Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 421 (2021) 62–76

FUZZY sets and systems

www.elsevier.com/locate/fss

Incremental fuzzy clustering of time series ☆

Ling Wang [a,b,*], Peipei Xu [a,b], Qian Ma [a,b]

[a] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China
[b] Key Laboratory of Knowledge Automation for Industrial Processes, Ministry of Education, University of Science and Technology Beijing, Beijing 100083, China

Received 1 February 2020; received in revised form 1 November 2020; accepted 8 January 2021
Available online 12 January 2021

Abstract

Clustering is one of the most popular data mining methods for analyzing the time series, not only due to its exploratory power, but also because it is often a preprocessing step or subroutine for other techniques. In this paper, an incremental fuzzy clustering algorithm of time series (IFCTS) is proposed, in which the clustering is divided into two stages: offline and online clustering. During the offline clustering process, the fuzzy clustering validity evaluation index is introduced to FCM to automatically obtain the optimal number of initial clusters for off-line data. Then, in the online clustering process, IFCTS algorithm can dynamically update the existing clusters in the incremental data snapshot, distinguish outliers and control the creation of new clusters adaptively by combining with the previous clusters. The experimental results show that the proposed algorithm has good clustering accuracy and efficiency for both equal-length and unequal-length time series.

Keywords: Time series; Fuzzy clustering; Incremental learning; Outliers

1. Introduction

The time series is one of the common data types in clustering problems. In view of its high-dimensional feature, time series can be viewed as a complex data object with a large number of time-varying data points, which make it more difficult to perform cluster analysis in a standard way [1]. Currently, the central problem on the time series clustering mainly focuses on the similarity measurement of sequences, which includes the Euclidean distance [2], Dynamic Time Warping (DTW) distance [3], Poisson correlation coefficient [4], and longest common subsequence [5], quantile auto-covariance functions (QAF) [6], the exponential transformation of dissimilarity measures [7], etc., which can be applied to compute the similarity between two time series or subsequences. Based on the appropriate similarity measurement method, the different clustering algorithms can effectively grasp the trend of time series such

☆ Fully documented templates are available in the elsarticle package on CTAN.

* Corresponding author at: School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China.
E-mail address: lingwang@ustb.edu.cn (L. Wang).

https://doi.org/10.1016/j.fss.2021.01.002
0165-0114/

as Hierarchical clustering [8], partitional clustering [9], model-based clustering [10], feature-based clustering [11], etc.

As a common clustering algorithm, fuzzy clustering algorithm has also made a lot of progress in the field of time series data. Different from the hard clustering algorithms that divide each data object directly into a cluster, fuzzy clustering algorithms enable data objects to be divided into multiple clusters with different memberships, which is more suitable. Several variants of the previous fuzzy clustering models have been proposed in [12]. In [13], based on appropriate measures of dissimilarity between time trajectories, Fuzzy C-Means objective function is adopted to distinguish the cross-sectional and longitudinal aspects of the trajectories. In [14], autocorrelation is adopted to transform time series data into the feature space, and then Fuzzy C-Means (FCM) algorithm is realized to obtain the clusters by the similarity measures with the Euclidean distance. In [15], based on the FCM algorithm, the longest common subsequence is introduced as the similarity measurement between time series. In [16], the FCM clustering algorithm, fuzzy centroids clustering algorithm and hybrid method based on DTW distance are respectively adopted to cluster time series with unequal length. When the interest is in capturing the differences concerning the variational pattern of the time series, a velocity based FCM clustering model is proposed in [17], in which the instantaneous and slope (velocity) features of the time series are simultaneously taken into account. In [7], to cope with the complexity of the features of each multivariate time series and the associated assignment uncertainty, Fuzzy C-Medoids clustering algorithm is adopted, in which the exponential transformation of dissimilarity measures is used to neutralize the effect of possible outliers. However, most fuzzy clustering algorithms for mining time series focus on the static data sets.

In real-world applications, time series data usually remain a dynamic state. With the time series increasing, some new implicit information may emerge and the existing clusters may become invalid by static fuzzy clustering algorithm. So, it is necessary to develop an efficient incremental fuzzy clustering algorithm to update the existed clusters for a dynamic time series database. Nowadays the existing incremental fuzzy clustering algorithms are usually extended to cluster the equal-length time series data[16-25]. For example, in [18] and [19], the traditional FCM and improved FCM algorithms are respectively adopted to conduct incremental clustering for text data and network log data. In [20], the differential evolution and random forest are incrementally incorporated into the fuzzy clustering to deal with the categorical data. In [21], a new fuzzy clustering objective function is proposed with data snapshot division to incrementally cluster the image data and malware data. Although these fuzzy clustering algorithms can incrementally perform well, they all need prior knowledge to predefine the number of clusters and remain the number of clusters unchanged throughout the clustering process, which limit their application scope. In order to solve this problem, an Incremental Fuzzy C-Means (IFCM) algorithm is proposed in [22], it firstly gets the initial clustering centers for the offline data, and then performs fuzzy clustering incrementally for the online data snapshots. However, in order to identify the outlier data objects and create new clusters, two control parameters need to be set manually, and all the generated clusters will participate in the subsequent clustering, which affects the execution efficiency and self-adaptability. In [23], a Dynamic Data Mining Based on Fuzzy Clustering (DDMFC) algorithm is proposed to deal with newly arrived data incrementally, which could not only update the existing clusters, but also create new clusters according to the current data and remove some empty clusters. However, three parameters need to be predefined manually to control the identification of outlier data objects, the creation of new clusters and the removal of empty clusters. The above incremental fuzzy clustering algorithms are also applicable to the equal-length time series, but they are incapable of dealing with the unequal-length time series. In order to deal with the unequal-length time series, the longest common subsequence is used to measure the similarity of time series in [24], the number of initial clusters are obtained from offline data by means of hierarchical clustering, and the online data is incrementally clustered with fuzzy cluster algorithm, but it still need to set parameters to identify the outlier data objects. In order to make the algorithm more robust, some research adopted the idea of adaptability [25] to deal with the specific problems. In [26], based on the new fuzzy neighborhood function, an adaptive clustering algorithm for finding the clusters with arbitrary shapes, densities, distributions and quantities is proposed. In [27], a local adaptive multi-core clustering algorithm based on the shared nearest neighbors is proposed, which can adjust the clusters or parameters automatically according to the current data and obtain the best results.

In view of the above algorithms, this paper proposes an incremental fuzzy clustering of the time series (IFCTS) algorithm. The fuzzy clustering validity evaluation index is introduced to FCM to automatically obtain the optimal number of initial clusters for off-line data. On this basis, IFCTS algorithm can dynamically update the existing clusters according to the incremental time series data snapshot and automatically identify outlier data objects, add new clusters and remove some empty clusters. In the incremental clustering process, except for the threshold of removed empty

cluster, the cluster needn't to predefine other parameters. In addition, IFCTS algorithm can distinguish outlier data objects and control the creation of new clusters adaptively by combining with the previous clusters. In the experiment, IFCTS algorithm is verified with multiple time series data sets and also applied to the symbolization of dynamic time series. The results show that the IFCTS algorithm can get good cluster performance for both equal-length and unequal-length time series.

The rest of the paper is organized as follows: Section 2 introduces fuzzy clustering of time series; The incremental fuzzy clustering of time series is detailed in Section 3; Section 4 presents the performance of the proposed algorithm on multiple time series datasets; Some conclusions are covered in Section 5.

2. Fuzzy clustering of time series

2.1. FCM clustering

In this time series framework, FCM algorithm is adopted to perform the time series subsequence clustering. After the subsequences are clustered, the symbolic expression of the long time series is labeled which can not only find the patterns in the long time series but also realize the dimensionality reduction of the time series. Because the subsequences may be unequal lengths, the similarity between unequal lengths subsequences cannot be measured with Euclidean distance. But, the Dynamic Time Warping (DTW) distance [3] is a desirable choice when grouping time series subsequences based on their shape information is of interest (shape-based clustering), which determines an optimal match between any two time series subsequences with different lengths by using stretching or compressing segments of temporal data.

Here, let $z_i = \{z_{i1}, z_{i2}, ..., z_{ip}, \ldots, z_{iT_i}\}$ is a time series subsequence with a length of T_i, and $v_j = \{v_{j1}, v_{j2}, ..., v_{jq}, \ldots, v_{jT_j}\}$ is the cluster center (also a time series) with a length of T_j ($T_i \neq T_j$).

The objective function of FCM algorithm is:

$$J(U, V) = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{w} dis_{DTW}(z_i, v_j)^2 \tag{1}$$

Where U is the fuzzy partition matrix; u_{ij} should satisfy $\sum_{j=1}^{C} u_{ij} = 1$ ($u_{ij} \in [0, 1]$); w is the fuzzy weighted index, and its value is 2; $dis_{DTW}(z_i, v_j)$ represents the DTW distance between the ith time series subsequence z_i and the jth cluster center v_j, which need to be iteratively calculated as follows [16]:

$$dis_{DTW}(z_i, v_j) = \gamma(z_{iT_i}, v_{jT_j}) \tag{2}$$

$$\begin{aligned}\gamma(z_{iT_i}, v_{jT_j}) = d(z_{iT_i}, v_{jT_j}) + \min\{&\gamma(z_{i(T_i-1)}, v_{jT_j}),\\ &\gamma(z_{iT_i}, v_{j(T_j-1)}), \gamma(z_{i(T_i-1)}, v_{j(T_j-1)})\}\end{aligned} \tag{3}$$

$$d(z_{iT_i}, v_{jT_j}) = \left\| z_{iT_i} - v_{jT_j} \right\|_2 \tag{4}$$

To minimize the objective function $J(U, V)$, the cluster center v_j and the membership matrix U need to be computed according to the following iterative formula [28]:

$$u_{ij} = \begin{cases} \left(\sum_{k=1}^{c} \left(\frac{dis(z_i, v_j)}{dis(z_i, v_k)}\right)^{\frac{2}{w-1}}\right)^{-1} & \text{if } \operatorname{dis}(z_i, v_j) \neq 0 \\ 1 & \text{if } \operatorname{dis}(z_i, v_j) = 0 \\ 0 & \text{if } \exists j \neq i \operatorname{dis}(z_i, v_j) = 0 \end{cases} \tag{5}$$

In addition, the cluster centers can not be calculated with DTW distance function in (2). To solve this problem, a global average strategy with DBA (DTW Barycenter Averaging) [16] was introduced to FCM, which could accurately and effectively obtain the centers of cluster on the basis of DTW distance whether equal length or unequal length time series.

Fig. 1 shows an example of DTW path between the cluster center v_j and each time series subsequence $z_i (i = 1, 2, ..., N)$. Let $\alpha_{jq}(i)$ represents the qth unit where z_i and v_{jq} intersected by traversing along DTW path, and $Sum_{jq}(i)$ represents the sum of elements corresponding to $\alpha_{jq}(i)$.

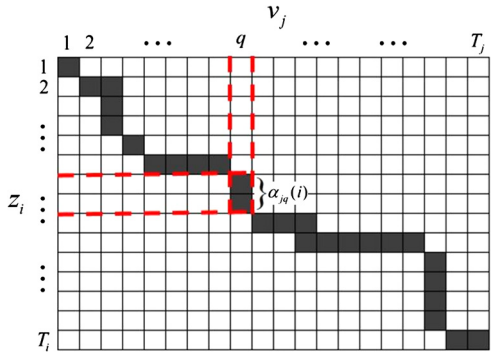

Fig. 1. An Example of DTW paths for cluster centers and time series.

The cluster center v_{jq} is updated as follows:

$$v_{jq} = \frac{\sum_{i=1}^{N} u_{ij}^{w} Sum_{jq}(i)}{\sum_{i=1}^{N} \alpha_{jq}(i) u_{ij}^{w}}, q = 1, 2, ..., T_j \tag{6}$$

2.2. Cluster validity

In order to obtain the appropriate clusters, the Extension of the Xie-Beni index [29] is combined with FCM to obtain the optimal number of clusters, which attempts to maximize intra-class similarity and inter-class differences. In this sense, the fuzzy clustering validity index is denoted as:

$$V_{XB} = \frac{\sum_{i=1}^{N_0} \sum_{j=1}^{C} u_{ij}^{w} dis(z_i, v_j)^2 + \frac{1}{C} \sum_{j=1}^{C} dis(v_j, \bar{v})^2}{\min\limits_{j \neq k} dis(v_j, v_k)^2} \tag{7}$$

$$\bar{v} = \frac{1}{N_0} \sum_{i=1}^{N_0} z_i \tag{8}$$

where N_0 is the total number of time series subsequences in the data snapshot. The first term of the numerator in (7) measures the intra-cluster similarity, that is, how compact every cluster is. The second term of numerator in (7) is a punishing function that avoid the index monotonically decreasing when the number of cluster C tends to N_0. The denominator in (7) denotes the inter-cluster separation, which measures the minimum distance between the cluster centers. The goal of this index is to find the optimal fuzzy clusters with the smallest value of V_{XB}.

3. Incremental fuzzy clustering of time series (IFCTS)

To enhance the effectiveness of the incremental fuzzy clustering of time series, reduce the complexity and find the evolving relationship of each cluster, the framework of the IFCTS is proposed. Firstly, the time series data is partitioned by data snapshots; the offline fuzzy clustering is used in the first data snapshot, the online fuzzy clustering is used in the data snapshots that comes in turn. In the off-line fuzzy clustering process, FCM is used to perform clustering to obtain the initial clusters by combining with the cluster validity index. According to equal-length or unequal-length of time series, Euclidean distance or DTW distance is respectively adopted as the distance function to metric the similarity in the objective function. In the online fuzzy clustering process, the initial clusters with the fuzzy radius and the merge distance that obtained from the first data snapshot will be inherited to participate in the clustering of the next incremental data snapshots. Based on the inherited cluster information, new clusters will be determined to add or not. If all time series subsequences in the data snapshot have a small membership to some cluster where no new sequence is clustered according to the principle of maximum membership, the cluster is initially recognized as an empty cluster by comparing with the removal threshold of empty cluster and removed. In the course of incremental learning, there is no other parameters that need to be set except for the removal threshold of empty cluster.

3.1. Related concepts

Definition 1. Incremental data snapshot: The time series data is partitioned by data snapshots. With the time series increasing dynamically, the first data snapshot data is viewed as offline data to provide the initial cluster information, denoted as $DS[0]$. The new data snapshot that comes in turn is viewed as on-line data to realize incremental learning, called incremental data snapshots, represented by $DS[1], DS[2], \ldots, DS[h]$. The size of the data snapshot is determined according to the rate of data arrival, which can be set as days, weeks, months, years and other time units.

Definition 2. Empty cluster: Let N_h is the total number of time series in a given data snapshot $DS[h](h = 1, 2, ...)$, and C_h is the numbers of the clusters obtained after incremental learning. If all time series in the data snapshot have a small membership to some cluster where no new sequence is clustered according to the principle of maximum membership, the cluster is initially recognized as an empty cluster. The Empty cluster identifier $E^{(j)}(j \in 1, ..., C_h)$ is defined as the number of times that the jth cluster is continuously recognized as an empty cluster in the subsequent data snapshot, then $E^{(j)}$ is calculated as:

$$E^{(j)} = \begin{cases} E^{(j)} + 1, \text{if the } j\text{th cluster is empty cluster.} \\ 0, \text{others} \end{cases} \tag{9}$$

During incremental learning, the empty clusters can be determined to remove with (9). Here, ε is the threshold for removing empty cluster. If $E^{(j)} > \varepsilon$, the jth cluster will be removed.

Definition 3. Maximum member distance: According to the maximum membership, each time series subsequence $z_i(i = 1, 2, ..., N_h)$ in the data snapshot $DS[h]$ is clustered into the corresponding cluster, $count^{(j)}$ represents the number of sequences contained in the jth cluster. For a cluster, if $count^{(j)} \geq 2$, the maximum member distance between sequences is defined as:

$$\begin{aligned} &D^{(j)} = \max\{dis(z_g^{(j)}, z_l^{(j)})\}, \\ &\text{s.t. } g, l = 1, ..., count^{(j)},\ g \neq l, count^{(j)} \geqslant 2 \end{aligned} \tag{10}$$

where $z_g^{(j)}$ and $z_l^{(j)}$ are respectively two different sequences in the jth cluster.

Definition 4. Merging distance: The maximum value for all maximum member distance of all clusters is called the merge distance:

$$D_h\_\max = \max_{j=1,...,C_h} \{D^{(j)}\} \tag{11}$$

Definition 5. Fuzzy radius: In view of [30], the membership was introduced to construct the fuzzy radius of the cluster. For data snapshot $DS[h]$, C_h clusters are obtained after clustering, the fuzzy radius of the $jth(j = 1, ..., C_h)$ cluster is calculated as:

$$r_h^{(j)} = \max_{i=1,...,N_h} u_{ij} dis(z_i, v_j) \tag{12}$$

where u_{ij} is the membership of time series subsequence z_i in the jth cluster, and $dis(z_i, v_j)$ is the distance between z_i and the cluster center v_j.

3.2. Incremental fuzzy clustering of time series

To improve the efficiency of incremental fuzzy clustering of time series, our proposed framework included four stages: First, FCM is performed on the offline time series data snapshot $DS[0]$, the initial clusters are obtained by combining with the empty cluster identifier $E^{(j)} = 0(j = 1, 2, ..., C_0)$, the merging distance $D_0\_\max$ and the fuzzy radius $r_0^{(j)}(j = 1, 2, ..., C_0)$ of each cluster. Next, not only the distance between the time series subsequences in the subsequent data snapshot and the existing cluster centers but also the distance between the existing cluster centers will be calculated. Then, the outlier time series subsequences will be identified whether the weighted distance between

the time series subsequence z_i and all the existing cluster centers are greater than the fuzzy radius of each cluster or the distances between the time series subsequence z_i and all the existing cluster centers are greater than half of the minimum distance between any two cluster centers. Finally, we would create new cluster or update the existing clusters by judging whether the outlier time series subsequences exist. The detailed IFCTS is described in Algorithm 1.

Algorithm 1 IFCTS.

Input:Incremental data snapshot $DS[h](h = 1, 2, ...)$, the cluster centers $v_j(j = 1, 2, ..., C_{h-1})$, Fuzzy radius $r_{h-1}^{(j)}$, Empty cluster identification $E^{(j)}$, Merge distance $D_{h-1}\_\max$.

Output: The clustering results of the time series in $DS[h]$.

Step 1. Initiate the number of clusters $C_h = C_{h-1}$,Fuzzy radius $r_h^{(j)} = r_{h-1}^{(j)}(j = 1, 2, ..., C_{h-1})$;

Step 2. Calculate the distance $dis(z_i, v_j)$ between the time series subsequence z_i in $DS[h]$ and the center v_j of the existing cluster, and the membership of the time series subsequence z_i belong to the existing cluster to get the weighted distance $weighted\_dis_{ij} = u_{ij} dis(z_i, v_j)$;

Step 3. Calculate the distance between the centers of the clusters $dis(v_j, v_k)(j, k = 1, 2, ..., C_h; j \neq k)$;

Step 4. For $\forall z_i \in DS[h]$

If the time series subsequence z_i satisfied the following two conditions:

$$\forall\ j = 1, 2, ..., C_h, \quad weighted\_dis_{ij} > r_h^{(j)} \text{ and } dis(z_i, v_j) > \frac{1}{2} \min_{\substack{k=1,...,C_h \\ k \neq j}} dis(v_j, v_k),$$

then z_i can be identified as an outlier time series subsequence away from the existing cluster, go to step 5

else

go to step 6

Step 5. If there is only one outlier time series subsequence then

a new cluster will be created and the time series subsequences will be looked as the new cluster center, and then update the number of the existing cluster C_h, go to step 7

else

multiple new clusters for each time series subsequences will be created, go to step 6

Step 6. Gradually merge two clusters that are closest to each other and the distance between them is less than $D_{h-1}\_\max$, and then update the number of the existing cluster C_h, go to step 7

Step 7. For $j = 1, 2, \ldots, C_h$

Adopt FCM algorithm to cluster time series in snapshots $DB[h]$ and update $E^{(j)}$ iteratively, which would not end until $\max_{ij}\{|u_{ij}(p+1) - u_{ij}(p)|\} < 0.01$ (p represents the number of iterations)

If $E^{(j)} > \varepsilon$ with (9)

Then remove the j-th empty cluster, update the number of clusters C_h, calculate the fuzzy radius $r_h^{(j)}$ of each cluster and the merge distance $D_h\_\max$

Step 8 Output the clustering results with $v_j(j = 1, 2, ..., C_h)$, $r_h^{(j)}$, $E^{(j)}$, $D_h\_\max$

3.3. Complexity analysis

For each data snapshot $DS[h](h = 1, 2, ...)$, according to the equal-length or unequal-length of time series, the time complexity of IFCTS is respectively taken into account with two cases. The one is the time series subsequences with equal length T. Assume that C_{h-1} is the number of initial clusters in the data snapshot, C_h is the total number of clusters after the new clusters are created. The complexity of calculating the Euclidean distance between two equal-length sequences is $O(T)$, and the complexity of calculating the center of a cluster based on Euclidean distance is $O(N_h)$.The complexity of calculating the weighted distance based on the distance matrix and membership matrix is $O(C_{h-1} \times N_h \times (T + 1))$. The complexity of calculating the distance between the cluster centers is $O(C_{h-1} \times N_h \times (T + 1) + {C_{h-1}}^2 \times T)$. The complexity of identifying the outlier time series subsequences in data snapshot is $O({C_{h-1}}^2 + C_{h-1} \times N_h)$. Assume the number of outlier time series subsequences found is denoted as $O_h (0 \leq O_h < N_h)$. The complexity of creating the new clusters for the outlier time series subsequences is $O((O_h - C_h + C_{h-1}) \times {O_h}^2 \times (T + 1))$. Let the number of iterations for clustering is p. The complexity of clustering N_h time series subsequences with C_h cluster centers iteratively is $O(p \times C_h \times N_h \times (T + 2))$. All the clusters and the empty cluster identifier need to be updated for the subsequent data snapshot in the incremental learning process, so the complexity is $O(C_h \times (N_h + 3))$. In conclusion, it can be seen that the total complexity of the algorithm depend on

Table 1
Description of data set.

Data set	The number of clusters	Time series length	Training set size	Test set size
Italy Power Demand (IPD)	2	24	67	1029
Synthetic Control (SC)	6	60	300	300
Phalanges Outlines Correct (POC)	2	80	1800	858
CBF	3	128	30	900
ECG5000	5	140	500	4500
Chlorine Concentration (CC)	3	166	467	3840
Medical Images (MI)	10	99	381	760
Face All (FA)	14	131	560	1690

the number of clusters, the number of outlier sequences, the length and the number of time series, which is simplified as $O(((T+2)(p+1)+1)C_h \times N_h + (T+1){C_h}^2)$. The other is the time series with unequal length in the data snapshot. Assume the average length of unequal length is T, the complexity of calculating the DTW distance between two unequal-length sequences is $O(T^2)$, and the complexity of calculating the center of a cluster based on DTW distance is $O(N_h \times (T^2+T))$. Expect this, it is similar to realize the incremental fuzzy clustering. So the total complexity of the algorithm with unequal-length time series subsequences is simplified as $O(((T^2+2)(p+1)+1)C_h \times N_h \times T^2 \times (T^2+T) + (T^2+1){C_h}^2)$.

4. Experimental studies

A series of experiments have been conducted to test the proposed IFCTS algorithm. In order to verify the accuracy and efficiency of the IFCTS, two other incremental clustering algorithms are compared on eight time series classification data sets that each has equal-length time series. A real dynamic time series data set with unequal length is applied to verify the cluster performance and evolving process of IFCTS. The fuzzy weighted index is set as 2, the maximum number of iterations is 50. All experiments were performed with Python3.5 running on a 2.40 GHz processor and 4.0 GB memory. To evaluate the clustering performance, the accuracy criterion is adopted as (13).

$$Acc = \frac{\sum_{j=1}^{C} s_j}{N} \tag{13}$$

where s_j is the number of patterns that have the correct clustering results corresponding to the correct category labels j. C is the number of categories included in the original data set, and N is the number of time series in the data set.

4.1. Performance evaluation with equal-length time series

To evaluate the performance of IFCTS algorithm, eight time series datasets from UCR database [31] such as IPD, SC, POC, CBF, MI, FA, CC, ECG5000 are adopted. The statistics of the data set are shown in Table 1. In order to visualize the different patterns of time series in data sets, Fig. 2 and Fig. 3 respectively take IPD and SC data sets as examples. As we have seen, IPD dataset has 2 categories and SC dataset has 6 categories, three time series from each category are randomly fetched and drawn with different color curves respectively for each dataset. The horizontal axis "time" represents each time point in the time series, and the vertical axis "value" is the value of the time series at the corresponding time point. In the experiment, the training set and test set are regarded as offline and online data respectively, and 10% of the test data set is regarded as incremental data snapshots, which is automatically determined according to Hoeffding [32]. So each data set is divided into 10 incremental data snapshots. Because the time series of each data set have equal length, Euclidean distance is adopted as a distance function.

In order to evaluate the performance of IFCTS, we adopted two incremental fuzzy clustering algorithms DDMFC [23] and IFCM [22] to compare the cluster results. For IFCTS algorithm, only one threshold for removing the empty cluster needs to be set. The empty cluster is not deleted but removed from the subsequent clustering to ensure the efficiency of incremental learning process, that is, the removed empty cluster will no longer participate in the subsequent clustering. In order to better compare the clustering performance for different algorithms, the threshold for

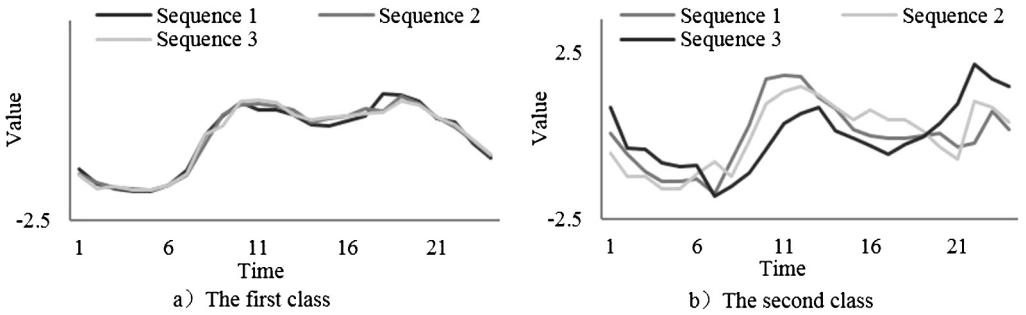

Fig. 2. IPD data set.

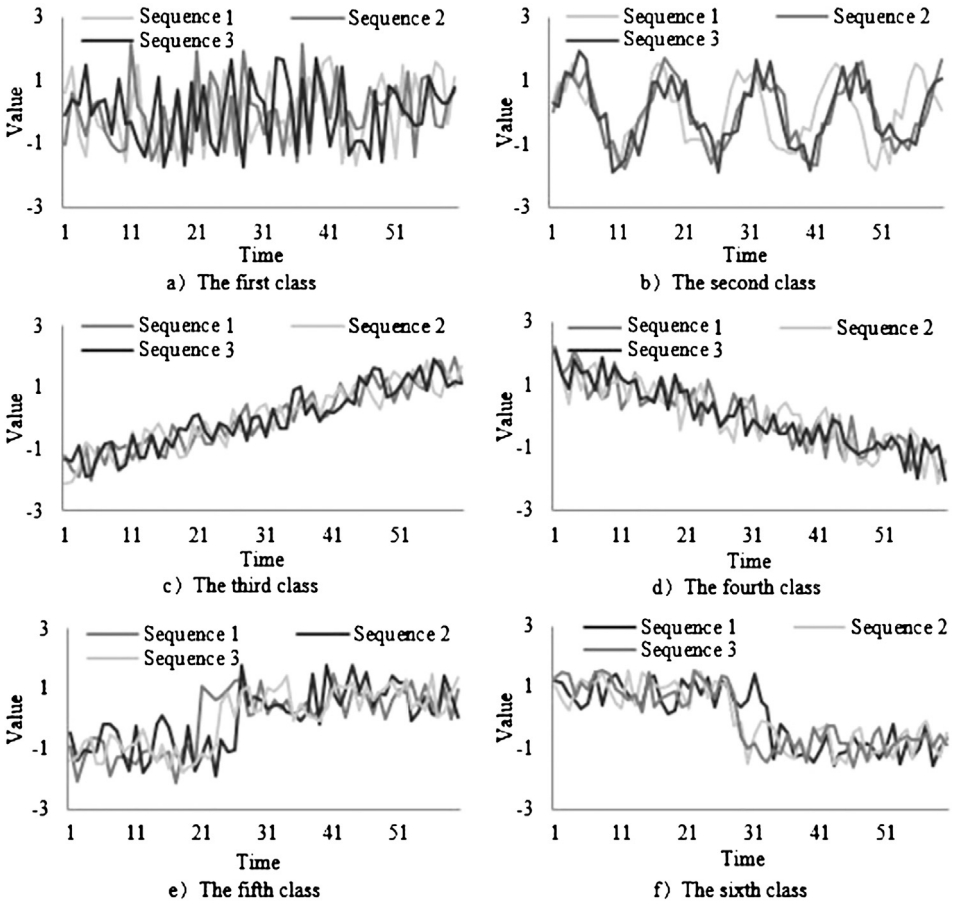

Fig. 3. SC data set.

removing the empty cluster is set to 10, which is equal to the number of data snapshots. For DDMFC algorithm, there are three parameters that need to be set. Parameter α is adopted to determine whether the data is outlier according to the difference between the membership of the data and the inverse of the number of cluster. Parameter β controls whether the cluster should be created by the ratio between the size of outlier data objects and the size of total data objects. Parameter T controls the removal of cluster. Here, the parameter $\alpha = 0.05$, $\beta = 0.2$ [18] and T = 10. For IFCM algorithm, there are two parameters that need to be set to identify the outlier data objects and control the creation of new clusters respectively. Since these two parameters can be converted into parameters α and β in DDMFC algorithm respectively, the parameter $\alpha = 0.05$, $\beta = 0.2$ are also adopted in the IFCM. In addition, the IFCM algorithm itself does not remove the clusters during incremental learning.

Table 2
Clustering comparison of different algorithms on different datasets.

Data set	IFCTS		IFCM		DDMFC	
	the number of the clusters	ACC	the number of the clusters	ACC	the number of the clusters	ACC
IPD	2	0.912±0.005	3	0.801±0.019	3	0.854±0.013
SC	6	0.941±0.005	6	0.883±0.010	6	0.901±0.008
POC	2	0.953±0.005	3	0.734±0.015	3	0.807±0.017
CBF	3	0.968±0.005	4	0.709±0.018	3	0.892±0.008
ECG5000	5	0.927±0.005	7	0.695±0.031	7	0.715±0.023
CC	4	0.890±0.005	6	0.601±0.037	6	0.606±0.035
MI	11	0.735±0.005	13	0.632±0.015	11	0.661±0.013
FA	14	0.812±0.005	15	0.750±0.029	16	0.733±0.022

After the experiment have completed over 20 independent runs, the cluster results are shown in Table 2. For IFCTS algorithm, the number of clusters obtained is the closest to the number of categories in the original data set, and the clustering precision obtained for each data set is also the highest among the three algorithms. In addition, because all these incremental fuzzy clustering algorithms can identify a few outlier data objects and add new clusters according to outlier data objects, the number of clusters obtained in some data sets will be more than the actual number of categories in the data set.

In order to compare the running efficiency of all algorithms, Fig. 4 shows the incremental learning time of all algorithms on each test set. It can be seen that IFCTS algorithm has the least running time for IPD, POC, CBF, ECG5000, CC, MI, and FA test sets. Only for the SC test set, the number of clusters obtained by the three algorithms are the same, so the overall running time is approximately the same. The ECG5000, CC, and FA test data sets contain more time series with longer lengths than other test data sets do. And the MI and FA test data sets contain more clusters than other test data sets do. So these four test sets ECG5000, CC, MI and FA take much more time than other test sets do. Compared with other algorithm, IFCTS performs best according to the cluster results, which has best adaptability and takes no more time in incremental learning without predefining the thresholds for identifying the outlier sample and controlling the generation of clusters.

In order to compare the incremental clustering results with different algorithm, Fig. 5 shows the Xie-Beni index value [29] for IPD data set in each data snapshot. It can be seen that V_{XB} value obtained by IFCTS algorithm in each data snapshot is the smallest, which indicate that IFCTS has the better clustering performance in each data snapshot compared with DDMFC and IFCM. For IPD dataset, the dynamics of the clusters formed in successive data snapshots with different algorithm are illustrated in Fig. 6. As we have seen, DDMFC created a new cluster in the ninth data snapshot that lead to the number of cluster centers changed from 2 to 3. Therefore, the V_{XB} value suddenly became larger than the previous data snapshot. Similarly, the IFCM created a new cluster in the sixth data snapshot, so a sudden change for the V_{XB} value also occurred. Through these above analysis, it can be proved that our proposed IFCTS algorithm has better scalability than other two algorithms for incremental clustering.

4.2. Performance evaluation with unequal-length time series

Because the unequal-length time series derived from segmenting the real time series have no priori class labels, it is difficult to compare the clustering performance between our proposed algorithm and other algorithm. In order to verify the clustering effect of IFCTS algorithm on unequal-length time series, the proposed algorithm is applied to the symbolization for the univariate time series, which means that the time point series are transformed into symbolic time interval series representation to discover the temporal patterns. By segmenting the time series into subsequences, they can be clustered to form the symbolic-value time intervals. Because the subsequences are with different length, the IFCTS algorithm will adopt DTW distance to measure the similarity between the subsequences. In this paper, the alphabetical representation is adopted to realize symbolization. After the clustering, the subsequences are clustered into different clusters, each cluster is represented as a letter in the alphabet=$\{A, B, C, ...\}$, each subsequence belongs to a cluster, which is transformed into a letter symbol.

In follows, our proposed algorithm is applied on multivariate hydrometeorological time series(HY) in the region of the Arecib in the United States [28], the HY dataset is collected with WINDSPEED (wind speed), GUSTS (gust)

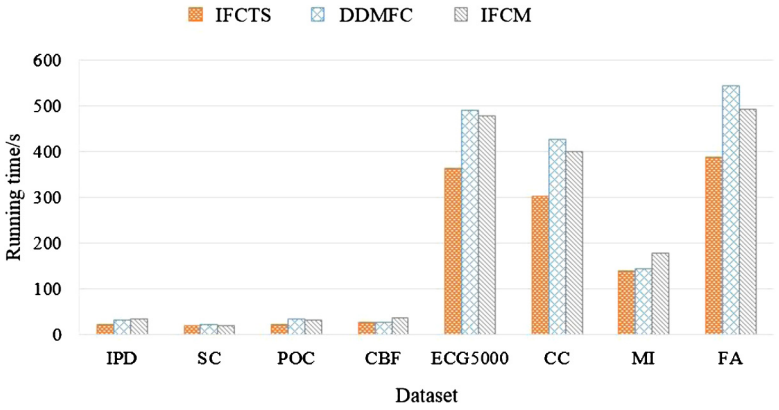

Fig. 4. Comparison of running time.

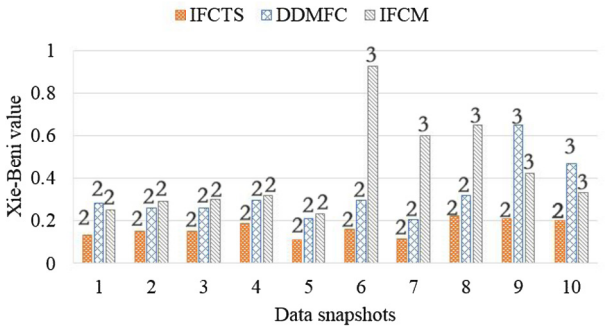

Fig. 5. Comparison of clustering effect in incremental learning process in the IPD data set. (The digit at the top of the bar graph represents the number of the clustering for corresponding algorithm.)

and DIR (wind direction) to evaluate the clustering results. In the experiment, the data collected every 6 minutes from January 1, 2013 to March 31, 2013 were fetched as the offline data, denoted as $DS[0]$, its length is 21600, from then, the data for subsequent every month was looked as the incremental data snapshot $DS[1]$, $DS[2]$, $\cdots$

First, the offline time series data in $DS[0]$ are segmented into subsequences with the segmentation algorithm based on dynamic programming [33], which has a good segmentation effect on multivariate time series data. After segmentation, the length of the subsequences obtained for each variable is unequal. Taking WINDSPEED as an example, 280 WINDSPEED subsequences are obtained. Due to unequal length of subsequences, DTW distance was adopted as the distance measurement to realize fuzzy clustering. The optimal number of clustering is 4 according to the Xie-Beni index. In order to show the general shape of the sequence in each cluster, any three subsequences randomly selected from each cluster are shown in Fig. 7. It can be seen that the subsequences between the four clusters have significantly different shapes while the subsequences in one cluster have similar shape.

Based on the above clustering results, IFCTS algorithm is adopted to make incremental learning for each incremental data snapshot. We still take the WINDSPEED sub-sequence as an example. Here, the threshold for removing the empty cluster is set to 2. Table 3 shows the evolution of the number of clusters for the offline data snapshot $DS[0]$ and 15 incremental data snapshots $DS[1]$, $DS[2]$, ..., $DS[15]$. The number of subsequences is obtained by segmenting the multivariate time series in each data snapshot, and the V_{XB} value is calculated according to the clustering results of each data snapshot. It can be seen from Table 3 that the number of clusters varies dynamically in the incremental learning process. IFCTS algorithm will create at least one new cluster as long as it finds the outlier sequence. Moreover, when a cluster is recognized as an empty cluster in two consecutive data snapshots, it will be removed from the next data snapshot and no longer participate in subsequent clustering to ensure the running efficiency of the algorithm.

In order to show the effect of the number of clusters on running efficiency more intuitively, Fig. 8 shows the running time of IFCTS algorithm for WINDSPEED sub-sequence in each incremental data snapshot. When the number of subsequences contained in each incremental data snapshot is not significantly different, the more clusters, the longer

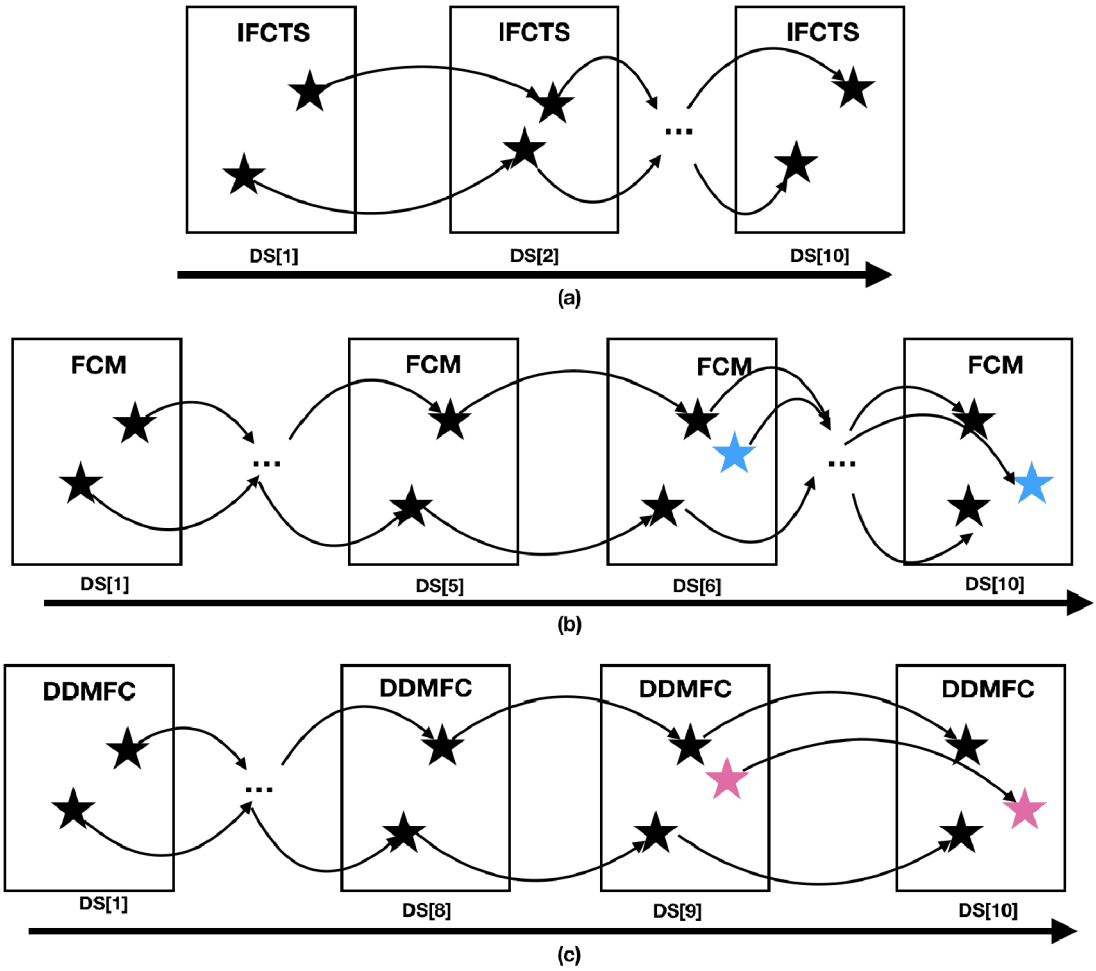

Fig. 6. Incremental learning processing of the clusters with different algorithm (a) IFCTS algorithm (b) IFCM algorithm (c) DDMFC algorithm.

Table 3
The evolution of the number of clusters.

	The number of sub-sequence	The number of Clusters	outlier sequence number	new cluster	empty cluster	removed cluster	V_{XB} value
DS[0]	280	4	-	-	-	-	0.341
DS[1]	96	4	0	0	0	0	0.312
DS[2]	95	5	3	1	0	0	0.607
DS[3]	95	6	2	1	0	0	0.841
DS[4]	101	6	0	0	1	0	0.732
DS[5]	93	8	5	2	1	0	1.105
DS[6]	89	7	0	0	0	1	0.884
DS[7]	97	7	0	0	1	0	0.697
DS[8]	93	8	4	1	2	0	1.019
DS[9]	95	8	2	1	0	1	0.983
DS[10]	99	8	0	0	0	0	0.801
DS[11]	91	9	4	1	2	0	1.199
DS[12]	95	9	0	0	2	0	0.845
DS[13]	100	8	2	1	0	2	0.828
DS[14]	98	8	0	0	0	0	0.702
DS[15]	96	9	1	1	2	0	1.003

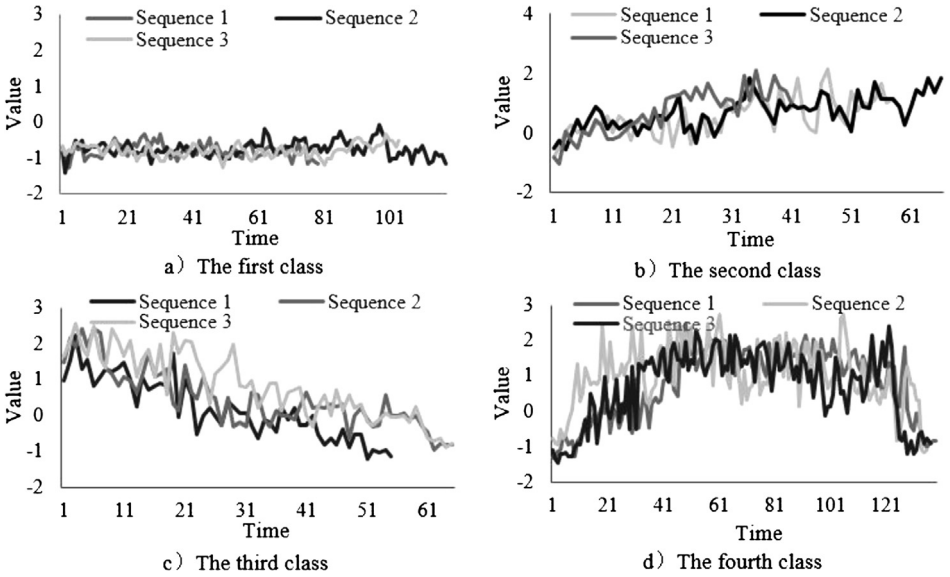

Fig. 7. Clustering results of WINDSPEED sequence in offline data.

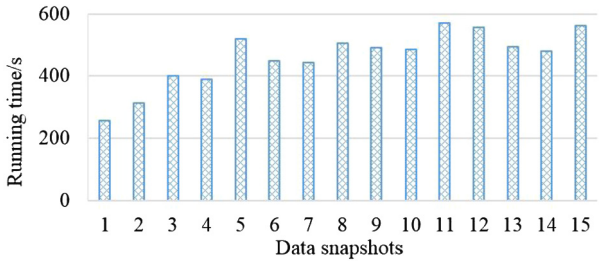

Fig. 8. The running time of incremental learning process.

time. The removal threshold can remove some empty clusters in time to make the clusters participating in subsequent clustering not excessive, which ensure that the running time of the algorithm will not keep increasing. Finally, the running time keeps stable about 500 s for each incremental data snapshot.

In addition, in order to evaluate the effect of different thresholds for removing the empty cluster on the algorithm, Fig. 9 shows the average V_{XB}, the running time, and the number of clusters for $DS[1]$, $DS[2]$, ..., $DS[15]$ under different thresholds for removing the empty cluster. As we have seen, when the threshold for removing the empty cluster is small, it is set to 1, the empty cluster identified in each incremental data snapshot is immediately removed, the overall running time and average V_{XB} of the clustering process are relatively small. However, in this case, if the next incremental data snapshot contains the pattern of the empty cluster, it is necessary to re-create new cluster, which leads to the total number of clusters obtained excessive due to partially overlapping clusters. When the threshold for removing the empty cluster is large, some of the empty clusters are not removed because new subsequences are added into the incremental data snapshot, which lead to the total number of clusters obtained relatively small by reducing the overlapping clusters. Moreover, due to the empty clusters, the running time of the whole clustering process will increase and the average V_{XB} will also become larger, which make the clustering results of each incremental data snapshot worse. In real applications, in order to obtain better efficiency and clustering results, the threshold for removing the empty cluster can be smaller; in order to reduce the overlapping clusters, the threshold should be set larger. In conclusion, the threshold should be set according to the clustering results.

After incrementally fuzzy clustering for the WINDSPEED subsequences, the subsequences contained in each cluster are transformed into symbolic interval representation. Taking the data snapshot $DS[1]$ and $DS[3]$ as examples, the WINDSPEED data records are extracted from the five consecutive days, the results of the segmentation and the symbols of subsequences are shown in Fig. 10 and Fig. 11 respectively. For example, $B(0.59)$ represents the symbol

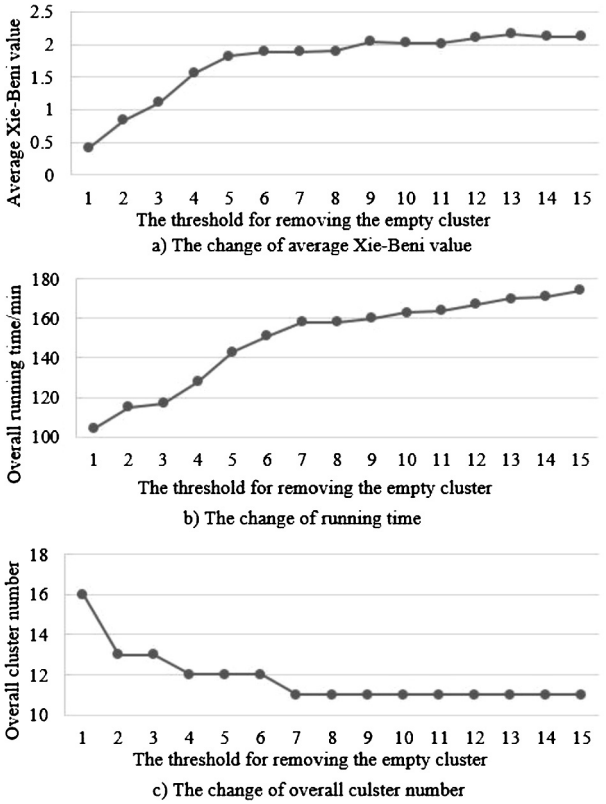

Fig. 9. The clustering effects about the threshold for removing the empty clusters.

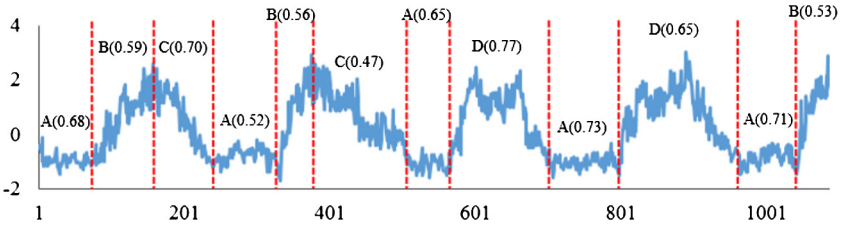

Fig. 10. Segmental symbolization of WINDSPEED sequence in $DS[1]$.

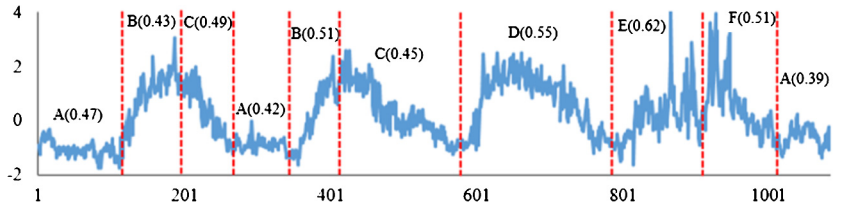

Fig. 11. Segmental symbolization of WINDSPEED sequence in $DS[3]$.

of subsequence, where the letter B represents the cluster symbol through the subsequence symbolization, and the value 0.59 represents the membership of the subsequence belongs to the cluster. According to the evolution of the number of clusters in Table 3, the WINDSPEED subsequences are respectively divided into 4 and 6 clusters in the data snapshot $DS[1]$ and $DS[3]$. Comparing with Fig. 10, two additional symbols E and F are obtained through the subsequence symbolization in Fig. 11, which indicates two new patterns are discovered. It can be found that IFCTS

can cluster the subsequences according to the tendency of the time series to obtain the symbolic representation of the whole sequence.

5. Conclusion

In this paper, we proposed an incremental fuzzy clustering of time series. It includes two stages: off-line clustering and on-line clustering. During the offline clustering process for time series, Xie-Beni index is introduced to FCM to automatically obtain the optimal number of initial clusters. On the basis of DTW distance, whether equal length or unequal length time series, the extensive FCM clustering can accurately and effectively obtain the centers of cluster. During the online clustering process for time series, IFCTS algorithm can inherit the off-line clustering results to update, create and remove the cluster structure according to the time series in the current incremental data snapshot, with which the outlier sequences are identified and the new clusters are created adaptively without predefining any parameters.

The experimental results show that IFCTS algorithm has achieved good clustering results by using different distance measurements for equal length and unequal length time series in order to capture the shape similarity between time series.

In the future, we will investigate some problems. First, the size of the data snapshots that varies with the distribution of the data will be studied. Second, we will apply our proposed algorithm to other domains to solve more complex problems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research work was supported by the National Natural Science Foundation of China (Grant No. 61572073), the National Natural Science Foundation of China (Grant No. 62076025), the Fundamental Research Funds for the Central Universities (FRF-GF-19-014B).

References

[1] S. Aghabozorgi, A.S. Shirkhorshidi, T.Y. Wah, Time-series clustering - a decade review, Inf. Sci. 53 (C) (2015) 16–38, https://doi.org/10.1016/j.is.2015.04.007.
[2] H. Izakian, W. Pedrycz, Agreement-based fuzzy c-means for clustering data with blocks of features, Neurocomputing 127 (127) (2014) 266–280, https://doi.org/10.1016/j.neucom.2013.08.006.
[3] Z. Bankó, J. Abonyi, Correlation based dynamic time warping of multivariate time series, Expert Syst. Appl. 39 (17) (2012) 12814–12823.
[4] V.S. Tseng, C.P. Kao, Efficiently mining gene expression data via a novel parameterless clustering method, IEEE/ACM Trans. Comput. Biol. Bioinform. 2 (4) (2005) 355–365, https://doi.org/10.1109/TCBB.2005.56.
[5] T. Górecki, Using derivatives in a longest common subsequence dissimilarity measure for time series classification, Pattern Recognit. Lett. 45 (1) (2014) 99–105, https://doi.org/10.1016/j.patrec.2014.03.009.
[6] J.A. Vilar, B. Lafuente-Rego, P. D'Urso, Quantile autocovariances: a powerful tool for hard and soft partitional clustering of time series, Fuzzy Sets Syst. 340 (2018) 38–72.
[7] P. D'Urso, L. De Giovanni, R. Massari, Robust fuzzy clustering of multivariate time trajectories, Int. J. Approx. Reason. 99 (2018) 12–38.
[8] F. Zhou, F.D.L. Torre, J.K. Hodgins, Hierarchical aligned cluster analysis for temporal clustering of human motion, IEEE Trans. Pattern Anal. Mach. Intell. 35 (3) (2013) 582–596, https://doi.org/10.1109/TPAMI.2012.137.
[9] X. Huang, Y. Ye, L. Xiong, R.Y.K. Lau, N. Jiang, S. Wang, Time series k-means: a new k-means type smooth subspace clustering for time series data, Inf. Sci. 367–368 (2016) 1–13, https://doi.org/10.1016/j.ins.2016.05.040.
[10] P. D'Urso, L. De Giovanni, R. Massari, GARCH-based robust fuzzy clustering of time series, Fuzzy Sets Syst. 305 (2016) 1–28.
[11] E.A. Maharaj, P. D'Urso, Fuzzy clustering of time series in the frequency domain, Inf. Sci. 181 (2011) 1187–1211.
[12] E.A. Maharaj, P. D'Urso, J. Caiado, Time Series Clustering and Classification, CRC Press, 2019.
[13] R. Coppi, P. D'Urso, P. Giordani, A fuzzy clustering model for multivariate spatial time series, J. Classif. 27 (2010) 54–88.
[14] P. D'Urso, E.A. Maharaj, Autocorrelation-based fuzzy clustering of time series, J. Classif. 160 (24) (2010) 3565–3589, https://doi.org/10.1016/j.fss.2009.04.013.

[15] S.R. Aghabozorgi, T.Y. Wah, A. Amini, M.R. Saybani, A new approach to present prototypes in clustering of time series, in: The 7th International Conference of Data Mining, vol. 1, 2011, pp. 214–220.
[16] H. Izakian, W. Pedrycz, I. Jamal, Fuzzy clustering of time series data using dynamic time warping distance, Eng. Appl. Artif. Intell. 39 (2015) 235–244, https://doi.org/10.1016/j.engappai.2014.12.015.
[17] J. Caido, E.A. Maharaj, P. D'Urso, Time series clustering, in: Handbook of Cluster Analysis, Chapman & Hall, 2016, pp. 241–264.
[18] S.R. Aghabozorgi, T.Y. Wah, Using incremental fuzzy clustering to web usage mining, in: 2009 International Conference of Soft Computing and Pattern Recognition, SOCPAR'09, IEEE, 2009, pp. 653–658.
[19] J.-P. Mei, Y. Wang, L. Chen, C. Miao, Incremental fuzzy clustering for document categorization, in: 2014 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE, IEEE, 2014, pp. 1518–1525.
[20] I. Saha, U. Maulik, Incremental learning based multiobjective fuzzy clustering for categorical data, Inf. Sci. 267 (2014) 35–57, https://doi.org/10.1016/j.ins.2013.12.057.
[21] Y. Wang, L. Chen, J.-P. Mei, Incremental fuzzy clustering with multiple medoids for large data, IEEE Trans. Fuzzy Syst. 22 (6) (2014) 1557–1568, https://doi.org/10.1109/TFUZZ.2014.2298244.
[22] B. Tudu, S. Ghosh, A. Bag, D. Ghosh, N. Bhattacharyya, R. Bandyopadhyay, Incremental FCM technique for black tea quality evaluation using an electronic nose, Fuzzy Inf. Eng. 7 (3) (2015) 275–289, https://doi.org/10.1016/j.fiae.2015.09.002.
[23] F. Crespo, R. Weber, A methodology for dynamic data mining based on fuzzy clustering, Fuzzy Sets Syst. 150 (2) (2005) 267–284, https://doi.org/10.1016/j.fss.2004.03.028.
[24] S. Aghabozorgi, M.R. Saybani, T.Y. Wah, Incremental clustering of time-series by fuzzy clustering, J. Inf. Sci. Eng. 28 (4) (2012) 671–688.
[25] D. Shan, X. Xu, T. Liang, S. Ding, Rank-adaptive non-negative matrix factorization, Cogn. Comput. 10 (1) (2018) 1–10.
[26] M. Du, S. Ding, Y. Xue, A robust density peaks clustering algorithm using fuzzy neighborhood, Int. J. Mach. Learn. Cybern. 12 (2017) 1–10, https://doi.org/10.1007/s13042-017-0636-1.
[27] S. Ding, X. Xu, S. Fan, Y. Xue, Locally adaptive multiple kernel k-means algorithm based on shared nearest neighbors, Soft Comput. 22 (14) (2017) 4573–4583, https://doi.org/10.1007/s00500-017-2640-5.
[28] Noaa/nos/co-ops, http://co-ops.nos.noaa.gov.
[29] S.H. Kwon, Cluster validity index for fuzzy clustering, Electron. Lett. 34 (22) (1998) 2176–2177, https://doi.org/10.1049/el:19981523.
[30] W. Ling, S. Hua, Evolving clustering method based on self-adaptive learning, Control Decis. 31 (3) (2016) 423–428, https://doi.org/10.13195/j.kzyjc.2014.1945.
[31] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, http://www.cs.ucr.edu/~eamonn/time_series_data, The UCR Time Series Classification Archive.
[32] P. Domingos, G. Hulten, Mining high-speed data streams, in: Kdd, vol. 2, 2000, p. 4.
[33] H. Guo, X. Liu, L. Song, Dynamic programming approach for segmentation of multivariate time series, Stoch. Environ. Res. Risk Assess. 29 (1) (2015) 265–273, https://doi.org/10.1007/s00477-014-0897-0.