Journal of Smart Technology Applications
Vol.2, No.1 (2021), pp.9-20
http://dx.doi.org/10.21742/jsta.2021.2.1.02

# Dynamic Matrix Clustering Method Based on Time Series

Liu Lu

*Harbin Institute of Petroleum, Harbin, Heilongjiang 150027, China*
*77112428@qq.com*

***Abstract***

*Time series event clustering is the basis of research event classification and mining analysis. Existing clustering methods mostly directly cluster continuous events with time attributes and complex structures, without considering the conversion of clustering objects, and the accuracy of clustering results is low, and the efficiency is poor. To solve these problems, a dynamic matrix clustering method for time series events is proposed. Construct an event neighbor evaluation system, and construct a candidate set through the backward difference calculation strategy of neighbor scores. This paper proposes a method for selecting candidate sets of diverse sequences based on combinatorial optimization, and quickly obtains the optimal solution of the diverse sequence RDS from the candidate sets. Finally, the distance matrix between RDS and the data set is dynamically constructed, and a matrix clustering method based on K-means is proposed to realize the effective division of the categories of time series events.*

***Keywords:*** *Time series, Dynamic matrix clustering, Combinatorial optimization, K-means*

## 1. Introduction

Time series refers to a data set that changes with time. A continuous sequence is formed on the time axis, which shows the law of the internal state of the data changing with time. Nowadays, time series [1][2] are more and more widely used in various important fields such as meteorology, medical treatment, finance and network security. As the main research method, clustering analysis of time series has attracted much attention in the current academic field[2]. In the field of data mining, cluster analysis can be used not only as a link in the data mining process, but also as a tool to obtain data distribution. The k-means algorithm is a typical algorithm in clustering algorithms, which has the advantages of simple operation and low time complexity. The shortcomings of the k-means algorithm have also been exposed with people's research, mainly including three points: one is that the K value needs to be given in advance. the other is that the k-means algorithm is very sensitive to the initially selected cluster center points, and different centers the point clustering results are very different [3]. The third is that the clustering on the time series is only static clustering multiple times, and the correlation between the time series is not used for dynamic clustering.

As an unsupervised learning method, clustering [4] does not need to mark any sample in advance, and it plays a vital role in analyzing the internal relationship of data and the information and knowledge contained in it. Time series data itself has complex structural

**Article history:**
Received (November 16, 2020), Review Result (December 21, 2020), Accepted (January 8, 2021)

eISSN: 2652-9807 JSTA

features such as pseudo-events, persistence and drift. Existing methods mostly directly cluster the continuous events with complex structure in the data set without transforming the clustering objects, and the accuracy of the clustering results is low and the efficiency is poor. Therefore, how to design an accurate and efficient time series event clustering [7] method has always been a difficult point in the field of streaming data mining [5] To solve these problems, this paper proposes a dynamic matrix clustering method RDMC (RDS Dynamic Matrix-based Clustering) for time series events. By accurately and efficiently constructing the distance matrix between RDS (Representative and Diversifying Sequences) and the data set, the clustering of the original data set is transformed into the clustering of the dynamic matrix, and the effective division of time series events is realized. [Figure 1] is a schematic diagram of the structure of the RDMC method.

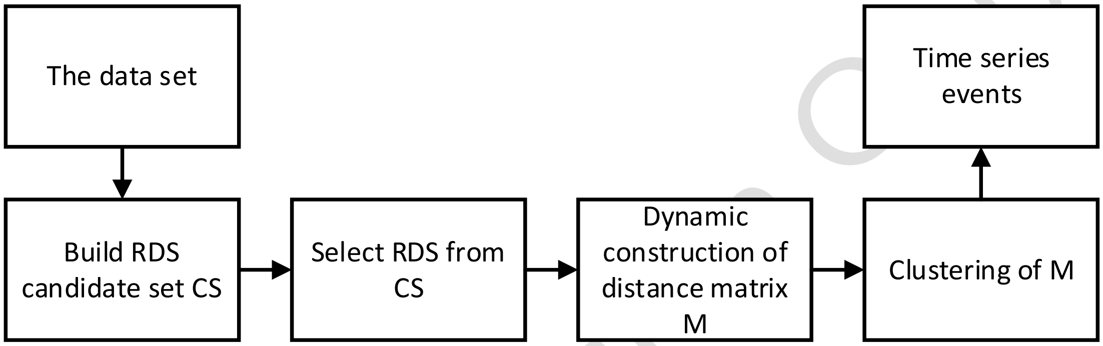

The data set
Time series events
Build RDS candidate set CS
Select RDS from CS
Dynamic construction of distance matrix M
Clustering of M

Figure 1. The schematic diagram of the structure of the RDMC method

## 2. Research status

At present, many experts and scholars have conducted in-depth research on time series event clustering methods and have achieved certain research results.

Euán C et al. [6] proposed a clustering method based on spectral density and total variational distance to construct clusters through similar oscillation behavior of events. This method does not transform the original data set, and directly clusters time series events. Affected by the complex structure of the events, the clustering accuracy is low. Azencott R et al. [7] proposed an automatic clustering method on large-scale time series data. The stochastic gradient descent method minimizes the cost function that measures the quality of clustering, thereby automatically generating optimal clusters. Because the optimization process of the cost function takes too long, the clustering efficiency of this method is poor. Zheng Jianwei et al. [9] proposed a combined Laplacian regularization term and adaptive feature learning method, which can perform feature selection and data clustering at the same time, but this method requires manual setting of multiple parameters, and different values of the parameters directly affect the aggregation. The accuracy of the class result. Zhang Dongyue et al. [8] proposed GC Stream, a clustering method based on grid coupling and core grid. The correlation between data can be expressed more accurately through grid coupling, but the grid established by this method requires equal side lengths, which is not suitable for time series data with complex structure. ZAKARIA J [10] and others proposed a method of clustering using characteristic local shapes (U-Shape lets) in the data set. The method has high accuracy. However, the time complexity of extracting U-Shape lets is too high, resulting in the clustering efficiency is extremely low. Madrague NS [11] and others proposed a deep temporal clustering method DTC, which uses an autoencoder to reduce the time dimension and performs cluster assignment through a novel temporal clustering layer. When iteratively training the temporal clustering layer, the optimization goal is to minimize the KL divergence loss of the prediction distribution and the target distribution. The target distribution is calculated by the prediction distribution and updated at each iteration, which leads to instability of the method and clustering the accuracy fluctuates greatly. This paper conducts an in-depth study on time series event clustering. Aiming at the shortcomings of the existing clustering methods, comprehensively considering the complex structural characteristics of events, a RDS-based time series event clustering method is proposed, which can improve the clustering efficiency while ensuring the accuracy of clustering.

This paper conducts an in-depth study on time series event clustering. Aiming at the shortcomings of the existing clustering methods, comprehensively considering the complex structural characteristics of events, an RDS-based time series event clustering method is proposed, which can improve the clustering efficiency while ensuring the accuracy of clustering.

## 3. Selection of diverse sequence candidate sets

A time series event is a series of discrete data points that meet certain conditions in the time domain. Events of the same type usually have the same or similar characteristics. A method for constructing an event's neighbor evaluation system is proposed, the representativeness of the event is measured according to the merits of the evaluation value, and the backward difference method is proposed to determine the boundary value of the neighbor score. The RDS candidate set is constructed through the relationship between the boundary value and the evaluation value.

### 3.1. Construction of an event neighbor evaluation system

The concept of RDS and its candidate set is proposed, and an event neighbor evaluation system is constructed based on the event neighbor density and the reverse neighbor number, and the representativeness of any event in the data set is accurately measured by the evaluation value. RDS definition 1 is shown.

Definition 1 (RDS) Given a time series event data set $\mathrm{T} = \{\mathrm{t}_1, \mathrm{t}_2, \dots, \mathrm{t}_n\}, \mathrm{t}_i(1 \leq \mathrm{i} \leq \mathrm{n})$is a continuous event composed of multiple discrete point sets. RDS is the k (1≤k≤T) events selected from T that can represent different clusters to the maximum extent with the greatest difference, and one event is recorded as $\mathrm{RDS}_i(1 \leq \mathrm{i} \leq \mathrm{k})$. Representing different clusters to the maximum means that $\mathrm{RDS}_i$ has the greatest similarity with different clusters. The largest difference means that the similarity between multiple $\mathrm{RDS}_i$ is smallest. [Figure 2] is a schematic diagram of the definition of RDS. The data set T is divided into four clusters C1~C4. The events contained in each cluster are shown by the small blue circles. The best RDS in T is shown by the red circles. Among them, the distance d1 between $\mathrm{RDS}_1$ and any event in the cluster C1 is small, which makes it have the greatest overall similarity with C1, that is, it can represent C1 to the maximum. in the same way, $\mathrm{RDS}_2$ and $\mathrm{RDS}_3$ can represent C2 and C3 to the maximum. The distances d4 to d6 between $\mathrm{RDS}_1$ and $\mathrm{RDS}_3$ are relatively large, that is, while they maximize the representation of different types of events in T, they also satisfy the greatest difference between each other.

The RDS selection strategy affects the efficiency of event clustering, and the cost of directly selecting from the data set T is too high, so the candidate set is constructed first, and then k events with the largest representation and difference at the same time are found from the candidate set to form the RDS.

The RDS candidate set C represents a set of representative m events selected from the data set T. That is, $\mathrm{C} = \{\mathrm{t}_1, \mathrm{t}_2, \dots, \mathrm{t}_m\}, \mathrm{t}_i (1 \leq \mathrm{i} \leq \mathrm{m})$ represents an event, and $\mathrm{k} \ll \mathrm{m} \ll \mathrm{n}$， , k is the number of events contained in the RDS, and n is the size of T. The selected event must have a high similarity to a certain type of cluster, but there is no requirement for large differences between multiple events.

Most events have complex structural features such as drift. In order to accurately measure the similarity between them, a DTW distance measurement method is given.

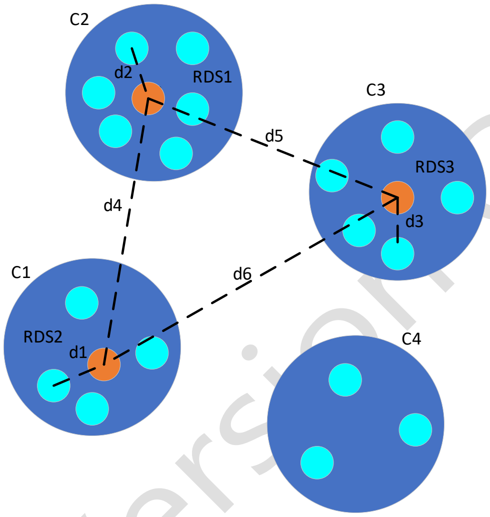


Figure 2. The schematic diagram of the definition of RDS

If there are two time series events t1 and t2 with lengths |t1| and |t2| respectively, record the regularization path $\mathrm{W} = \{w_1, w_2, \dots w_r\}$ . Where max(|t1|,|t2|)<=r<=|t1| |t2|, $w_r$ has the form (i,j), i is an element in t1, and j is an element in t2. Let the cost matrix of t1 and t2 be D, then a path in D is formula 1:

$$D(i,j) = ED(i,j) + min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases} \tag{1}$$

Among them, $ED(i,j)$ is the Euclidean distance between i and j, and finally the DTW distance between t1 and t2 is obtained as formula 2:

$$d(t1,t2) = \ D(|t1|,|t2|) \tag{2}$$

In the following, an event neighbor evaluation system is first constructed, and the representativeness of events in the data set T is measured by the evaluation value. The neighborhood of event r is defined as follows.

Definition 2 (r-neighborhood) Given a data set T, the r-neighborhood $NNr\ (t)$ of an event t is defined as the set of r events that are closest to t, that is, formula 3:

$$NNr(t) = \{x | d(t, x \leq dr(t)\} \tag{3}$$

Among them, $d(t,x)$ represents the DTW distance between t and event x, and $dr(t)$ represents the rth nearest neighbor DTW distance between t and other events. For an event t, the neighbor density of t is used to quantify the overall similarity between t and the events in its r neighborhood.

Definition 3 (Nearest Neighbor Density) Given the r-neighborhood $NNk\ (t)$ of an event t, the nearest neighbor density P(t) of t is defined as formula 4:

$$P(t) = \frac{1}{\sum_{x \in NN_r(t)} d(x,t)} \qquad (4)$$

The larger P(t) is, the higher the overall similarity between t and all events in its r neighborhood is. on the contrary, the lower the similarity. In addition, the reverse neighbor count can measure the similarity between event t and other events from a reverse perspective.

Definition 4 (Reverse nearest neighbor number) The reverse nearest neighbor number $Nb(t)$ represents the total number of times that event t is neighbors of other events in the process of calculating the r nearest neighbor distance for each event in T. That is: $x \in T$, if $t \in NNr\ (x)$, then $b(t) = \ Nb(t) + 1，\ 1 \leq r < n$。

The larger $Nb(t)$ is, it means that t is the neighbor of more events. Inversely, it means that t is closer to more events, that is, the similarity is higher. Based on the density of neighbors and the number of inverse neighbors, the concept of event neighbor score is proposed, and the overall similarity between event t and other events is comprehensively measured from both positive and negative aspects.

Definition 5 (Nearest Neighbor Score) Given the number of reverse neighbors Nb(t) and neighbor density P(t) of a given event t, the neighbor score $S(t)$ of t is the ratio of $ln(Nb(t) + 1)$ and $P(t)$ The product, that is, formula 5:

$$S(t) = \frac{ln(Nb(t)+1)}{\sum_{x \in NN_r(t)} d(t,x)} \qquad (5)$$

In the formula, the scale of $Nb(t)$ is reduced by taking the logarithm to make it change more smoothly. Since $Nb(t)$ may be 0, add 1 to it to avoid the negative value of the logarithm calculation, that is to make $ln(Nb(t)\ + 1) \geq\ 0$. The formula shows that, given the r-neighborhood $NNr(t)$ of t, if the number of reverse neighbors $Nb(t)$ and the neighbor density $P(t)$ of t are larger, the larger S(t) is, indicating that t and more events Similar, that is, t is a representative event.

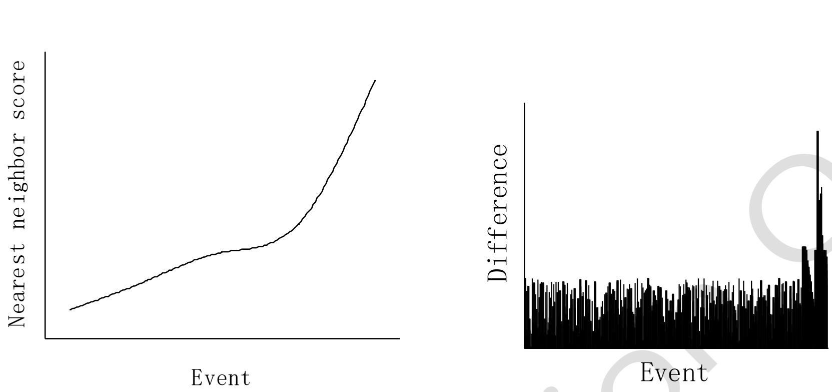
### 3.2. RDS candidate set construction

Based on the neighbor evaluation system, a backward difference calculation strategy for neighbor scores is proposed, and the RDS candidate set is quickly screened based on the difference calculation results. The number of reverse neighbors and the density of neighbors of different events are quite different, so that the calculated value of neighbor score is not on the same scale, and it needs to be normalized to the interval of [0,1]. Take the Plane data set from the UCR database as an example, calculate the neighbor scores for all events in the data set, normalize and sort, and obtain the neighbor score schematic diagram in [Figure 3(a)].

As shown in the figure, the neighbor scores of events in candidate set C are higher than normal events, so the events in C are all located in the high-scoring area, and the normal events are in the low-scoring area, and at the boundary between the two, the jump range of neighbor scores Big. Based on this, this paper proposes a backward difference calculation strategy for neighbor scores. The boundary is determined by comparing the size of the difference value and the candidate set is constructed. The backward difference calculation formula is as follows:

$$S(t) = S(t) – S(t-1) \tag{6}$$

Among them, ▽S(t) is the difference operator, $S(t)$ and $S(t-1)$ are two adjacent events. Figure 3(b) shows the backward difference calculation result of the neighbor score. It can be seen that the point of the maximum difference is located at the position where the score has a big jump in [Figure 3(a)].



(a) Nearest neighbor score (b) difference

Figure 3. Schematic diagram of the r-nearest neighbor score and its difference

The specific process of RDS candidate set C construction is shown in Algorithm 1.

```
Algorithm 1: Construct C algorithm
Input: data set T. threshold parameter φ
Output: RDS candidate set result C
(1) Initialization: $r = 0$, $N(r) = 0$.
(2) Calculate the r-th nearest neighbor w for each event in T.
(3) $Nb(w) \leftarrow Nb(w) + 1$, $NN_r(t) \leftarrow NN_r(t) \cup \{w\}$.
(4) Calculate t's neighbor score $S(t)$ according to the formula.
(5) If $Nb(t) = 0$, then $N(r) = N(r) + 1$.
(6) $(N(r-1) – N(r))/N(r-1) < \varphi$,, go to step 7. otherwise, let $r = r + 1$, go to step 2.
(7) Normalize $S(t)$ and Sort.
(8) Calculate the backward difference D of S(t) according to the formula, and record the index value of the maximum value as max.
(9) If $S(t) > S(max)$, then $t \in C$, let $C = C \cup t$, Otherwise, $t \neq C$.
(10) Return candidate set result C.
```

First, in the process of constructing all event neighborhoods in T, calculate the r-th nearest neighbor sequence W for each event t, add 1 to the reverse neighbor number Nb(w) of W, and add W to the r neighbor of t Domain NNr(t), calculate the nearest neighbor score S(t) of t. At this time, count the number of reverse neighbors of the event in T equal to the number N(r) of 0, if the decline rate is less than the given threshold ∮ (0≤ ∮ ≤1, generally a number between [0, 0.1]) , The domain construction is completed and the iteration ends, otherwise, continue to construct the r 1st domain. as shown in Figure 4, at the beginning, as r increases, N(r) decreases rapidly. when r increases to a certain value, N (r) The decreasing speed tends to be flat and eventually decreases to zero. When N(r) decreases gently, r gets the best cut-off value. Take a certain event t as an example. If it is increased again, events far away from t will be mistakenly added to the neighborhood of t. Improper establishment of neighbor relationships will make the neighbor scores unable to correctly measure the representativeness of t. secondly, perform differential calculation on the normalized and sorted neighbor score S, the maximum index in the calculation result D is marked as Max. finally, the events on the Max side belong to the RDS candidate set, add them to C, and return to C.

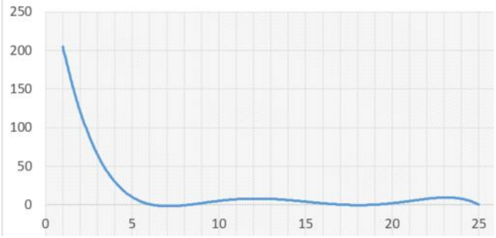


Figure 4. Schematic diagram of the change trend of N

## 4. Time series time clustering

On the basis of the candidate set C, the optimal selection method of RDS based on combinatorial optimization is proposed, the distance matrix between RDS and the data set is dynamically constructed, and the matrix clustering method based on K-means is proposed to realize the classification of events.

### 4.1. Optimal RDS solution selection

It can be seen from Definition 1 that RDS needs to meet the two conditions of "similarity" and "difference" at the same time, so the selection of RDS is a multi-condition constraint optimal solution problem, which can be solved by a combination optimization method, and all the candidates in the candidate set are calculated based on the neighbor score. The quality index of the feasible solution, the optimal solution is obtained by comparing the size. Let a combination of any k events in C be a feasible solution of RDS, all feasible solutions constitute a set F, for a specific feasible solution, the mean score of k events in f is denoted as u, and the mean distance between them Denoted as v.

$$u = \frac{1}{k}\sum_{t\in f} S(t) \tag{7}$$

$$v = \frac{1}{k(k-1)}\sum_{x,y\in f} d(x,y) \tag{8}$$

u, v are not under the same scale, normalize them to the interval [0, 1] to make the two comparable. In addition, in order to accurately quantify the selection quality of feasible solutions f, this paper proposes the concept of quality indicators.

Definition 6: Quality Index is used to simultaneously measure the dual constraint conditions of a feasible solution f of RDS. The calculation formula is as follows:

$$QI(f) = \alpha u + (1-\alpha)v \tag{9}$$

Among them, 0≤α≤1, and the coefficient α is used to balance the proportion of u and v. By default, α=0.5. The larger u,v, the larger QI(f), which indicates that the greater the "similarity" and "difference" of f, the higher the selection quality. When QI(f) is maximum, RDS obtains the optimal solution o, the formula is as follows

$$o = \arg\max QI(f),\ f \in F \tag{10}$$

In summary, the process of selecting the optimal RDS solution is as follows:

Algorithm 2: Extract RDS algorithm

Input: data set T. threshold parameter α . RDS size k

Output: RDS optimal solution o

(1) The candidate set C is obtained by algorithm 1.

(2) Take all feasible solutions of k events from C as F.

(3) For each feasible solution f, calculate u, v according to formula (5-6), and normalize them.

(4) Calculate the QI of f according to the formula

(5) Obtain the optimal solution o according to the formula.

## 4.2. Matrix clustering based on K-means

The k-means algorithm is an indirect clustering method based on the similarity measurement between samples, also known as the K-means algorithm. The main idea of the algorithm is to divide the data set into different categories through an iterative process, so that the criterion function for evaluating clustering performance is optimized. The algorithm is described as follows:

Input: the number of clusters K and the data set ξ.

Output: K clusters, each data belongs to category label.

(1) Randomly select k points as the center of the initial cluster category.

(2) Allocate the data in the data set to the class closest to the center point.

(3) Use the mean of the sample data in each cluster as the new cluster center.

(4) Repeat steps 2) and 3) until the algorithm converges.

(5) End, get K result clusters.

On the basis of the RDS selected in Section 4.1, the distance matrix between the RDS and the data set is dynamically constructed to transform the clustering object, and the matrix clustering method based on K-means is proposed, and the category label of the event is obtained according to the matrix clustering result.

The form of the distance matrix A between RDS and the data set is shown below.

$$A = \begin{matrix} m_{11} & m_{12} & \cdots & m_{1f} \\ m_{21} & m_{22} & \cdots & m_{2f} \\ \vdots & \vdots & \vdots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nf} \end{matrix}$$

The size of M is [n,i], $i \le k$, which represents the number of events actually participating in clustering in RDS, n represents the number of all events in the data set T, and the elements $m_{ji}$ in M represent the distance between RDSi and event j in T.

In addition, M is not directly formed by the distance vector between RDS and the data set, but is gradually expanded and dynamically constructed. Its purpose is to further improve the accuracy and efficiency of clustering. After calculating the distance vector between RDSi and the data set, add this vector as a column of matrix M and extend it horizontally. According to the current clustering purity, decide whether it is necessary Continue to expand M and perform clustering operations. The calculation formula of cluster purity is as follows.

$$Pur(C,\Omega)=\frac{1}{n}\sum_{k}\max_{j}\left|w_k \cap c_j\right| \tag{11}$$

Among them, n represents the total number of events, and $C=\{c_1,c_2,\cdots c_j\}$ represents the real cluster division。

$\Omega=\{w_1,w_2,\cdots w_k\}$ Indicates the cluster division of the current cluster。

The specific process of the matrix clustering method based on K-means is shown in Algorithm 3

```
Algorithm 3: Clustering algorithm
Input: data set T. threshold parameter. RDS size k1. number of event categories k2
Output: Event category labels
(1) Initialization: i=0, iters=1, MaxIters=50.
(2) Obtain the optimal solution R of the candidate set RDS by algorithm 2.
(3) While i<k:
(4) Calculate the distance vector dist between RDSi and T.
(5) Dist = [Dist dist].
(6) [Labels Centroids] = K- means(Dist, k2). iters++.
(7) If iters<MaxIters, go to step 6.
(8) CP(i)=1-Pur (Labels (i-1), Labels(i )).
(9) If CP (i)< τ , go to step 11.
(10) i++.
(11) Return the clustering result Labels(i).
```

First, construct the candidate set C and get the RDS optimal solution R. second, perform the clustering process iteratively (lines 4-17), when i<k, calculate the distance vector dist between RDSi and T and add it to Dist horizontally Where Dist is the distance matrix between the first i events in RDS and T at this time. Use K-means to cluster Dist. In order to make the clustering results more accurate, K-means was run multiple times. Remember the change in the cluster purity Pur as CP (Change in Pur), assuming that the cluster label obtained in the i-1th cycle is correct, then the formula can be obtained, the corresponding CP(i) in the i-th cycle =1-Pur (Labels (i-1), Labels(i )). When CP is less than τ ,( 0⩽1 ⩽ τ , generally a number between [0, 0.1], the cycle ends. As the number of RDSs participating in the construction of the matrix Dist increases, Dist contains more detailed events relative to each other The distance information prompts the RI to show an upward trend. When the CP is small, it indicates that the best clustering result has been obtained at this time, and there is no need to continue the clustering process. finally, the event category label obtained through the clustering is returned.

## 5. Experimental analysis

The experimental platform is Intel(R) i5-6500 3.20 GHz processor, 4GB memory, and Window 10 (64bit) operating system. Use six data sets selected from the UCR public database to conduct experiments, as shown in Table 1. The accuracy of clustering verifies the effectiveness of the RDMC method proposed in this paper. The comparison methods are K-means, DBSCAN, Spectral, U-Shape lets, etc. The threshold parameter $\tau$ is set to 0.05, $\varphi$ is set to 0.07 on the two larger data sets Ford A and Starlight Curves, and $\varphi$ is set to 0.03 on other small-scale data sets.

Table 1. Experimental data set

| Data set | Data set size | Length of time | Number of categories |
|---|---|---|---|
| 50Words | 905 | 270 | 50 |
| Cricket_Z | 780 | 300 | 12 |
| Gun-Point | 200 | 150 | 2 |
| Plane | 210 | 144 | 7 |
| FordA | 4921 | 500 | 2 |
| StarLightCurves | 9236 | 1024 | 3 |

### 5.1. Comparison of clustering accuracy

This section evaluates the clustering accuracy of the RDMC method on the benchmark data set in Table 1, using RI (Rand index) as the evaluation index. The RI calculation formula is as follows:

$$RI = \frac{TP+TN}{TP+FP+FN+TN} \tag{12}$$

Among them, TP represents the number of events that belong to the same category and are classified into the same cluster, TN represents the number of events that belong to different categories and are classified into different clusters, and FP represents the number of events that belong to different categories but are classified into the same cluster. FN represents the number of events that belong to the same category but are divided into different clusters. $0 \leq RI \leq 1$, the larger the RI value, the higher the clustering accuracy.

## 6. Conclusion

This paper has carried out an in-depth study on time series event clustering. Aiming at the problems of low clustering accuracy and poor efficiency of existing clustering methods, a dynamic matrix clustering method for time series events is proposed. First, an event neighbor evaluation system is proposed to achieve an effective measurement of event representativeness. the score is defined according to the backward difference calculation strategy of neighbor scores to achieve rapid construction of the RDS candidate set, thereby improving the efficiency of RDS selection. second, a combination optimization based on the selection method of RDS realizes the efficient and optimal search of RDS. finally, for the problem of low accuracy of direct clustering of time series events, on the basis of dynamically constructing the distance matrix between RDS and the data set, a K-means-based the matrix clustering method realizes the correct division of the categories of events. Experiments verify the effectiveness of the proposed method.

## References

[1] Li Hailin and Guo Chonghui, “Research review of feature representation and similarity measurement in time series data mining,” Computer Application Research, vol.30, no.5, pp.1285-1291, **(2013)**

[2] Yang Yiming, Pan Rong, Pan Jialin, etc., “Algorithm comparison of time series classification,” Chinese Journal of Computers, vol.30, no.8, pp. 1259-1266, **(2007)**

[3] Cong Si'an and Wang Xingxing, “Overview of K-means algorithm research,” Electronic Technology and Software Engineering, no.17, pp.155-156, **(2018)**

[4] Chen Chaowei and Chang Dongxia, "Automatic clustering algorithm based on density difference," Journal of Software, vol.29, no.4, pp.935-944, **(2018)**

[5] Shu K., Sliva A., Wang S, et al., "Fake news detection on social media: A data mining perspective," ACM SIGKDD Explorations Newsletter, vol.19, no.1, pp. 22-36, **(2017)**

[6] Euán C., Ombao H., and Ortega J., "The hierarchical spectral merger algorithm: a new time series clustering procedure," Journal of Classification, vol.35, no.1, pp.71-99, **(2018)**

[7] Azencott R., Muravina V., Hekmati R., et al., "Automatic clustering in large sets of time series," Contributions to Partial Differential Equations and Applications. Springer, Cham, pp.65-75, **(2019)**

[8] Zhang D. Y., Zhou L. H., Wu X. Y., and Zhao L. H., "Data stream clustering based on grid coupling," Journal of Software, vol.30, no.3, pp.667-683, **(2019)**

[9] Zheng J. W., Li Z. R., Wang W. L., and Chen W. J., "Clustering with joint Laplacian regularization and adaptive feature learning," Journal of Software, vol.30, no.3, pp.3846-3861, **(2019)**

[10] Zakaria J., Mueen A., and Keogh E., "Clustering time series using unsupervised-shape lets," IEEE 12th International Conference on Data Mining, pp.785-794, **(2012)**

[11] Madiraju N.S., Sadat S.M., Fisher D., and et al., "Deep temporal clustering: Fully unsupervised learning of time-domain features," arXiv preprint arXiv:1802.01059, **(2018)**

***This page is empty by intention.***