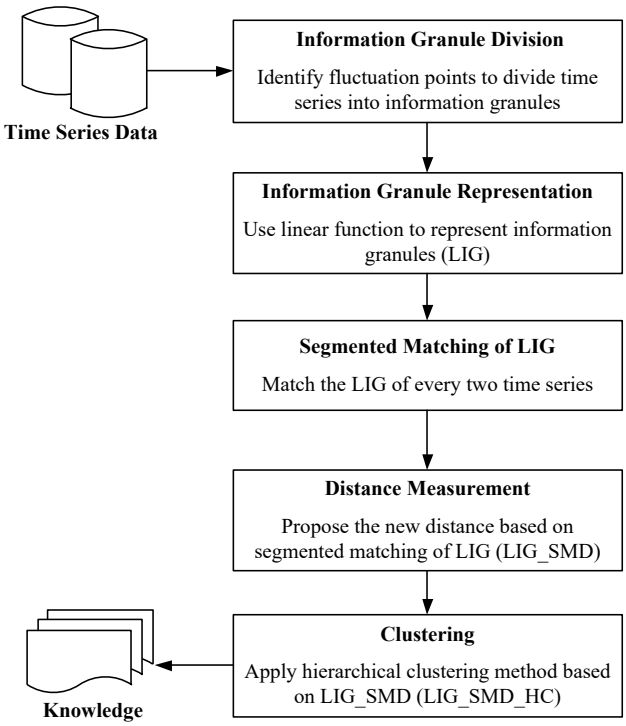

Figure 8 The framework of LIG_SMD_HC

4 EXPERIMENTAL STUDIES

In this section, we will apply the proposed clustering method (LIG_SMD_HC) on some public and real datasets to show its effectiveness. The compared methods are hierarchical clustering based on Euclidean distance (ED_HC) and hierarchical clustering based on Dynamic Time Warping distance (DTW_HC). In order to validate the clustering quality of these methods, we use F-Measure and Accuracy as evaluation metrics. Before the experiment, we first introduce the calculation method of these metrics.

4.1 Evaluation Metrics

Given any two sample points, if they belong to the same class before and after clustering, they are called positive events *T*; in contrast, if they belong to the same class before clustering, but not after clustering, they are called negative events *F*.

Then True Positive (*TP*), False Negative (*FN*), False Positive (*FP*) and True Negative (*TN*) can be defined. *TP* refers to the number of sample pairs that are in the same class before and after clustering. *FN* refers to the number of sample pairs that are in the same class before clustering but not in the same cluster after clustering. *FP* refers to the number of sample pairs that are not in the same class before

clustering but are clustered in the same cluster after clustering. *TN* refers to the number of sample pairs that are not in the same class before and after clustering.

Based on *TP*, *FN*, *FP* and *TN*, we could calculate Recall, Precision, Accuracy and *F*-Measure, and the equations of them are given as below.

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN} \tag{8}$$

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

4.2 Experiments

4.2.1 Experiment A on FaceFour Dataset

Experiment A operates on FaceFour dataset in the UCR Time Series Classification Archive [31]. 6 time series from 3 classes is selected to experiment as shown in Fig. 9. In addition, the clustering results of different methods are shown in Fig. 10.

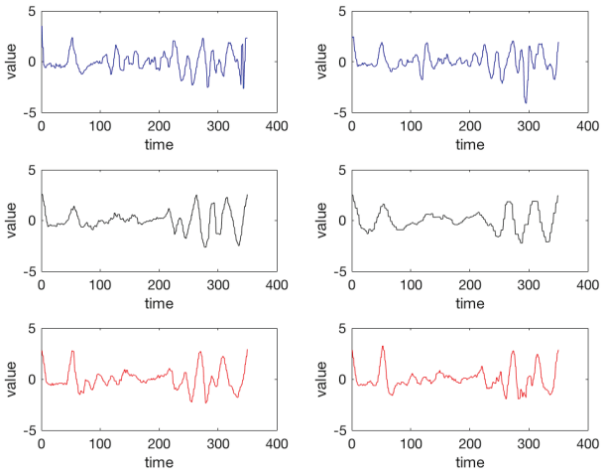

Figure 9 6 time series from 3 classes of FaceFour dataset

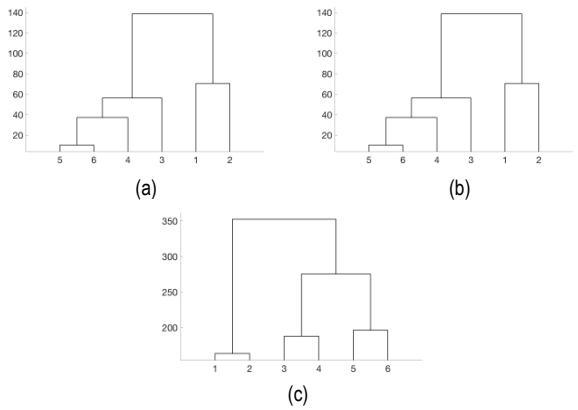

Figure 10 The clustering results of different methods (a) the clustering result of ED_HC; (b) the clustering result of DTW_HC; (c) the clustering result of LIG_SMD_HC

As shown in Fig. 10, the clustering result of ED_HC is {{1, 2}, {2, 4, 5, 6}}, which is the same as the clustering result of DTW_HC. Whereas time series are clustered as {{1, 2}, {3, 4}, {5, 6}} using LIG_SMD_HC. This result is in line with the real classes division. In Fig. 8, the trend and shape of the 6 time series across in the 3 classes are very similar, and inter-class difference is not very obvious. The above experiment proves that the proposed clustering method in this paper is effective, particularly in distinguishing time series that have similar shapes of different classes.

4.2.2 Experiment B on 8 UCR Datasets

The data of experiment B are eight datasets in the UCR Time Series Classification Archive [31]. The basic information of these datasets is shown in Tab. 3. We also use ED_HC and DTW_HC as the compared methods, and *K*-Means method is added in comparison. Moreover, we use *F*-Measure and Accuracy to validate the clustering quality of these methods. Tab. 4 presents the calculated results of 4 clustering methods.

Table 3 8 UCR datasets used in experiment B

Dataset	Samples	Length	Classes
ECG5000	500	140	5
Beef	30	470	5
Plane	105	144	7
SwedishLeaf	500	128	15
ChlorineConcentration	467	166	3
ArrowHead	36	251	3
Ham	109	431	2
synthetic_control	300	60	6

Table 4 Comparison of four clustering methods on 8 UCR datasets in terms of *F*-Measure

Dataset	LIG_SMD_HC	ED_HC	DTW_HC	*K*-Means
ECG5000	0.6933 (1)	0.6857 (3)	0.6867 (2)	0.6229 (4)
Beef	0.3667 (2)	0.3257 (3)	0.3257 (3)	0.3943 (1)
Plane	0.8834 (3)	0.8853 (2)	0.9905 (1)	0.7425 (4)
SwedishLeaf	0.9352 (1)	0.9352 (1)	0.8067 (3)	0.7644 (4)
ChlorineConcen-tration	0.5130 (1)	0.4547 (4)	0.4567 (3)	0.4659 (2)
ArrowHead	0.7175 (2)	0.7534 (1)	0.6436 (4)	0.6821 (3)
Ham	0.6445 (1)	0.6299 (2)	0.5828 (4)	0.5910 (3)
synthetic_control	0.6706 (2)	0.6556 (3)	0.8353 (1)	0.6229 (4)
Average Rank	1.625	2.375	2.625	3.125

Table 5 Comparison of four clustering methods on 8 UCR datasets in terms of Accuracy

Dataset	LIG_SMD_HC	ED_HC	DTW_HC	*K*-Means
ECG5000	0.7000 (2)	0.8000 (1)	0.7000 (2)	0.6833 (4)
Beef	0.4667 (1)	0.4667 (1)	0.4667 (1)	0.3911 (4)
Plane	0.9714 (2)	0.9619 (3)	0.9905 (1)	0.7676 (4)
SwedishLeaf	0.9333 (1)	0.9333 (1)	0.9333 (1)	0.7556 (4)
ChlorineConcen-tration	0.7400 (1)	0.4800 (4)	0.6200 (2)	0.5313 (3)
ArrowHead	0.8889 (1)	0.8889 (1)	0.8611 (2)	0.6389 (4)
Ham	0.9541 (1)	0.6972 (3)	0.7615 (2)	0.5789 (4)
synthetic_control	0.8667 (2)	0.8667 (2)	0.9333 (1)	0.6067 (4)
Average Rank	1.375	2	1.5	3.875

As shown in Tab. 4 and Tab. 5, LIG_SMD_HC achieves the highest *F*-Measure on 4 datasets and the highest Accuracy on 5 datasets respectively. To make it easier to observe the results, we sort the *F*-Measure and Accuracy values calculated by each clustering method for eight datasets, then calculate the average rank. The average ranks of *F*-Measure for these four methods are 1.625, 2.375, 2.625 and 3.125, with average rank of Accuracy of 1.375, 2, 1.5 and 3.875 respectively. Therefore, compared

to other three clustering methods, the result of LIG_SMD_HC performs better in clustering.

4.2.3 Experiment C on Real Stock Dataset

Experiment C is carried on a real stock dataset which consists of the closing prices from 2016.1.4 to 2017.9.22 for 37 stocks, totalling 423 trading days. The stock price data were downloaded from Netease Finance [32]. We normalize the dataset and fill in the missing values. The stock industry in this dataset mainly involves 8 industries, including finance, security, real estate, food and beverage, energy, electronic equipment, medical biology, information technology. The industry of stocks is used as a class label to evaluate the clustering results (Tab. 6).

Table 6 Comparison of three clustering methods on real stock dataset in terms of F-Measure and Accuracy

Evaluation Indexes	LIG_SMD_HC	ED_HC	DTW_HC
F-Measure	0.5659	0.5427	0.4718
Accuracy	0.6486	0.6216	0.5676

Since the trend features of stocks are not completely related to the industry, the *F*-Measure and Accuracy values are not very good considering the industry as the class label. Even so, the proposed method LIG_SMD_HC is still more accurate than ED_HC and DTW_HC.

5 CONCLUSION

In this study, a new time series clustering algorithm LIG_SMD_HC is proposed. We first improve the method of fluctuation points identification by using threshold set, and this method turns out to be more accurate in identifying fluctuation points. Then using fluctuation points we segment the original time series into several information granules, and linear function is applied to represent the information granules. After information granulation, several linear information granules form a granular time series on behalf of the original time series. Then, we propose a new distance measurement LIG_SMD, which consists of two steps: matching the segmented *LIG* and calculating the corresponding distance. Finally, the experiments operate on some public and real datasets, and the results show that the LIG_SMD_HC algorithm is superior in accuracy than ED_HC and DTW_HC with respect to *F*-Measure and Accuracy metrics.

Since the representation of information granules is linear function, it is more suitable for the time series with severe fluctuations. In the future, we will study the method that more accurately represents the various shapes and features of information granules for time series.

Acknowledgements

This work is supported by national science fund of China (No. 71272161).

6 REFERENCES

[1] Chen, Y. J., Chen, Y. M., Tsao, S. T., & Hsieh, S. F. (2018). A novel technical analysis-based method for stock market forecasting. *Soft Computing*, *22*(4), 1295-1312. https://doi.org/10.1007/s00500-016-2417-2

[2] Li, S. (2017). Estimating time-dependent ROC curves using data under prevalent sampling. *Statistics in Medicine*, *36*(8), 1285-1301. https://doi.org/10.1002/sim.7184

[3] Ong, B. T., Sugiura, K., & Zettsu, K. (2016). Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2.5. *Neural Computing and Applications*, *27*(6), 1553-1566. https://doi.org/10.1007/s00521-015-1955-3

[4] Chen, Y. F., Shu, L., & Wang, L. (2017). Traffic Flow Prediction with Big Data: A Deep Learning based Time Series Model.In *IEEE Conference on Computer Communications Workshops*, Atlanta, GA, USA, 1010-1011. https://doi.org/10.1109/INFCOMW.2017.8116535

[5] Paparrizos, J. & Gravano, L. (2017). Fast and Accurate Time-Series Clustering. *ACM Transactions on Database Systems*, *42*(2), 1-49. https://doi.org/10.1145/3044711

[6] Ma, R. & Angryk, R. (2017). Distance and Density Clustering for Time Series Data. In *2017 IEEE International Conference on Data Mining Workshops*, New Orleans, LA, USA, 25-32. https://doi.org/10.1109/ICDMW.2017.11

[7] Xue, R., Zhang, T., Chen, D., Le, J., & Lavassani, M. (2016). Sensor time series association rule discovery based on modified discretization method. In *2016 First IEEE International Conference on Computer Communication and the Internet*, Wuhan, China, 196-202. https://doi.org/10.1109/CCI.2016.7778907

[8] Zhang, W., Zhang, Z., Qi, D., & Liu, Y. (2014). Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors*, *14*(10), 19307-19328. https://doi.org/10.3390/s141019307

[9] Zhang, W., Zhang, Z., Chao, H. C., & Tseng, F. H. (2018). Kernel mixture model for probability density estimation in Bayesian classifiers. *Data Mining and Knowledge Discovery*, *32*(3), 675-707. https://doi.org/10.1007/s10618-018-0550-5

[10] Chandra, R., Ong, Y. S., & Goh, C. K. (2017). Co-evolutionary multi-task learning with predictive recurrence for multi-step chaotic time series prediction. *Neurocomputing*, *2017*(243), 21-34. https://doi.org/10.1016/j.neucom.2017.02.065

[11] Qiao, J., Wang, L., Yang, C., & Gu, K. (2018). Adaptive Levenberg-Marquardt Algorithm Based Echo State Network for Chaotic Time Series Prediction. *IEEE Access*, *6*(99), 10720-10732. https://doi.org/10.1109/ACCESS.2018.2810190

[12] Esling, P. & Agon, C. (2012). Time-series data mining. *ACM Computing Surveys*, *45*(1), 12. https://doi.org/10.1145/2379776.2379788

[13] Hammouda, K. M. & Kamel, M. S. (2004). Efficient phrase-based document indexing for Web document clustering. *IEEE Transactions on Knowledge and Data Engineering*, *16*(10), 1279-1296. https://doi.org/10.1109/TKDE.2004.58

[14] Li, H. & Guo, C. (2013). Survey of feature representations and similarity measurements in time series data mining. *Application Research of Computers*, *30*(5), 1285-1291. https://doi.org/10.3969/j.issn.1001-3695.2013.05.002

[15] Chen, H., Liu, C., & Sun, B. (2017). Survey on similarity measurement of time series data mining. *Control and Decision*, *32*(1), 1-11. https://doi.org/10.13195/j.kzyjc.2016.0462

[16] Berndt, D. J. & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA, 359-370.

[17] Xiao, R. & Liu, G. (2014). Research on trend-based time series similarity measure and cluster.*Application Research of Computers*, *31*(9), 2600-2605. https://doi.org/10.3969/j.issn.1001-3695.2014.09.009

[18] Lu, W., Pedrycz, W., Liu, X., Yang, J., & Li, P. (2014). The modeling of time series based on fuzzy information granules. *Expert Systems with Applications*, *41*(8), 3799-3808.

https://doi.org/10.1016/j.eswa.2013.12.005
[19] Al-Hmouz, R., Pedrycz, W., & Balamash, A. (2015). Description and prediction of time series: a general framework of granular computing. *Expert Systems with Applications*, *42*(10), 4830-4839. https://doi.org/10.1016/j.eswa.2015.01.060
[20] Froelich, W. & Pedrycz, W. (2017). Fuzzy cognitive maps in the modeling of granular time series. *Knowledge-Based Systems*, *2017*(115), 110-122. https://doi.org/10.1016/j.knosys.2016.10.017
[21] Duan, L., Yu, F., Pedrycz, W., Wang, X., & Yang, X. (2018). Time-series clustering based on linear fuzzy information granules. *Applied Soft Computing*, *2018*(73), 1053-1067. https://doi.org/j.asoc.2018.09.032
[22] Zadeh, L. A. (1996). Fuzzy Sets and Information Granularity. *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems*, 433-448. https://doi.org/10.1142/9789814261302_0022
[23] Pedrycz, W., Succi, G., Sillitti, A., & Iljazi, J. (2015). Data description: A general framework of information granules. *Knowledge-Based Systems*, *2015*(80), 98-108. https://doi.org/10.1016/j.knosys.2014.12.030
[24] Lu, W., Chen, X., Pedrycz, W., Liu, X., & Yang, J. (2015). Using interval information granules to improve forecasting in fuzzy time series. *International Journal of Approximate Reasoning*, *57*(1), 1-18. https://doi.org/10.1016/j.ijar.2014.11.002
[25] Kaneiwa, K. & Kudo, Y. (2012). A sequential pattern mining algorithm using rough set theory. *International Journal of Approximate Reasoning*, *52*(6), 881-893. https://doi.org/10.1016/j.ijar.2011.03.002
[26] Zhu, X., Pedrycz, W., & Li, Z. (2017). Granular Data Description: Designing Ellipsoidal Information Granules. *IEEE Transactions on Cybernetics*, *47*(12), 4475-4484. https://doi.org/10.1109/tcyb.2016.2612226
[27] Yang, X., Yu, F., & Pedrycz, W. (2017). Long-term forecasting of time series based on linear fuzzy information granules and fuzzy inference system. *International Journal of Approximate Reasoning*, *2017*(81), 1-27. https://doi.org/10.1016/j.ijar.2016.10.010
[28] Kim, S., Koh, K., Boyd, S., Gorinevsky, D., & Review, T.S. (2009). l1 trend filtering. *Siam Review*, *51*(2), 339-360. https://doi.org/10.1137/070690274
[29] Gao, X. & Chen, H. (2016). Frequent patterns discovery and analysis granularity recognition for Shanghai Composite Index sequence. In *2016 International Conference on Logistics, Informatics and Service Sciences*, Sydney, NSW, Australia, 1242-1248. https://doi.org/10.1109/LISS.2016.7854511
[30] Wu, S. & Gao, X. (2003). *Data Warehouse and Data Mining*. Beijing, China: Metallurgical Industry Press.
[31] http://www.cs.ucr.edu/~eamonn/time_series_data/.
[32] http://money.163.com/stock/.

Contact information:

Hailan CHEN, PhD Candidate
Donlinks School of Economics and Management,
University of Science and Technology Beijing,
No. 30 Xueyuan Road, Haidian District, Beijing, China
chl_hld@163.com

Xuedong GAO, Full Professor
(Corresponding author)
Donlinks School of Economics and Management,
University of Science and Technology Beijing,
No. 30 Xueyuan Road, Haidian District, Beijing, China
gaoxuedong@manage.ustb.edu.cn

Yifan GUO, M.S. Candidate
School of Management,
China University of Mining and Technology,
Ding No. 11 Xueyuan Road, Haidian District, Beijing, China
guo_onlyIdol@163.com