Computing and Informatics, Vol. 36, 2017, 1361–1384, doi: 10.4149/cai_2017_6_1361

REPRESENTATIVE POINTS AND CLUSTER ATTRIBUTES BASED INCREMENTAL SEQUENCE CLUSTERING ALGORITHM

Di Wu

Department of Information and Electronic Engineering
Hebei University of Engineering, China

Jiadong Ren

College of Information Science and Engineering
Yanshan University, China
e-mail: bestmoogoo@163.com

Abstract. In order to improve the execution time and clustering quality of sequence clustering algorithm in large-scale dynamic dataset, a novel algorithm RPCAISC (Representative Points and Cluster Attributes Based Incremental Sequence Clustering) was presented. In this paper, density factor is defined. The primary representative point that has a density factor less than the prescribed threshold will be deleted directly. New representative points can be reselected from non-representative points. Moreover, the representative points of each cluster are modeled using the K-nearest neighbor method. The definition of the relevant degree (RD) between clusters was also proposed. The RD is computed by comprehensively considering the correlations of objects within a cluster and between different clusters. Then, whether the two clusters need to merge is determined. Additionally, the cluster attributes of the initial clustering are retained with this process. By calculating the matching degree between the incremental sequence and the existing cluster attributes, dynamic sequence clustering can be achieved. The theoretic experimental results and analysis prove that RPCAISC has better correct rate of clustering results and execution efficiency.

Keywords: Sequence clustering, incremental clustering, representative point, cluster attributes, similarity

1 INTRODUCTION

In recent years, with the development of science and technology progress, the data size of the practical application field increased rapidly. It has called great attention from all walks of life. Big data has a profound effect on the future of technological and economic development [1], even national security. Large data has the characteristics of volume, variety, velocity. Therefore, how to quickly mine and analyze the potential information and knowledge from mass data has been an emergent task [2].

Sequence clustering has become an important branch of data mining, and widely used in many fields [3, 4, 5, 6], such as pattern recognition, bioinformatics, web recommendation system, text mining, image analysis, and so on. However, with the dramatic increase in the amount of data processing, traditional sequence clustering algorithms still remain a challenging goal [7].

Guha et al. presented an efficient clustering algorithm called CURE [8] for use with large databases. Each cluster is represented by a certain fixed number of representative points; then, representative points are moved towards the clustering center by a specified fraction. The CURE algorithm has been shown to be more robust to outliers and to identify non-spherical clusters and wide variances in size [9]. However, the efficiency of the CURE algorithm is not ideal.

With the development of the CURE algorithm, researchers are now investigating representative points based clustering algorithms [10, 11, 12, 13, 14]. Many relevant algorithms have been proposed.

Chen et al. designed the CARDC (Clustering Algorithm using Representative Data of Clusters) algorithm [15]. In this algorithm, some scattered data points can be viewed as the representatives of clusters. Then, the pair of clusters that have the smallest distance among all of the pairs of clusters are merged into one cluster. The algorithm can capture the geometry features of clusters with different shapes and sizes. In [16], the SCM (Subspace Classification based on Multi-representatives) algorithm was proposed. The SCM algorithm allows the samples that belong to different classes in the entire space to be easily separable. Arajo et al. [17] developed a new cost evaluation function for clustering using all points in a dataset; their function uses only representative points to quantify the interaction between distributions without any loss of the original properties of the potential cross information. In addition, Huang et al. [18] presented a clustering algorithm called the CAMFT algorithm that is based on a multi-representation feature tree. In [19], an improved clustering algorithm called the REPBFC (Representative Points Based Fast Clustering) algorithm was developed based on the CURE algorithm and the 90/10 rule.

The above representative points clustering algorithms are all based on CURE. According to selecting a sufficient number of data points in each cluster as representative points, the datasets are clustered by analyzing these representative points. The probability that the noise in cluster edge will be selected as a representative point is large.

DBSCAN algorithm [20] is a classical clustering algorithm based on density. The high density dataset can be effectively divided into clusters, and arbitrary shape clusters can be found in the spatial dataset with noises. In DBSCAN, if the number of objects in ε-neighborhood of point P is larger than MinPts, then a new cluster which has the core object P is established. By repeatedly finding the points which is density-reachable from the core object, the corresponding cluster is enlarged, and the final clustering results can be obtained.

Based on the DBSCAN, the density of the dataset should be considered comprehensively. Appropriate representative points will be chosen. While calculating the similarity between two clusters, only the minimum distance between the representative points of two clusters is considered. This strategy allows the similarity accuracy of the clusters to be seriously affected by the individual representative point of the clusters. Additionally, a portion of the information in the clusters will be overlooked.

Recently, in some practical applications, a dataset was updated constantly, and all data were analyzed using static clustering algorithms. Usually, the size of the incremental dataset is large, the computation cost can increase rapidly. Therefore, incremental algorithms have been studied by scholars to address the problem of dynamic datasets. Incremental data must only be processed by an incremental clustering algorithm. The clustering results produced then can be incrementally updated by taking a full advantage of the previous results [21]. This can improve efficiency significantly.

Luhr et al. designed an incremental clustering algorithm called RepStream that uses connectivity-based representative points [22]. The representative points in this algorithm are applied to incrementally cluster new data and selectively retain important cluster information within a knowledge repository. Then, the repository can be subsequently used to assist in the processing of new data, the archival backup of critical features for off-line analysis, and in the identification of recurrent patterns. In [23], based on the relationship between new and existing representative points, the cluster is determined where the new points should be added to. Where one cluster is containing the existing representative points, the other is promoted with new representative points. Thus the algorithm is not sensitive to the parameter values and reduces memory consumption.

Although the execution efficiency of these incremental clustering algorithms for dynamic database is improved, however, the similarity measurement and distance calculation are mostly adopted to determine the cluster where the incremental data belongs to. When the error rate of similarity measurement function is large, the clustering accuracy will be affected seriously [24].

Based on the traditional representative points algorithm and incremental clustering algorithm, a novel algorithm representative points and cluster attributes based incremental sequence clustering is presented. In order to reduce the possibility of noise data as a representative point, the density factors of all primary representative points are obtained by considering the densities. Representative points are selected effectively. Moreover, whether the two clusters need to merge is determined. The relevant degree is computed by comprehensive considering the correlations of objects within a cluster and between different clusters. For dynamic dataset, by calculating the matching degree between the incremental sequence and the existing cluster attributes, dynamic sequence clustering can be achieved.

The reminder of this paper is organized as follows. In Section 2, we describe the problem definitions. Section 3 gives the RPCAISC algorithm. In Section 4, the efficiency of the proposed method on several experimental results is analyzed. Finally, we offer our conclusions and future work in Section 5.

2 PROBLEM DEFINITIONS

Given that SD denotes the sequence database [25] where the number of sequences in SD is represented as N. $S = a_1 a_2 \dots a_i \dots a_n$ is any sequence of SD, a_i indicates the i^{th} item, and $a_i \in L$ $(1 \leq i \leq n)$ where $L = \{a_1, a_2, \dots a_m\}$ indicates the set of items, the number of items in L is m. Let $SE = \{SE_1, SE_2, \dots, SE_M\}$ be a set of sequence elements in SD, $SE_M = a_p a_q$ represents the M^{th} pair of items, $p < q$, $a_p \in L$ and $a_q \in L$. The number of sequence elements is M, and $M = m^2$.

Assume that $SC_i = \{S_1, S_2, \dots, S_j, \dots, S_N\}$ is the i^{th} cluster of SD, where S_j $(1 \leq j \leq N)$ is any sequence in SC_i. If point Rep_x cannot be represented by any other object Rep_y, and the number of the objects in D-neighborhood is greater than K, then Rep_x is viewed as a representative point where the parameter D is the distance threshold from the sequence. According to the size and distribution of the selected dataset, the D-neighborhood of the algorithm is set, then repeatedly changes D by observing the results. It is not hard to see that the objects in D-neighborhood can be represented by the corresponding representative point.

Given that the representative points set of SC_i is $Rep(SC_i) = \{Rep_{i1}, Rep_{i2}, \dots, Rep_{ii}, \dots, Rep_{iR}\}$, here the number of representative points in cluster is expressed by R. The cluster partition $\{SC_{i1}, SC_{i2}, \dots, SC_{ii}, \dots, SC_{iR}\}$ can be gained by the minimum distance between each sequence and the corresponding representative point. There is a consistent one-to-one match between each cluster partition SC_{ii} and representative point Rep_{ii}. All the sequences in SC_{ii} have the minimum distance from Rep_{ii}.

In this paper, the distance $DISim(S_i, S_j)$ between sequences S_i and S_j is shown as follows.

$$DISim(S_i, S_j) = \sqrt{\sum_{t=1}^{M} (S_i(SE_t) - S_j(SE_t))^2} \tag{1}$$

where $S_i(SE_t)$ and $S_j(SE_t)$ are the corresponding values of t-dimensional vectors of S_i and S_j, respectively. M represents the number of sequence elements.

How to select the appropriate cluster representative points is an essential part of representative points based clustering algorithm. It will directly affect the accuracy of the final clustering results. Therefore, the density, distribution and other information of dataset should be well demonstrated by representative points as realistically as possible. In this paper, the definition of density factor is presented.

Definition 1. Density factor of representative point. The density factor of representative point Rep_{ii} is the ratio of the number of sequences in cluster partition SC_{ii} and the number of objects in cluster SC_i. The density factor of Rep_{ii} is described as $DF(Rep_{ii})$, which can be shown as below.

$$DF(\mathrm{Re}p_{ii}) = \frac{|SC_{ii}|}{|SC_i|}. \tag{2}$$

It is easy to see from the Definition 1 that for the representative points in intensive area the corresponding density factor is larger. Meanwhile, for the objects in sparse area, the value is small. Hence density distribution of sequence dataset can be reflected by dense factor of representative point. Outlier or noise will also be identified to some extent.

When merging the representative points of clusters, based on the KNN (K-Nearest Neighbor) method, the model of representative points in each cluster is built in RPCAISC algorithm. In order to determine whether the two clusters need to merge, the definition of relevant degree between clusters is designed below.

Definition 2. Relevant degree between clusters. If the sequence set SS_i and SS_j simultaneously satisfy the following two conditions:

1. $SS_i = \{S\,|S \in \mathrm{Re}p(SC_i) \wedge (\exists S')\ S' \in \mathrm{Re}p(SC_j), S' \in KNN(S)\}$;
2. $SS_j = \{S\,|S \in \mathrm{Re}p(SC_j) \wedge (\exists S')\ S' \in \mathrm{Re}p(SC_i), S' \in KNN(S)\}$,

then relevant degree $RD(SC_i, SC_j)$ between clusters SC_i and SC_j can be shown as follows.

$$RD(SC_i, SC_j) = \sum_{S \in SS_i \cap SS_j} DF(S) \tag{3}$$

where $Rep(SC_i) = \{Rep_{i1}, Rep_{i2}, \ldots, Rep_{iR}\}$ and $Rep(SC_j) = \{Rep_{j1}, Rep_{j2}, \ldots, Rep_{jR}\}$ describe the representative points set of SC_i and SC_j. For any Rep_{iq} of SC_i, $DF(Rep_{iq})$ is the density factor of Rep_{iq}. For all representative points that are contained in SC_i and SC_j, $KNN(Rep_{iq})$ represents the former K-nearest neighbor set of Rep_{iq}.

Example 1. Compute $RD(SC_1, SC_2)$ and $RD(SC_3, SC_4)$ in Figures 1 a) and 1 b).

In Figures 1 a) and 1 b), 3-nearest neighbor graphs of SC_1 and SC_2, SC_3 and SC_4 are established. According to Definition 2, in SC_1, there does not exist the 3-nearest neighbor of the representative point whose density factor is 0.6 of SC_2. Thus $RD(SC_1, SC_2) = 0.4 + 0.5 = 0.9$. However, in SC_3, there exists the 3-nearest neighbor of the representative point whose density factor is 0.7 of SC_4. Thus $RD(SC_3, SC_4) = 0.2 + 0.3 + 0.7 = 1.2$.

In addition, the center of SC_i is described as $SCCen_i$, $|SC_i|$ expresses the number of sequences in SC_i, $Ave(SC_i)$ represents the average value of SC_i, the variance of SC_i named $SCVar_i$, the third-order central moment $ThirdCM(SC_i)$ and the fourth-order central moment $FourthCM(SC_i)$ are all seen as the cluster

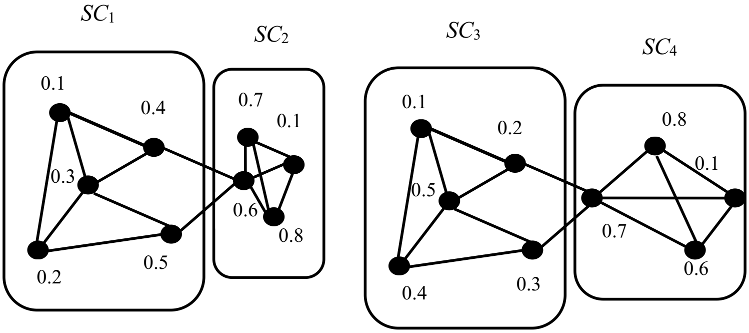

Figure 1. The 3-nearest neighbor graphs of clusters

attributes of SC_i. The corresponding attributes of each cluster and the clustering results are saved. The results will prepare for the incremental sequence clustering.

Therefore, the attribute vector of SC_i can be expressed as $AV(SC_i) = \{SCCen_i, |SC_i|, Ave(SC_i), SCVar_i, ThirdCM(SC_i), FourthCM(SC_i)\}$. Calculation method for each attribute is shown as follows.

$$SCCen_i = \sum_{S_j \in SC_i} \frac{S_j}{|SC_i|}, \tag{4}$$

$$Ave(SC_i) = \sum_{j=1}^{|SC_i|} \frac{DISim(S_j, SCCen_i)}{|SC_i|}, \tag{5}$$

$$SCVar_i = \sum_{j=1}^{|SC_i|} \frac{(DISim(S_j, SCCen_i) - Ave(SC_i))^2}{|SC_i|}, \tag{6}$$

$$ThirdCM(SC_i) = \sum_{j=1}^{|SC_i|} \frac{(DISim(S_j, SCCen_i) - Ave(SC_i))^3}{|SC_i|}, \tag{7}$$

$$FourthCM(SC_i) = \sum_{j=1}^{|SC_i|} \frac{(DISim(S_j, SCCen_i) - Ave(SC_i))^4}{|SC_i|} \tag{8}$$

where $|SC_i|$ is the number of sequences in SC_i, and $DISim(S_j, SCCen_i)$ is the distance between S_j and $SCCen_i$.

3 INCREMENTAL SEQUENCE CLUSTERING ALGORITHM

In this paper, representative points and cluster attributes based incremental sequence clustering algorithm called RPCAISC is proposed. It includes two stages.

They are the initial sequence clustering stage and the incremental sequence clustering stage. The framework of RPCAISC is shown in Figure 2.

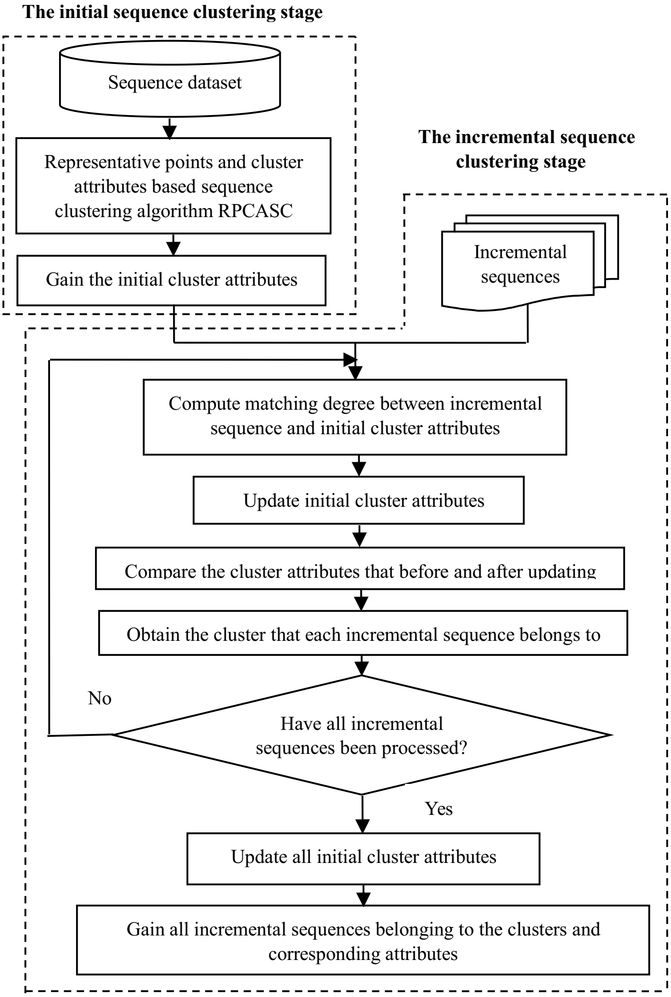
The initial sequence clustering stage
Sequence dataset
Representative points and cluster attributes based sequence clustering algorithm RPCASC
Gain the initial cluster attributes
The incremental sequence clustering stage
Incremental sequences
Compute matching degree between incremental sequence and initial cluster attributes
Update initial cluster attributes
Compare the cluster attributes that before and after updating
Obtain the cluster that each incremental sequence belongs to
Have all incremental sequences been processed?
No
Yes
Update all initial cluster attributes
Gain all incremental sequences belonging to the clusters and corresponding attributes

Figure 2. The framework of RPCAISC

3.1 Representative Points and Cluster Attributes Based Sequence Clustering Algorithm RPCASC

Representative points and cluster attributes based sequence clustering named RPCASC is presented in this section. First and foremost, the representative points of each cluster are selected. By calculating density factors of representative points in corresponding clusters, whether the object that is a real representative point or a noise can be determined. Moreover, by establishing K-nearest neighbor graph of the representative points and computing relevant degree between clusters, it can be determined whether two clusters need to merge.

3.1.1 Cluster Representative Points Selecting

In this section, the initial clustering of SD is implemented before selecting the representative points of each cluster. In classical CURE algorithm, each data is viewed as a separate cluster in early clustering. By merging constantly the nearest two clusters, and reselecting new representative points in updated merging cluster, the representative points of the final clusters are gained. If the number of objects in any cluster is less than the fixed number of representative points, then all objects in the cluster can be directly seen as representative points. For large-scale dataset, the time cost is not ideal.

Therefore, cluster representative points selection algorithm named CRPSA is proposed. Compared with the traditional similar algorithms, the calculation of density factor of representative point is added in CRPSA. In this way, noises will be excluded as much as possible. The probability that the representative point appears in the sparse area of cluster will be also reduced. The process of CRPSA is given below.

Algorithm 1. CRPSA

```
for each $j \in N$ do
    if $DISim(S_j, S_i) \leq \eta$ then // $\eta$ is the distance threshold;
        $S_i$ is the $D$-neighborhood of $S_j$;
    end if
    compute $NSD(S_j)$; // $NSD(S_j)$ is the number of objects
                        // in $D$-neighborhood of $S_j$
end for
for each $NSD(S_j) \in NSD(S_N)$ do
    if $NSD(S_j)$ is larger than any other $NSD(S_i)$ then
        $S_j$ is viewed as the first clustering center;
    end if
end for
for $q = 2$ to $Q$ do
    $S_i$ with the minimum similarity form $S_j$ will be seen as the next clustering
    center;
```

```
14. end for
15. for each $SC_m \in SC_Q$ do // $Q$ is the number of user-defined clusters;
16.   if $|SC_m| \leq R$ // $|SC_m|$ is the number of objects in cluster $SC_m$ then
17.     if $NSD(S_i) \geq K$ and $S_i$ is not represented by any determined
        representative point then // $K$ is the number of user-defined objects
                                   // in $D$-neighborhood
18.       $S_i$ is added to $Rep(SC_m)$;
19.     end if
20.   end if
21. end for
22. for each $Rep_{iq}$
23.   calculate $DF(Rep_{iq})$;
24.     if $DF(Rep_{iq}) < \theta$ then
25.       $Rep_{iq}$ is deleted from $Rep(SC_i)$, select a new $Rep_{inew}$ which does not
          belong to $Rep(SC_i)$ and the object in $D$-neighborhood of $Rep(SC_i)$;
26.       $Rep_{inew}$ is added to $Rep(SC_i)$;
27.     end if
28. end for
29. the final $R$ representative points in each $Rep(SC_i)$ can be gained.
```

In CRPSA algorithm, the initial clustering process is described by 1–14. According to the number of objects in D-neighborhood of sequence, whether the object that is a representative point can be determined in 15–21. Noises and edge points in the representative points set can be excluded. The CRPSA tends to select more cluster representative points in dense regions. In order to obtain better results and efficiency in the process of representative point adjustment, we set the density factor threshold $\theta = \frac{1}{5\mathrm{R}}$ where $0 < \theta < 1$. R is the number of representative points in cluster. A fixed number of initial representative points in each cluster is obtained. They are as dispersive as possible.

3.1.2 Cluster Representative Points Merging

After selecting representative points in each cluster by CRPSA algorithm, the density distribution and shape information can be described accurately by the obtained representative points. It will make a good foundation for the process of merging cluster representative points.

In this section, cluster representative points merging algorithm called CRPMA is presented. The process of CRPMA is given as follows.

Algorithm 2. CRPMA

```
for each $Rep(SC_i) \in Rep(SC_Q)$ do
  compute the $K$-nearest neighbor of the objects, $K$-nearest neighbor graph
  of the representative points is modeled;
end for
```

```
calculate the relevant degree $RD(SC_x, SC_y)$ between $SC_x$ and $SC_y$;
for each $SC_x, SC_y \in SC_Q$ do
  if $RD(SC_x, SC_y) \geq \varepsilon$ or all objects are combined into a cluster then
    // $\varepsilon$ is the threshold of relevant degree between clusters;
    if $RD(SC_x, SC_y)$ is larger than any other one then
      $SC_x$ and $SC_y$ will be combined;
      compute the relevant degree $RD(SC_x, SC_y)$ between $SC_{xy}$ and any
        other cluster in $SD$;
      $Rep(SC_{xy})$ is obtained;
      build the corresponding $K$-nearest neighbor graph;
      the two clusters which are the most relevant with $SC_x$ and $SC_y$ are
        gained;
      recalculate the relevant degree between the clusters and any other
        cluster in $SD$;
    end if
  end if
end for
```

It should be noted that if the growth rate of a cluster is very slow in the merging process of the clusters, then it is viewed as a noise and deleted. In algorithm CRPMA, through computing the relevant degree between any two clusters, it is determined whether the two clusters will be merged. The process has the following four characteristics.

First and foremost, K-nearest neighbor graph is modeled by the representative points within each cluster. Thus the scale of the graph is small, then CRPMA has a better time efficiency. Moreover, the definition of relevant degree between clusters is based on KNN method. The neighborhood of representatives can be captured dynamically. It makes the relevant degree not only relate with the connection between clusters, but also link to the connection between sequences in the same cluster. By using the concept of "relative" to calculate the relevant degree between clusters, the merging process of the clusters will be more reasonable. For those clusters which are close to each other, in the traditional algorithms it is difficult to determine whether the clusters need to merge. CRPMA is based on density to compute the relevant degree. The two clusters which have little difference in density are selected to combine. Finally, the density information of representative points is full considered in the merging process.

After obtaining the combined clusters, the sequence cluster center $SCCen_i$, the number of sequences in SC_i called $|SC_i|$, the average of sequence clusters $Ave(SC_i)$, the sequence cluster variance $SCVar_i$, the third-order central moment $ThirdCM(SC_i)$, the fourth-order central moment $FourthCM(SC_i)$ are selected as cluster attributes. Representative points and cluster attributes based sequence clustering algorithm named RPCASC is described as follows.

Algorithm 3. RPCASC

```
for each $Rep(SC_i) \in Rep(SC_Q)$ do
    compute $R$ representative points;
end for
for each $SC_x, SC_y \in SC_Q$ do
    calculate the $RD(SC_x, SC_y)$ between $SC_x$ and $SC_y$;
    if $RD(SC_x, SC_y) \geq \varepsilon$ or all objects are combined into a cluster then
        if $RD(SC_x, SC_y)$ is larger than any other one then
            $SC_x$ and $SC_y$ will be combined, and $Rep(SC_{xy})$ is computed;
            the clusters which need to merge are combined;
        end if
    end if
end for
for each final determined cluster $SC_Q$ do
    compute $AV(SC_Q)$, the cluster results and the corresponding cluster
      attributes vector are saved.
end for
```

To RPCASC, $AV(SC_Q)$ describes the attribute vector of SC_Q, where $AV(SC_Q) = \{SCCen_Q, |SC_Q|, Ave(SC_Q), SCVar_Q, ThirdCM(SC_Q), FourthCM(SC_Q)\}$. It will be saved and used in the process of incremental sequence clustering.

3.2 Incremental Sequence Clustering

Before incremental sequence clustering, incremental sequences are needed to preprocess. It is the same as the process stage in [26]. Furthermore, based on the saved cluster attributes vector, the matching degree between each incremental sequence S_j and the existing cluster center is calculated. The matching degree between S_j and $SCCen_i$ can be represented as $MD(S_j, SCCen_i)$ and shown as follows.

$$MD(S_j, SCCen_i) = (1 - \alpha)Sim(S_j, SCCen_i) - \alpha DISim(S_j, SCCen_i) \tag{9}$$

where α is the weight factor, $0 \leq \alpha \leq 1$. $Sim(S_j, SCCen_i)$ is the similarity between sequences S_j and $SCCen_i$. It is shown as follows.

$$Sim(S_j, SCCen_i) = \frac{|SE(S_j) \cap SE(SCCen_i)|}{|SE(S_j) \cup SE(SCCen_i)|} \tag{10}$$

where $SE(S_j)$ and $SE(SCCen_i)$ are the sequence elements that S_j and $SCCen_i$ support. $|SE(S_j) \cap SE(SCCen_i)|$ expresses the number of collective sequence elements between S_j and $SCCen_i$. The number of sequence elements both S_j and $SCCen_i$ support is represented as $|SE(S_j) \cup SE(SCCen_i)|$.

The cluster with the largest matching degree $SC_{\max}$ is recorded as below.

$$SC_{\max} = \arg \underset{1 \leq i \leq Q}{MaxMD}(S_j, SCCen_i)\,. \tag{11}$$

According to the results of the formula (9), the cluster attribute vector with the largest matching degree is recorded as $AV(SC_i)_{pre}$. S_j is added to the corresponding cluster. Then the cluster attributes vector is updated and described as $AV(SC_i)_{now}$. By comparing with $AV(SC_i)_{pre}$ and $AV(SC_i)_{now}$, the affiliation relationship between S_j and SC_i can be determined by the following three conditions.

1. $\mathit{Ave}(SC_i)_{now} < \mathit{Ave}(SC_i)_{pre}$;
2. $SCVar_{inow} < SCVar_{ipre}$, and $\mathit{Ave}(SC_i)_{now} < \mathit{Ave}(SC_i)_{pre}$;
3. $\mathit{ThirdCM}(SC_i)_{now} < \mathit{ThirdCM}(SC_i)_{pre}$, and $\mathit{FourthCM}(SC_i)_{now} < \mathit{FourthCM}(SC_i)_{pre}$.

If the relationship of $AV(SC_i)_{pre}$ and $AV(SC_i)_{now}$ is satisfied one of the above three conditions, then S_j will be added to SC_i. The cluster identification of S_j is same as SC_i. On the contrary, the identification is recorded as $Q + 1$. Here we set Q clusters are obtained by RPCASC. Repeat the judgment process until all the incremental sequences are dealt with. The incremental sequences with cluster identification $Q + 1$ are added to a new cluster SC_{Q+1}. Meanwhile, the number of clusters is changed from Q to $Q + 1$.

Attribute vectors of all clusters are updated. Additionally, the clustering results of incremental sequences are also saved. The process of incremental clustering behind is consistent with the above. Representative points and cluster attributes based incremental sequence clustering algorithm called RPCAISC is given as follows.

Algorithm 4. RPCAISC

```
for each $S_j$ do // $S_j$ is any incremental sequence
    compute the matching degree between each incremental sequence
    and the final clustering centers which are obtained by RPCASC;
    if $MD(S_j, SCCen_i)$ is larger than any other one then
        $SC_i$ is selected as the most similar cluster to $S_j$;
        compute $AV(SC_i)_{pre}$;
        $S_j$ is added to $SC_i$, $AV(SC_i)_{pre}$ is updated to $AV(SC_i)_{now}$;
        compare $AV(SC_i)_{pre}$ and $AV(SC_i)_{now}$, the cluster identification of $SC_j$
        is recorded;
        if cluster identification of $SC_j$ is $Q + 1$ then
            $SC_{Q+1}$ is increased to $SD$, the number of clusters is changed from $Q$
            to $Q + 1$;
        end if
    end if
end for
```

In RPCAISC, instead of repeating clustering incremental sequences with the original sequences in SD, incremental sequences are only clustered with attribute vectors. The time cost can be decreased effectively.

In order to better describe the meaning of the symbols in this paper, a list of symbols in RPCAISC is shown as Table 1.

Number	Symbol	The meaning of the symbol
1	SD	Sequence database
2	N	The number of sequences in SD
3	S	Any sequence of SD
4	a_i	The i^{th} item
5	L	The set of items
6	m	The number of items in L
7	SE	The set of sequence elements in SD
8	SE_M	The M^{th} pair of items
9	M	The number of sequence elements
10	SC_i	The i^{th} cluster of SD
11	S_j	The j^{th} sequence in SD
12	D	The distance threshold from the sequence
13	K	The user-defined number of neighborhoods
14	Rep_x	The representative point
15	$Rep(SC_i)$	The representative points set of SC_i
16	R	The number of representative points in cluster
17	SC_{ii}	The cluster partition
18	Rep_{ii}	The representative point in SC_{ii}
19	$Sim(S_i, S_j)$	The similarity between sequences S_i and S_j
20	$SE(S_j)$	The sequence elements that S_j support
21	$DF(Rep_{ii})$	The density factor of Rep_{ii}
22	$\|SC_{ii}\|$	The number of sequences in cluster partition SC_{ii}
23	$\|SC_i\|$	The number of objects in cluster SC_i
24	$KNN(Rep_{iq})$	The former K-nearest neighbor set of Rep_{iq}
25	$RD(SC_i, SC_j)$	The relevant degree between SC_i and SC_j
26	$SCCen_i$	The center of SC_i
27	$Ave(SC_i)$	The average value of SC_i
28	$SCVar_i$	The variance of SC_i
29	$ThirdCM(SC_i)$	The third-order central moment
30	$FourthCM(SC_i)$	The fourth-order central moment
31	$AV(SC_i)$	The attribute vector of SC_i
32	$DISim(S_j, SCCen_i)$	The distance between SC_j and $SCCen_i$
33	$S_j(SE_t)$	The value of t-dimensional vector of S_j
34	$SCCen_i(SE_t)$	The value of t-dimensional vector of $SCCen_i$
35	$NSD(S_j)$	The number of objects in D-neighborhood
36	η	The distance threshold
37	θ	The density factor threshold
38	ε	The threshold of relevant degree between clusters
39	α	The weight factor
40	$MD(S_j, SCCen_i)$	The matching degree between S_j and $SCCen_i$
41	$SC_{\max}$	The cluster with the largest matching degree
42	$AV(SC_i)_{pre}$	The cluster attribute vector of SC_i
43	$AV(SC_i)_{now}$	The updated cluster attribute vector of SC_i

Table 1. A list of symbols in RPCAISC

4 EXPERIMENTAL RESULTS AND ANALYSIS

In order to verify the performances of RPCAISC, DBSCAN, CURE, CARDC and RepStream, the experimental tests were conducted with two datasets. The two datsets are the synthetic sequence dataset D100C6T14S5N001 and the real dataset Segmentation-all of the classic machine learning database UCI [27]. Segmentation is one of the classic dataset in UCI. It commonly includes 210 objects. In this paper, the data collection of Segmentation is used, and it is 2 310 samples in Segmentation-all. D100C6T14S5N001 is generated by IBM data generator [28], it contains 100 000 sequences, 15 attribute dimensions and 20 clusters, the essential parameters of D100C6T14S5N001 are N, D, C, T, S, I, the specific meaning of the parameters is described in Table 2.

Parameters	The Meaning of the Parameters
N	The number of different items
D	The number of the sequences in sequence database
C	The average number of the item sets in each sequences
T	The average number of items in each item sets
S	The average length of potential maximal sequence patterns
I	The average length of maximal frequent item sets

Table 2. The parameters of IBM data generator

In addition, the parameters of real dataset Segmentation-all are described as the following Table 3.

The Number of Sequences	Attribute Dimensions	The Number of Clusters
2 310	19	7

Table 3. The parameters of real dataset

In this section, incremental clustering and non-incremental clustering algorithms are divided into two groups to experiment. For the synthetic dataset D100C6T14S5N001, the number of sequences is 20 000 in the stage of initial clustering. 1 000 sequences are randomly selected in each cluster. In the stage of incremental clustering, 10 000 sequences are increased each time. 500 are randomly selected in each cluster. The incremental process is executed eight times. To non-incremental clustering algorithm, the number of sequences in each clustering processes are 30 000, 40 000, 50 000, 60 000, 70 000, 80 000, 90 000 and 100 000.

For the real dataset Segmentation-all, in the stage of initial clustering, the number of sequences is 462, 66 sequences were randomly selected in each cluster. In the stage of incremental clustering, 231 sequences are increased each time. 33 sequences are randomly selected in each cluster. The incremental process is executed eight times. To non-incremental clustering algorithm, the number of sequences in each clustering processes are 693, 924, 1 155, 1 386, 1 617, 1 848, 2 079 and 2 310.

Before the experiment, all the sequences are needed to preprocess first and foremost [26]. In order to ensure comparability of results, no matter incremental clustering or non- incremental clustering, all sequences are used by the same pretreatment method.

Our experiments are run on the Intel Core 2 Duo 2.93 GHz CPU, 4 GB main memory and Microsoft XP. All algorithms are written in MyEclipse 8.5. For testing the performance of RPCAISC, we compare it with RPCAISC, DBSCAN, CURE, CARDC and RepStream in two aspects, the clustering quality and execution time. For RPCAISC, the values of parameters are set as follows: $R = 20$, $K = 7$, $\eta = 4.2$, $\theta = 0.01$, $\varepsilon = 1.5$.

4.1 Clustering Quality Test

In this section, the evaluation index of purity and entropy are applied to measure the quality of the merits of RPCAISC and the other four algorithms. The clustering purity is shown as follows.

$$Purity = \frac{\sum_{n=1}^{SC_n} Max_i(Num_{in})}{N} \tag{12}$$

where SC_n is the predefined number of clusters, Num_{in} represents the number of sequences in the n^{th} cluster that are contained in the i^{th} cluster. N is the total number of sequences in SD.

A new clustering purity will be gained after each incremental clustering. Consequently, the clustering purity which applies to incremental algorithm is given as below.

$$Purity_m = \frac{Pur_0 \times N_0 + Pur_{m-1} \times \sum_{i=1}^{m} N_i}{N_0 + \sum_{i=1}^{m} N_i} \tag{13}$$

where Pur_0 and N_0 describe the purity and the number of sequences in the stage of initial clustering, respectively. Pur_0 can be gained by Formula (11), and $N_0 = 20\,000$. N_i denotes the number of sequences for each incremental clustering. Pur_i is the purity of N_i sequences after i incremental clustering. $Purity_m$ represents the purity of all the $N_0 + \sum_{i=1}^{m} N_i$ sequences after m incremental clustering. Here, $1 \leq m \leq 8$, $N_i = 10\,000$. The other evaluation index of clustering is entropy, it is shown as follows.

$$Entropy = -\sum_{SC_i \in SC} \sum_{SC_j \in SC} PCO(SC_i, SC_j) \log PCO(SC_i, SC_j) N_i / N \tag{14}$$

where N describes the total number of sequences in SD. The probability of co-occurrence of SC_i and SC_j is denoted as $PCO(SC_i, SC_j)$.

During incremental clustering, the method for computing the entropy is as similar as purity. Entropy formulation used in this section is as follows.

$$Entropy_m = \frac{Entropy_{m-1} \times (N_0 + \sum_{i=1}^{m-1} N_i) + Entro_m \times N_m}{N_0 + \sum_{i=1}^{m} N_i} \tag{15}$$

where N_0 is the number of sequences in the stage of initial clustering. After m incremental clustering, $Entropy_m$ represents the entropy of all the $N_0 + \sum_{i=1}^{m} N_i$ sequences. $Entro_0$ describes the entropy in the stage of initial clustering. It can be gained by Formula (14). After m incremental clustering, the entropy of N_m sequences is denoted as $Entro_m$. N_i and m are the same as the corresponding values in purity.

Clustering purity and entropy of five algorithms are tested in both synthetic dataset and real dataset. The comparisons of clustering purity of five algorithms in different number of sequences are shown in Figure 3 and Figure 4. The comparisons of entropy of five algorithms in different number of sequences are given in Figures 5 and 6.

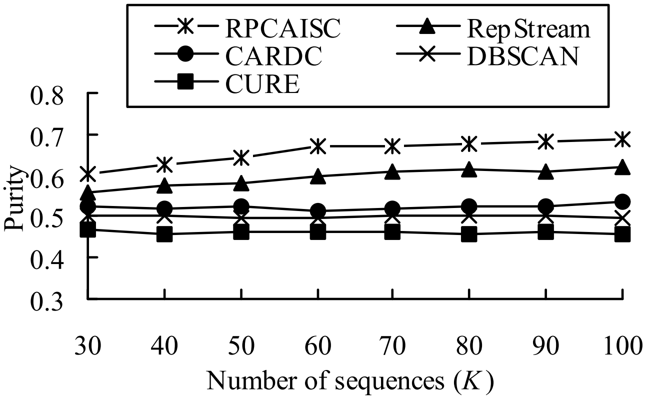

Figure 3. The comparisons of clustering purity of five algorithms in synthetic dataset

As illustrated in Figures 3, 4, 5 and 6, with the increasing number of incremental sequences, the clustering purity and entropy of RPCAISC are better than the other four algorithms DBSCAN, CURE, CARDC and RepStream. Meanwhile, the purity and entropy of RPCAISC in the real dataset is slightly better than that in the synthetic dataset.

For DBSCAN, by finding the sequences which are density-reachable to the corresponding core object, the scope of clusters becomes larger. The final clustering results are determined. The algorithm is based on the thought of density. Noises will be considered non-strong-correlation and discarded. The clustering quality will be improved to a certain extent.

Regarding the CURE, the similarity between two clusters is obtained by computing the minimum distance between two representative points in corresponding clusters, respectively. If the selection of individual representative point in cluster is not proper, it may more seriously affect the similarity between clusters. Thus it has

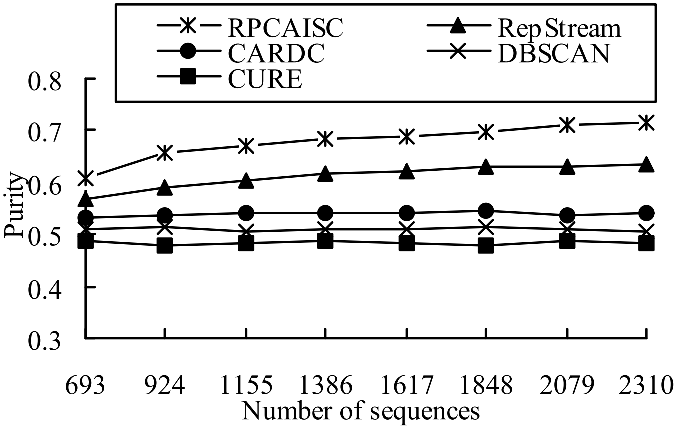

Figure 4. The comparisons of clustering purity of five algorithms in real dataset

a serious effect on the results.

However, for the three algorithms DBSCAN, CURE and CARDC, the updated dataset is clustered again every time. Therefore, if the proportion of abnormal sequence is large in incremental dataset, the clustering result of entire dataset will be seriously affected by the incremental dataset. Thus the purity and entropy of these three algorithms in both synthetic dataset and real dataset are worse than the

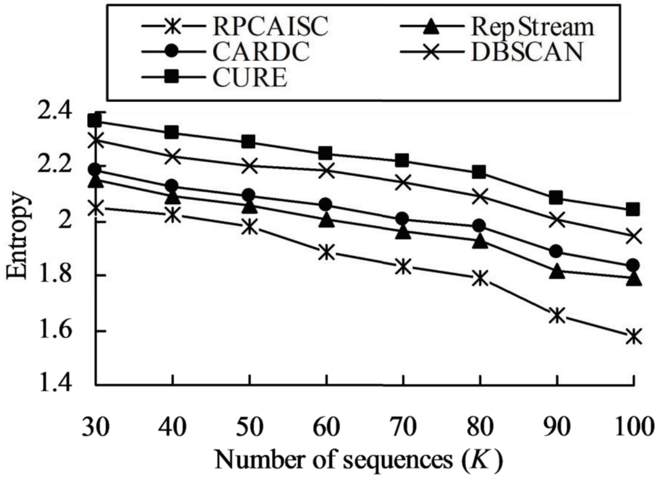

Figure 5. The comparisons of entropy of five algorithms in synthetic dataset

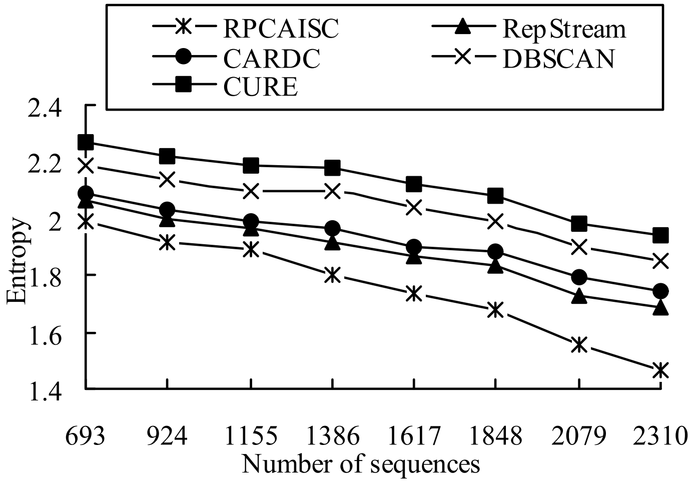

Figure 6. The comparisons of entropy of five algorithms in real dataset

other two incremental clustering algorithms.

In RepStream, the incremental sequences are only processed by incremental clustering algorithm. And the ratio of each incremental sequences number and the existed sequences number in SD is small. Thus the impact of incremental sequences for the whole dataset is also small. The purity and entropy of RepStream is better than CARDC.

The distribution densities of objects are considered synthetically by calculating density factor in RPCAISC. The cluster representative points are adjusted, and then the possibility of noise data as a representative point can be reduced. When selecting the cluster that the incremental sequence belongs to, the matching degrees between incremental sequence and each existed clustering center are computed. The cluster with the largest matching degree will be recorded. Through comparing with the change of $AV(SC_i)_{pre}$ and $AV(SC_i)_{now}$, the affiliation relationship between the incremental sequence and the corresponding cluster can be finally determined. Because the proposed matching degree is based on similarity and dissimilarity, it is better than that just using the similarity or dissimilarity hence the cluster that the incremental sequence belongs to is determined. So RPCAISC has a higher accuracy.

The Segmentation-all of UCI has 2 310 sequences and 7 clusters, while the synthetic dataset has 100 000 sequences and 20 clusters. In synthetic dataset, the number of incremental sequences is 10 000. The information which incremental sequence carried by cannot be ignored. Thus, with respect to the number of existing sequences in the cluster, the number is large. Meanwhile, the chance of appearing abnormal sequences can be increased correspondingly. So the purity and entropy of three algorithms in real dataset is slightly better than that in synthetic dataset.

4.2 Execution Time Analyzing

In synthetic dataset D100C6T14S5N001, we analyze the execution time of five algorithms in different number of sequences. Specifically, the numbers are 30 000, 40 000, 50 000, 60 000, 70 000, 80 000, 90 000 and 100 000, respectively. The experimental results are shown in Figure 7.

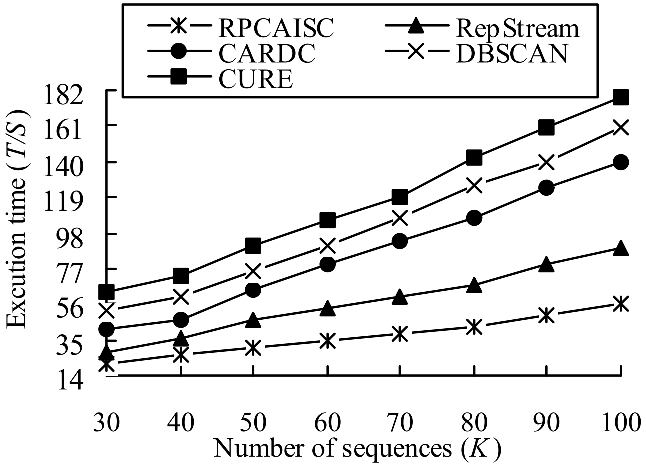

Figure 7. The comparisons of entropy of five algorithms in synthetic dataset

In real dataset, we analyze the execution time of five algorithms in different number of sequences. Specifically, the numbers are 693, 924, 1 155, 1 386, 1 617, 1 848, 2 079 and 2 310, respectively. The experimental results are shown in Figure 8.

It is difficult to see that the RPCAISC algorithm has shown better performance in real dataset and synthetic dataset from Figures 7 and 8.

For RPCAISC, in the process of merging clusters, K-nearest neighbor graph is established by the representative points within each cluster. So that the scale of the graph is small, then the proposed algorithm has the better time efficiency. Moreover, after obtaining the combined clusters, the center of SC_i, the number of sequences in SC_i, average value of SC_i, variance of SC_i, the third-order central moment and the fourth-order central moment are all seen as the cluster attributes of SC_i. The corresponding attributes of each cluster and the clustering results are saved.

When there is an incremental sequence, it is only needed to cluster with attribute vectors in RPCAISC. In addition, only attribute vectors for each obtained cluster are retained, and the time cost can be decreased effectively.

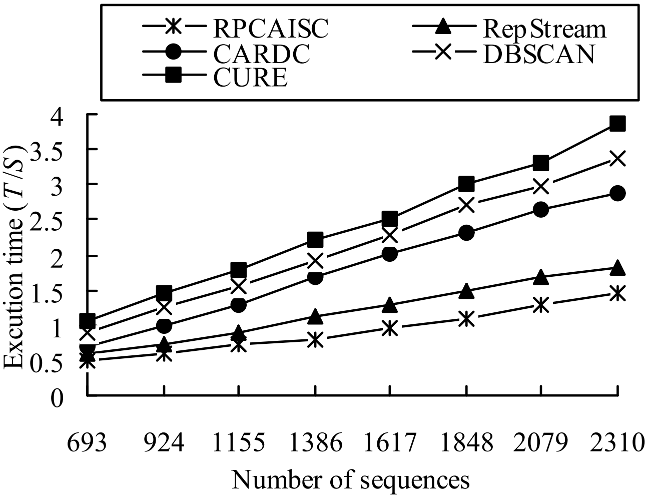

Figure 8. The comparisons of entropy of five algorithms in synthetic dataset

In CURE, two representative points Rep_i and Rep_j are selected from SC_i and SC_j, respectively. The minimum distance between Rep_i and Rep_j is viewed as the distance between SC_i and SC_j. When the fixed number of representative points in each cluster is relatively large, the time cost is not ideal.

To DBSCAN, the process of clustering is based on computing the ε-neighborhood of each sequence, if the scale of dataset is very large, it has a higher time cost.

However, because DBSCAN, CURE and CARDC are non-incremental algorithms, the updated dataset will be clustered every time. The time cost is significantly larger than in the other two incremental clustering algorithms.

In order to determine whether need to merge the two clusters, density relationship of each object in the sparse graph is used in RepStream. However, the time of constructing sparse graph takes more time. At the same time, all the attributes of cluster are selected to calculate the similarity. Therefore, the execution time is worse than using RPCAISC.

5 CONCLUSIONS

In order to solve the problem that clustering accuracy and execution efficiency of incremental sequence clustering is not ideal, in this paper, the definition of density factor, relevant degree between clusters, the matching degrees between incremental sequences and the cluster attributes vector are presented. An algorithm representative points and cluster attributes based incremental sequence clustering called RPCAISC is discussed. After preselecting cluster representative points, if the density factor is less than the corresponding threshold, then the corresponding representative point will be deleted directly. A new one will be chosen. Noises will be excluded as much as possible, and the probability that the representative point appears in the sparse area of cluster will be also reduced. Furthermore, by establishing K-nearest neighbor graph of the representative points, and computing relevant degree between clusters, whether the two clusters will be merged, is judged. The time efficiency is improved. In addition, parts of features are regarded as cluster attributes in the stage of initial clustering. Due to the proposed matching degree is based on similarity and dissimilarity, it is better than that just using the similarity or dissimilarity. Further, the cluster which incremental sequence belongs to is determined. The clustering quality is improved, and effective clustering for dynamic database can be also realized. Experimental results show that in both synthetic and real datasets, RPCAISC has higher clustering quality and better execution efficiency. In future work, we would like to concentrate on how to select better attributes to represent the cluster, and further improve the accuracy rate of clustering results. Lastly, how to keep our algorithm close to the practical application will also be considered.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 61170190), and the Nature Science Foundation of Hebei Province (F20154021-14, F2015402070). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] Song, M.—Yang, H.—Siadat, S. H.—Pechenizkiy, M.: A Comparative Study of Dimensionality Reduction Techniques to Enhance Trace Clustering Performances. Expert Systems with Applications, Vol. 40, 2013, No. 9, pp. 3722–3737, doi: 10.1016/j.eswa.2012.12.078.

[2] Chen, X. Y.—Huo, M. M.—Liu, Y.: MST-Based Semi-Supervised Clustering Using M-Labeled Objects. Computing and Informatics, Vol. 31, 2012, No. 6+, pp. 1557–1574.

[3] Hou, S. Z.—Zhang, X. F.: Analysis and Research for Network Management Alarms Correlation Based on Sequence Clustering Algorithm. Proceedings of the 2008 International Conference on Intelligent Computation Technology and Automation (ICICTA), 2008, pp. 982–986, doi: 10.1109/ICICTA.2008.263.

[4] Hoonlor, A.—Szymanski, B. K.—Zaki, M. J.—Chaoji, V.: Document Clustering with Bursty Information. Computing and Informatics, Vol. 31, 2012, No. 6+, pp. 1533–1555.

[5] Chen, W. B.—Zhang, C. C.: A Hybrid Framework for Protein Sequence Clustering and Classification Using Signature Motif Information. Integrated Computer-Aided Engineering, Vol. 16, 2009, No. 4, pp. 353–365.

[6] Małyszko, D.—Stepaniuk, J.: Adaptive Rough Entropy Clustering Algorithms in Image Segmentation. Fundamenta Informaticae, Vol. 98, 2010, No. 2-3, pp. 199–231.

[7] Song, Y. C.—Meng, H. D.—Wang, S. L.—O'Grady, M.—O'Hare, G.: Dynamic and Incremental Clustering Based on Density Reachable. Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC (NCM '09), 2009, pp. 1307–1310, doi: 10.1109/NCM.2009.376.

[8] Guha, S.—Rastogi, R.—Shim, K.: CURE: An Efficient Clustering Algorithm for Large Databases. Information Systems, Vol. 26, 2001, No. 1, pp. 35–58, doi: 10.1016/S0306-4379(01)00008-4.

[9] Zhang, J. J.—Peng, Y. W.—Li, H. F.: A New Semiparametric Estimation Method for Accelerated Hazards Mixture Cure Model. Computational Statistics and Data Analysis, Vol. 59, 2013, No. 3, pp. 95–102, doi: 10.1016/j.csda.2012.09.017.

[10] Wang, X. J.—Shen, H.: Clustering High Dimensional Data Streams with Representative Points. Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '09), 2009, pp. 449–453, doi: 10.1109/FSKD.2009.341.

[11] Delibašič, B.—Vukićević, M.—Jovanović, M.—Kirchner, K.—Ruhland, J.—Suknović, M.: An Architecture for Component-Based Design of Representative-Based Blustering Algorithms. Data and Knowledge Engineering, Vol. 75, 2012, No. 5, pp. 78–98.

[12] Cesmeci, D.—Gullu, M. K.: Phase-Correlation-Based Hyperspectral Image Classification Using Multiple Class Representatives Obtained with K-Means Clustering. International Journal of Remote Sensing, Vol. 30, 2009, No. 14, pp. 3827–3834.

[13] Feher, M.—Schmidt, J. M.: Fuzzy Clustering as a Means of Selecting Representative Conformers and Molecular Alignments. Journal of Chemical Information and Modeling, Vol. 43, 2003, No. 3, pp. 810–818.

[14] Pang, Y. J.—Pan, W.—Liu, K. D.: A Supervised Clustering Algorithm Based on Representative Points and Its Application to Fault Diagnosis of Diesel Engine. Advanced Materials Research, Vol. 121-122, 2010, pp. 958–963, doi: 10.4028/www.scientific.net/AMR.121-122.958.

[15] Chen, E. H.—Wang, S. F.—Yan, N.—Wang, X. F.: The Design and Implementation of Clustering Algorithm Using Representative Data. Pattern Recognition and Artificial Intelligence, Vol. 14, 2010, No. 4, pp. 417–422.

[16] Zhang, J. F.—Chen, L. F.—Guo, G. D.—Li, N.: Multi-Representatives-Based Algorithm for Subspace Classification. Journal of Frontiers of Computer Science and Technology, Vol. 5, 2011, No. 11, pp. 1037–1047.

[17] Araújo, D.—Neto, A. D.—Martins, A.: Information-Theoretic Clustering: A Representative and Evolutionary Approach. Expert Systems with Applications, Vol. 40, 2010, No. 10, pp. 4190–4205.

[18] Huang, T. Q.—Qin, X. L.—Wang, J. D.: Multi-Representation Feature Tree and Spatial Clustering Algorithm. Journal of Computer Science, Vol. 33, 2006, No. 12, pp. 189–195.

[19] Jia, R. Y.—Geng, J. W.—Ning, Z. Z.—He, C. G.: Fast Clustering Algorithm Based on Representative Points. Journal of Computer Engineering and Applications, Vol. 46, 2010, No. 33, pp. 121–126.

[20] Arlia, D.—Coppola, M.: Experiments in Parallel Clustering with DBSCAN. Lecture Notes in Computer Science, Vol. 2150, 2001, pp. 326–331.

[21] Meng, H. D.—Song, Y. C.—Wang, S. L.: An Incremental Clustering Algorithm Based on Subcluster Feature. Proceedings of the 2009 First IEEE International Conference on Information Science and Engineering (ICISE), 2009, pp. 786–789, doi: 10.1109/ICISE.2009.282.

[22] Lühr, S.—Lazarescu, M.: Incremental Clustering of Dynamic Data Streams Using Connectivity Based Representative Points. Data and Knowledge Engineering, Vol. 68, 2010, No. 1, pp. 1–27.

[23] Meng, F. R.—Li, X. C.—Zhou, Y.: Incremental Clustering Algorithm Based on Representative Points. Application Research of Computers, Vol. 29, 2012, No. 8, pp. 2865–2867.

[24] Chou, C. H.—Hsieh, Y. Z.—Su, M. C.: A New Measure of Cluster Validity Using Line Symmetry. Journal of Information Science and Engineering, Vol. 30, 2014, No. 2, pp. 443–461.

[25] Pham, T. T.—Luo, J. W.—Hong, T. P.—Vo, B.: An Efficient Algorithm for Mining Sequential Rules with Interestingness Measures. International Journal of Innovative Computing, Information and Control, Vol. 9, 2013, No. 12, pp. 4811–4824.

[26] Wu, D.—Ren, J. D.: K-Means Sequence Clustering Algorithm Based on Top-K Maximal Frequent Sequence Patterns. International Journal of Advancements in Computing Technology, Vol. 4, 2012, No. 20, pp. 405–413.

[27] IBM Almaden Research Center. Quest Data Mining Project [DB/OL]. (1996-03-12) [2007-05-26]. `http://www.almaden.ibm.com/cs/quest/syndata.html`.

[28] Xie, J. Y.—Guo, W. J.—Xie, W. X.—Gao, X. B.: K-Means Clustering Algorithm Based on Optimal Initial Centers Related to Pattern Distribution of Samples in Space. Application Research of Computers, Vol. 29, 2012, No. 3, pp. 888–892.

Di Wu received her B.Sc. and M.Sc. degrees in computer application technology from Hebei University of Engineering, and Ph.D. degree in computer application technology from Yanshan University. She is Lecturer at the School of Computer Science and Technology, Hebei University of Engineering, China. Her research interests include data mining, sequence clustering, software security analysis.

Jiadong Ren received his B.Sc. and M.Sc. degrees in computer application technology from the Northeastern Heavy Machinery Institute, and Ph.D. degree in computer application technology from the Harbin Institute of Technology. He is Full Professor at the School of Information Science and Engineering, Yanshan University, China. His research interests include data mining, database security, information security.