Engineering Applications of Artificial Intelligence 89 (2020) 103451

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Self-adaption neighborhood density clustering method for mixed data stream with concept drift[☆]

Shuliang Xu[a], Lin Feng[b,*], Shenglan Liu[b], Hong Qiao[c,d]

[a] Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, 116024 Dalian, China
[b] School of Innovation and Entrepreneurship, Dalian University of Technology, 116024 Dalian, China
[c] Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
[d] State Key Laboratory for Management and Control of Complex Systems, Beijing, 100190, China

ARTICLE INFO

Keywords:
Data stream
Concept drift
Rough set
Clustering analysis
Neighborhood entropy

ABSTRACT

Clustering analysis is an important data mining method for data stream. In this paper, a self-adaption neighborhood density clustering method for mixed data stream is proposed. The method uses a significant metric criteria to make categorical attribute values become numeric and then the dimension of data is reduced by a nonlinear dimensionality reduction method. In the clustering method, each point is evaluated by neighborhood density. The k points are selected from the data set with maximum mutual distance after k is determined according to rough set. In addition, a new similarity measure based on neighborhood entropy is presented. The data points can be partitioned into the nearest cluster and the algorithm adaptively adjusts the clustering center points by clustering error. The experimental results show that the proposed method can obtain better clustering results than the comparison algorithms on the most data sets and the experimental results prove that the proposed algorithm is effective for data stream clustering.

1. Introduction

With the development of information society, many fields continue to generate massive data streams such as online shopping, satellite remote sensing, weather forecast and traffic flow monitoring etc. Different from the traditional static data, data stream often has the characteristics of unlimited number, rapid arrival and concept drift which make data stream mining face prodigious challenges (Babcock et al., 2002; Krishnaswamy, 2005; Xu and Wang, 2017, 2016). In practical applications, the data needing to be analyzed is often unlabeled and it will have a large cost to obtain the class labels for data stream. Therefore it is valuable to develop a data stream clustering algorithm. Data stream clustering has gained wide attention from researchers and there are many remarkable achievements (Silva et al., 2013; Ding et al., 2015; Kaur et al., 2015). Up to now, the most existing data stream clustering algorithms can be classified into two types according to the type of data set: clustering algorithms for categorical data and numerical data. For the categorical data set, simple matching distance is used to measure the similarity of two data points. Bai et al. (2016) propose an optimization model for clustering categorical data stream; for a data stream model, the proposed algorithm continuously updates the clustering parameters to make the error between the current clustering model and the previous clustering model as small possible; the parameters can be solved by EM algorithm. In order to detect concept drift, a new measure is defined. If the value of the new measure is greater than a predefined threshold, concept drift can be detected. Jiang and Brice (2009) present a Context-Trees algorithm for categorical data stream; the algorithm expands existing clustering techniques for static categorical data sets to predictive models of data streams based on variable length Markov models of clusters. The stored clusters along with the distributional information can be used to create and analyze aggregated clusters over user-specified time intervals to detect the changes of data streams. Maji and Pal (2007) develop a rough-fuzzy C-Medoids algorithm for amino acid sequence analysis; it uses rough fuzzy set to decide which cluster the data point belongs to; the decision can be derived from the upper approximation set and lower approximation set. Cao et al. (2010) propose a framework for clustering categorical time-evolving data; the algorithm employs the uncertainty of rough set to define the membership function of fuzzy set; the distance of between a data point and a cluster and the distance of two clusters are also defined based on the membership function. In addition, the algorithm can create a graph to visual concept drift. Li et al. (2014) propose an incremental entropy clustering algorithm for categorical data stream; the dissimilarity between a data point and

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.engappai.2019.103451.

[*] Corresponding author.
E-mail addresses: xushulianghao@126.com (S. Xu), fenglin@dlut.edu.cn (L. Feng).

https://doi.org/10.1016/j.engappai.2019.103451
Received 19 October 2018; Received in revised form 10 November 2019; Accepted 23 December 2019
Available online 16 January 2020

a cluster can determined by incremental entropy; the algorithm can autonomously determined the threshold of the dissimilarity distance which is an advantage of the incremental entropy clustering algorithm; both concept drift and outliers can be detected by the algorithm.

For the problem of clustering numeric data stream, there are many related work. Shindler et al. (2011) propose an fast and accurate *k*-means algorithm for large data set; when handling data points, the distance of the current point to the nearest the cluster centering is computed to determine whether generating a new cluster or partitioning to the nearest cluster; the greater the distance is, the higher the probability of generating new class clusters will be; after the number of clusters exceeding a value, the parameters will be adjusted to reduce the number of clusters. Cao et al. (2006) present a density-based clustering algorithm called as *DenStream*. DenStream is a development of *DBSCAN* algorithm and the decision that the data point is a outlier cluster or belongs to a micro cluster is made by the statistical information of the current data; the outdated cluster will be deleted from the system and a new cluster can also be generated if the discriminant condition is satisfied. Chen and Tu (2007) propose a density-based clustering method for real-time data stream called as *D-Stream*. In D-Stream, density-based clustering algorithm and grid-based clustering algorithm are integrated; the grid is divide into three types: dense grid, sparse grid and other type of grid. The three types of grid can be converted to each other by density; finally, clusters can be generated by merging grids and deleting outdated grids. Hahsler and Bolanos (2016) develop a shared density clustering algorithm called as *DBStream*; for a data point, if the number of the neighbors is lower than a threshold, the data point is seen as outlier cluster, otherwise it updates the micro clusters of the neighbors and the micro clusters of shared regions; then in offline phase, the graph of shared density is constructed and the data set is clustered again. Zhang et al. (2014a) present a data stream clustering algorithm with affinity propagation called as *STRAP*; for each cluster, the algorithm utilizes a four tuples to represent the statistical information of cluster; firstly, *STRAP* uses *AP* algorithm (Frey and Dueck, 2007) to produce initial clusters; after obtaining a data point, *STRAP* selects a cluster with minimum distance between the data point and a cluster; if the distance is less than the predefined threshold, the data point belongs to the cluster; otherwise the data point is added into the buffer; concept drift can be detected by PH test; if the PH assumption is violated, concept drift has happened and the clustering model is updated.

From the above summarization, it is obvious that the situation that the algorithms can only deal with single type of data stream restricts the applications of clustering algorithms. Mixed data stream is common in practical applications. For example, in medical data analysis, some biochemical indexes such as *ALT*, *PCT* and *WBC* etc. can be measured by numerical values and some clinical symptoms such as cough, headache and palpitation etc. can be expressed by categorical characteristic values. On the other hand, simple matching distance or Minkowski distance cannot effectively measure the similarity of data points for mixed data stream. Therefore a self-adaption neighborhood density clustering method for mixed data stream with concept drift (SNDC) is proposed in this paper. *SNDC* employs a significance criterion to evaluate categorical attribute values and makes categorical attribute values become numeric values at first, then a nonlinear dimensionality reduction algorithm based on neighborhood similarity is presented to reduce the complex of data set. In the clustering phase, the clustering center points can be automatically adjusted according to clustering result. The weight of each cluster will be decayed with time and outdated clusters can be deleted from the system. For *SNDC* method, concept drift can be detected by the similarity of adjacent data block clusters. The main contributions of this paper are as follows:

- A mapping method which makes categorical attribute values become numeric values is introduced. After executing the method, categorical attribute values can be replaced by the significance values which ensures that the data can be further processed.
- In order to reduce the complexity of data set, a nonlinear dimensionality reduction method based on neighborhood similarity is presented. Neighborhood similarity decreases the effect of geometric spatial structure on similarity measurement.
- A self-adaption neighborhood density clustering method is proposed. The method can automatically select the best initial clustering center. The center points can be adjusted according to the clustering result. By comparing the similarity of the clusters of adjacent data block, concept drift can be detected and a series of measures are taken to adapting to new concept in data stream.

The rest of this paper is organized as follows: Section 2 reviews some background knowledge and brief introductions about data stream model, rough set and neighborhood rough set are given; Section 3 introduces a nonlinear dimensionality reduction algorithm based on neighborhood similarity; Section 4 explains the details of *SNDC* method in clustering phase; in Section 5, the experiments are performed to show the effectiveness of SNDC algorithm; Section 6 concludes the paper and gives some future research directions.

2. Backgrounds

In this section, we will introduce the basic model of data stream and then some fundamental concepts of rough set and neighborhood rough set are explained. The uncertainty measure methods of rough set and neighborhood rough set are also presented.

2.1. Data stream model

Let a data stream $\boldsymbol{S} = \left\{\dots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \dots\right\}$ where $\boldsymbol{x}_t$ is a data point generated at t moment and $\boldsymbol{x}_t \in \mathbb{R}^{1\times m}$. In the data stream environment, it faces with massive data. The space needed to store data points is far beyond the capacity of memory. In order to effectively deal with massive data, a sliding window mechanism is generally adopted to divide data stream into many data blocks, and only one or several data blocks are allowed into memory at each moment. After processing the data block in the current window, the data block of the next window can be allowed to enter the memory. To facilitate the description of data stream clustering problem, there are following definitions (Xu and Wang, 2017, 2016).

Definition 1. Several data points are organized into a data set according to their time sequence and the data set is named as data block which is denoted as $\boldsymbol{B}_i = \left\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\right\}$, where $\boldsymbol{x}_i \in \mathbb{R}^{1\times m}, i = 1, 2, \dots, n$ and m is the number of data points in a data block.

From Definition 1, it is known that sliding window mechanism is an effective way to deal with massive stream data. Massive stream data is divided into large number of data blocks and data blocks can be handled in the memory which makes big data analysis become possible.

Definition 2. Let the data model be M at t moment, after Δt time, the data model changes to N; when $M \neq N$, it is said that concept drift has happened during this period. If Δt is a long period, concept drift is called as gradual concept drift; if Δt is a short period, concept drift is named as abrupt concept drift.

A discriminant criteria of concept drift is presented in Definition 2. Actually, when concept drift appearing, the distribution of data in data stream has changed, therefore concept drift can be detected by comparing the similarity of data distribution in different sliding windows.

2.2. Rough set and neighborhood rough set

Rough set is an important mathematical tool for analyzing data. Rough set can analyze data depending on the knowledge of data itself; it does not need priori knowledge and is applied in attribute reduction and knowledge discovery (Swiniarski and Skowron, 2003; Mi et al., 2004; Yao and Yao, 2012; Zhang et al., 2016; Song et al., 2017). For rough set, there is a consistency hypothesis that the samples with the same attribute values should be partitioned into the same decision class. For rough set, there are following definitions which can be seen from Swiniarski and Skowron (2003), Mi et al. (2004), Yao and Yao (2012), Zhang et al. (2016) and Song et al. (2017).

Definition 3. An information system can be expressed as a four tuples $IS = \langle U, A, V, f \rangle$, where U is a universe, A is the set of attributes, V is the values set of A which is called as the domain of values, f is an information function, $f\colon U \times A \to V$ and it means $\forall \boldsymbol{x} \in U, \forall a \in A, f(\boldsymbol{x}, a) \in V_a$. If $A = C \cup D$ and $C \cap D = \emptyset$ where C is a condition attribute set and D is a decision attribute set, the information system is called as decision table.

Definition 4. Let $IS = \langle U, A, V, f \rangle$ and $B \subseteq A$, the indistinguishable relation *IND(B)* is defined as

$$IND(B) = \{(x, y) \in U \times U \mid f(x, a) = f(y, a), \forall a \in B\} \tag{1}$$

It is obvious that indistinguishable relation is an equivalence relation. If $(x, y) \in IND(B)$, it represents that the attributes values of x and y are the same on the attribute set B.

Definition 5. Let $IS = \langle U, A, V, f \rangle$ and $B \subseteq A$; for $\forall \in U$, the equivalent class of x on B is as

$$[x]_B = \{y \in U \mid (x, y) \in IND(B)\} \tag{2}$$

The equivalence partition of all equivalent classes according to *IND(B)* is denoted as *U/IND(B)* or *U/B*. For an equivalence partition $U/IND(B) = \left\{X_1, X_2, \dots, X_l\right\}$, it is known that if $\forall i \neq j$, there are $X_i \cap X_j = \emptyset$ and $\bigcup_{i=1}^{l} X_i = U$.

Definition 6. Let $IS = \langle U, A, V, f \rangle$, $B \subseteq A$ and $X \subseteq U$, the lower approximation set $\underline{R_B}X$ and the upper approximation set $\overline{R_B}X$ are as

$$\underline{R_B}X = \left\{x \in U \mid [x]_B \subseteq X\right\} \quad \text{and} \quad \overline{R_B}X = \left\{x \in U \mid [x]_B \cap X \neq \emptyset\right\} \tag{3}$$

Rough set divides the universe into three parts. The positive region is as $POS_R(X) = \underline{R_B}X$, the negative region is as $NEG_R(X) = U - \overline{R_B}X$ and the boundary region is as $BN_R(X) = \overline{R_B}X - \underline{R_B}X$. The positive region is the set of the samples which certainly belong to the concept X, the boundary region is the set of the samples which potentially belong to the concept X and the samples in the negative region can confirm not belonging to the concept X.

Let $X \subseteq U$ and $X \neq \emptyset$, the approximation accuracy $\alpha_{R_B}(X)$ and approximation quality $\gamma_{R_B}(X)$ of X on B are as

$$\alpha_{R_B}(X) = \frac{\left|\underline{R_B}X\right|}{\left|\overline{R_B}X\right|} \quad \text{and} \quad \gamma_{R_B}(X) = \frac{\left|\underline{R_B}X\right|}{|X|} \tag{4}$$

where $|\cdot|$ is the cardinal number of a set, $0 \le \alpha_{R_B}(X) \le 1$ and $\gamma_{R_B}(X) \le 1$. The approximation accuracy and approximation quality are the uncertainty metrics of the rough set. The larger the values are, the smaller the uncertainty of a set will be.

From the above, it is known that rough set can only deal with categorical data set and it cannot be applied in numeric data set. In order to extend rough set to deal with numeric data set, neighborhood rough set is proposed (Hu et al., 2008; Zhang et al., 2014b; Chen et al., 2014).

Definition 7. Let $x \in U$, $\delta \geq 0$, the set $\delta(x)$ is named as the neighborhood of x

$$\delta(x) = \{y \mid \Delta(x, y) \le \delta, y \in U\} \tag{5}$$

where $\Delta(x, y)$ is the distance between x and y, and δ is the radius of the neighborhood.

Definition 8. Let $NI = \langle U, A, V, f, \delta \rangle$ be a neighborhood information system and $B \subseteq A$, the neighborhood relation $NR_\delta(B)$ and neighborhood class $n_B^\delta(x)$ of x on B are defined as

$$\begin{aligned} &NR_\delta(B) = \{(x, y) \in U \times U \mid \Delta(x, y) \le \delta\} \quad \text{and} \\ &n_B^\delta(x) = \{y \mid \Delta(x, y) \le \delta, y \in U\} \end{aligned} \tag{6}$$

Definition 9. Let $NI = \langle U, A, V, f, \delta \rangle$ be a neighborhood information system, $B \subseteq A$ and $X \subseteq U$, the lower approximation set $\underline{NR}X$ and upper approximation set $\overline{NR}X$ are as

$$\underline{NR}X = \left\{x \mid n_B^\delta(x) \subseteq X, x \in U\right\} \quad \text{and} \quad \overline{NR}X = \left\{x \mid n_B^\delta(x) \cap X \neq \emptyset, x \in U\right\} \tag{7}$$

Generally speaking, the positive region is as $POS(X) = \underline{NR}X$, the negative region is as $NEG(X) = U - \overline{NR}X$ and the boundary region is as $BN(X) = \overline{NR}X - \underline{NR}X$. The uncertainty measure of neighborhood rough set can be also defined as Eq. (4). Neighborhood rough set can handle data set with numeric attributes. After the radius δ is determined, the uncertainty of the knowledge can be computed which can be utilized to measure the similarity of data points.

3. The nonlinear dimensionality reduction method for mixed data

In this section, we will introduce a mapping method which can make categorical attribute value become numeric attribute, then a nonlinear dimensionality reduction method based on neighborhood similarity is presented to decrease the data dimensions.

3.1. The attribute mapping method for mixed data

For a mixed data set, categorical attribute has no geometric structure which is inconvenient for data preprocessing. Let attributes set be $A = A^c \cup A^n$ and data set be $\boldsymbol{X} = \boldsymbol{X}^c \cup \boldsymbol{X}^n$ where A^c is the categorical attributes set, A^c is the numeric attributes set, $\boldsymbol{X}^c$ is the data set of categorical attributes, $\boldsymbol{X}^n$ is the data set of numeric attributes, $\boldsymbol{X}^c \in \mathbb{R}^{n \times m_c}$ and $\boldsymbol{X}^n \in \mathbb{R}^{n \times m_n}$. Therefore the partition of the universe by A^c is as $U/A^c = \left\{\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_l\right\}$ and $\bigcup_{i=1}^{l} \boldsymbol{X}_i = \boldsymbol{X}$. For each $\boldsymbol{X}_i$, the samples belong to the same equivalence class if only considering categorical attributes. The points in the same equivalence class should have a large similarity and the points in the different equivalence class should have a small similarity.

Definition 10. Let $IS = \langle U, A^c, V, f \rangle$ be an information system, for $\forall a \in A^c, \boldsymbol{x}_i \in U$ and $\boldsymbol{x}_i^n = \left\{x_{i1}, x_{i2}, \dots, x_{i|A^n|}\right\}$ which is the numeric attribute values of $\boldsymbol{x}_i$, the significance of $\boldsymbol{x}_i$ $(i = 1, 2, \dots, |U|)$ is as

$$sig(\boldsymbol{x}_i, a) = \min_{\boldsymbol{y} \notin [\boldsymbol{x}_i]_a}(\Delta(\boldsymbol{x}_i^n, \boldsymbol{y}^n)) - \max_{\boldsymbol{y} \in [\boldsymbol{x}_i]_a}(\Delta(\boldsymbol{x}_i^n, \boldsymbol{y}^n)) \tag{8}$$

where $\Delta(\boldsymbol{x}_i^n, \boldsymbol{y}^n) = \sqrt{\sum_{j=1}^{|A^n|}(x_{ij} - y_j^n)^2}$, $\boldsymbol{y}^n$ is the set of numeric attribute values of $\boldsymbol{y}$, $\boldsymbol{y}^n = \left\{y_{i1}, y_{i2}, \dots, y_{i|A^n|}\right\}$ and $\left[\boldsymbol{x}_i\right]_a$ is the equivalence class of $\boldsymbol{x}_i$ on a.

For each $\boldsymbol{x}_i$, the first item of Eq. (8) is the dissimilarity degree between $\boldsymbol{x}_i$ and the data point in different equivalence class; the second item of Eq. (8) is the dissimilarity degree of the points in the same equivalence class. If the values of the first item in Eq. (8) is larger, the data points in different clusters are separated well; if the values of the second item in Eq. (8) is smaller, the data points in same cluster are more compact. $sig(\boldsymbol{x}_i, a)$ reflects the quality of the attribute a to partition universe, in other words, it is the importance of a for $\boldsymbol{x}_i$. For $\forall \boldsymbol{x}_i \in U$ $(i = 1, 2, \dots, |U|)$, $sig(\boldsymbol{x}_i, a)$ is expected as large as possible which means the attribute $a \in A$ provides more information for $\boldsymbol{x}_i$. Therefore $sig(\boldsymbol{x}_i, a)$ can replace $f(\boldsymbol{x}_i, a)$ to make the categorical attribute of universe become numeric attribute. The attribute mapping method for mixed data is as Algorithm 1.

Algorithm 1 The attribute mapping method for mixed data

```
Input: An universe $U$; the attribute set $A$ of $U$.
Output: The numeric attribute set $A$.
Divide the attribute set $A$ into $A^c$ and $A^n$;
Form equivalence partition $U/A^c$;
for each $a \in A^c$ do
    for each $\boldsymbol{x}_i \in U$ do
        Calculate $sig(\boldsymbol{x}_i, a)$;
        $f(\boldsymbol{x}_i, a) \leftarrow sig(\boldsymbol{x}_i, a)$;
    end for
end for
```

From Algorithm 1, it is known that the attribute mapping method has many advantages. Categorical attributes become numeric attribute which facilitates the preprocessing of data set; the transformed attribute value synthetically considers the impact of categorical attributes and numeric attributes and it reflects the importance of the categorical attributes for data point. After executing Algorithm 1, the attribute set of universe has a geometric structure which means geometric distance can be employed to measure dissimilarity.

3.2. The nonlinear dimensionality reduction method based on neighborhood similarity

Laplacian Eigenmaps (LE) is an important nonlinear dimensionality reduction method (Belkin and Niyogi, 2001). It reconstructs data points according to the neighbors. In this section, a nonlinear dimensionality reduction method based on neighborhood similarity is developed.

For $\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in U$ and $B \subseteq A$, the neighborhood classes of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ on B are as

$$n_B^\delta(\boldsymbol{x}_i) = \left\{\boldsymbol{y} | \Delta(\boldsymbol{x}_i, \boldsymbol{y}) \le \delta\right\} \quad \text{and} \quad n_B^\delta(\boldsymbol{x}_j) = \left\{\boldsymbol{y} | \Delta(\boldsymbol{x}_j, \boldsymbol{y}) \le \delta\right\} \tag{9}$$

If $\boldsymbol{x}_j \in n_B^\delta(\boldsymbol{x}_i)$, the similarity of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is as

$$sim(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mu_{\boldsymbol{x}_j}^B(\boldsymbol{x}_i) = \frac{\left|n_B^\delta(\boldsymbol{x}_i) \cap n_B^\delta(\boldsymbol{x}_j)\right|}{\left|n_B^\delta(\boldsymbol{x}_i)\right|} \tag{10}$$

where $\mu_{\boldsymbol{x}_j}^B(\boldsymbol{x}_i)$ is the membership function. The dissimilarity of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is as

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - sim(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{11}$$

If $\boldsymbol{x}_j \notin n_B^\delta(\boldsymbol{x}_i)$, the similarity of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is as

$$sim(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - Dijkstra(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{12}$$

where $Dijkstra(\boldsymbol{x}_i, \boldsymbol{x}_j)$ represents that it employs Dijkstra algorithm (Dijkstra, 1959) to solve the minimum dissimilarity of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. It is obvious that there is $sim(\boldsymbol{x}_i, \boldsymbol{x}_j) \ne sim(\boldsymbol{x}_j, \boldsymbol{x}_i)$, therefore $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is not equal to $d(\boldsymbol{x}_j, \boldsymbol{x}_i)$.

For $\boldsymbol{X} \in \mathbb{R}^{n\times m}$, $\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X}$, $\boldsymbol{x}_i \in \mathbb{R}^{1\times m}$ and $\boldsymbol{x}_j \in \mathbb{R}^{1\times m}$ where m is the dimension of data point. A weight matrix $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ can be defined as

$$\boldsymbol{W} = \begin{bmatrix} sim(\boldsymbol{x}_1, \boldsymbol{x}_1) + sim(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & sim(\boldsymbol{x}_1, \boldsymbol{x}_n) + sim(\boldsymbol{x}_n, \boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ sim(\boldsymbol{x}_n, \boldsymbol{x}_1) + sim(\boldsymbol{x}_1, \boldsymbol{x}_n) & \cdots & sim(\boldsymbol{x}_n, \boldsymbol{x}_n) + sim(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix} \tag{13}$$

Let $\boldsymbol{Y} = \left\{\boldsymbol{y}_1^T, \boldsymbol{y}_2^T, \dots, \boldsymbol{y}_n^T\right\}^T \in \mathbb{R}^{n\times d}$ be the data points of $\boldsymbol{X}$ after dimensionality reduction where d is the dimension of $\boldsymbol{Y}$. The similar samples are still close to each other after dimensionality reduction in the target subspace, therefore the objective function is as follows:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|_2^2 \tag{14}$$

If considering $\sum_{i=1}^n \sum_{j=1}^n W_{ij} \left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|_2^2$ independently, it can deliver the following result:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|_2^2 = 2\boldsymbol{tr}(\boldsymbol{Y}(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{Y}^T) = 2\boldsymbol{tr}(\boldsymbol{Y}\boldsymbol{L}\boldsymbol{Y}^T) \tag{15}$$

where D is a diagonal matrix, $D_{ii} = \sum_{j=1}^n W_{ij}$ and $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$. The above optimization problems can be expressed as follows:

$$\begin{aligned} &\min\ \boldsymbol{tr}(\boldsymbol{Y}\boldsymbol{L}\boldsymbol{Y}^T) \\ &s.t.\ \boldsymbol{Y}\boldsymbol{D}\boldsymbol{Y}^T = \boldsymbol{I} \end{aligned} \tag{16}$$

Let $L = \boldsymbol{tr}(\boldsymbol{Y}\boldsymbol{L}\boldsymbol{Y}^T) - \lambda(\boldsymbol{Y}\boldsymbol{D}\boldsymbol{Y}^T - \boldsymbol{I})$, the optimization problem can be solved according to the KKT condition (Zhang, 2013). The result is as

$$\frac{\partial L}{\partial \boldsymbol{Y}} = 2\boldsymbol{L}\boldsymbol{Y} - 2\lambda \boldsymbol{D}\boldsymbol{Y} = 0 \Rightarrow \boldsymbol{L}\boldsymbol{Y} = \lambda \boldsymbol{D}\boldsymbol{Y} \tag{17}$$

From Eq. (17), it is known that each column vector of $\boldsymbol{Y}$ is the eigenvector corresponding to generalized eigenvalue. In general, the eigenvectors corresponding to d minimum nonzero eigenvalues are chosen as the result of dimension reduction, $\boldsymbol{Y} = \left[\boldsymbol{Y}_1, \boldsymbol{Y}_2, \dots, \boldsymbol{Y}_d\right]$ and $\boldsymbol{Y}_i \in \mathbb{R}^{n\times 1}$ $(i = 1, 2, \dots, d)$.

After executing Algorithm 1, the data set can be further processed by the nonlinear dimension reduction method. Therefore the dimensionality reduction method based on neighborhood similarity are as Algorithm 2.

Algorithm 2 The nonlinear dimension reduction method based on neighborhood similarity

```
Input: A data set $\boldsymbol{X}$, $d$ and $\delta$.
Output: The data set $\boldsymbol{Y}$.
 1: for each $\boldsymbol{x}_i \in \boldsymbol{X}$ do
 2:     Obtain the neighborhood class $n_B^\delta(\boldsymbol{x}_i)$ of $\boldsymbol{x}_i$;
 3: end for
 4: Initialize the weight matrix $\boldsymbol{W}$;
 5: for each $\boldsymbol{x}_i \in \boldsymbol{X}$ do
 6:     for each $\boldsymbol{x}_j \in n_B^\delta(\boldsymbol{x}_i)$ do
 7:         $W_{ij} \leftarrow W_{ij} + sim(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
 8:         $W_{ji} \leftarrow W_{ji} + sim(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
 9:     end for
10: end for
11: for each $\boldsymbol{x}_i \in \boldsymbol{X}$ do
12:     for each $\boldsymbol{x}_j \notin n_B^\delta(\boldsymbol{x}_i)$ do
13:         $W_{ij} \leftarrow W_{ij} + 1 - Dijkstra(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
14:         $W_{ji} \leftarrow W_{ji} + 1 - Dijkstra(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
15:         if $W_{ij} < 0$ then
16:             $W_{ij} = W_{ji} \leftarrow 0$;
17:         end if
18:     end for
19: end for
20: Calculate of the diagonal matrix $\boldsymbol{D}$ and the matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$;
21: Calculate the $d$ eigenvectors $\boldsymbol{Y}_i \in \mathbb{R}^{n\times 1}$ corresponding to the
    minimum $d$ nonzero generalized eigenvalues $(i = 1, 2, \dots, d)$;
22: Obtain the matrix $\boldsymbol{Y} = \left[\boldsymbol{Y}_1, \boldsymbol{Y}_2, \dots, \boldsymbol{Y}_d\right]$
```

From Algorithm 2, it is known that $sim(\boldsymbol{x}_i, \boldsymbol{x}_j) \ne sim(\boldsymbol{x}_j, \boldsymbol{x}_i)$. The steps 5–10 and 11–19 ensure that the matrix $\boldsymbol{W}$ is a symmetric matrix which makes Eq. (17) be true. In addition, the steps 15–17 ensure that $\boldsymbol{W}_{ij}$ is representation of $sim(\boldsymbol{x}_i, \boldsymbol{x}_j)$ because the similarity cannot be less than zero. After dimension reduction, the complex of data set is reduced. Beyond that, the similarity measure of this method does not depend on distance and it is determined by the uncertainty of neighborhood which avoids the adverse effects of Euclidean distance on similarity measurement in manifold space.

4. The clustering processes of *SNDC* for mixed data stream

In this section, we will explain the detail principles of *SNDC*. First, a new distance is defined based on neighborhood entropy which is used to measure the distance of two objects. Then a self-adaption neighborhood density clustering method for fixed data stream is proposed and concept drift detection method is also presented.

4.1. The distance of neighborhood entropy

Entropy is an important measure of uncertainty. The entropy value illuminates the uncertainty of an information system (Liang et al., 2009; Liang and Qian, 2008). In an information system, the Shannon entropy is as

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i \tag{18}$$

where $X = \left[x_1^T, x_2^T, \dots, x_n^T\right]^T$ and p_i is the probability of x_i $(i = 1, 2, \dots, n)$. If the entropy value is greater, the degree of disorder will be greater. For a neighborhood information system $NI = \langle U, A, V, f, \delta\rangle$, neighborhood entropy is defined as Definition 11.

Definition 11. Let $NI = \langle U, A, V, f, \delta\rangle$ and $B \subseteq A$, for $\forall \boldsymbol{x}_i \in U$, the neighborhood entropy of $\boldsymbol{x}_i$ on B is as

$$H_B^\delta(\boldsymbol{x}_i) = -\sum_{\boldsymbol{x}_j \in n_B^\delta(\boldsymbol{x}_i)} \frac{\left|n_B^\delta(\boldsymbol{x}_j)\right|}{|U|} \log(\frac{\left|n_B^\delta(\boldsymbol{x}_j)\right|}{|U|} \cdot \frac{1}{\left|U - n_B^\delta(\boldsymbol{x}_j)\right|}) \tag{19}$$

where $n_B^\delta(\boldsymbol{x}_i)$ is the neighborhood class of $\boldsymbol{x}_i$ on B and let $\log x = 0$ $(x \to \infty)$.

Proposition 1. *For $\forall \boldsymbol{x}_i \in U$, $H_\delta(\boldsymbol{x}_i)$ possesses the following properties:*

1. *$H_B^\delta(\boldsymbol{x}_i)$ can obtain a minimal value 0 if and only if $n_B^\delta(\boldsymbol{x}_j) = U$ $(j = 1, 2, \dots, \left|n_B^\delta(\boldsymbol{x}_i)\right|)$;*
2. *$H_B^\delta(\boldsymbol{x}_i)$ can obtain a maximum value $-\frac{1}{|U|} \log \frac{1}{|U|(|U|-1)}$ if and only if $n_B^\delta(\boldsymbol{x}_i) = n_B^\delta(\boldsymbol{x}_j) = \{\boldsymbol{x}_i\} = \{\boldsymbol{x}_j\}$.*

Theorem 1. *Let $NI = \langle U, A, V, f, \delta\rangle$ and $C \subseteq B \subseteq A$, for $\forall \boldsymbol{x}_i \in U$, there is $H_C^\delta(\boldsymbol{x}_i) \leq H_B^\delta(\boldsymbol{x}_i)$.*

Proof. The proof can be seen in Appendix A. □

Theorem 2. *Let $NI = \langle U, A, V, f, \delta\rangle$, $B \subseteq A$ and $\delta_1 \leq \delta_2$, for $\forall \boldsymbol{x}_i \in U$, there is $H_B^{\delta_1}(\boldsymbol{x}_i) \geq H_B^{\delta_2}(\boldsymbol{x}_i)$.*

Proof. The proof can be seen in Appendix B. □

Theorem 3. *Let $NI = \langle U, A, V, f, \delta\rangle$, $B \subseteq A$ and $\delta = \delta_1 + \delta_2$ $(\delta_1 \geq \delta_2)$, for $\forall \boldsymbol{x}_i \in U$, if $\boldsymbol{x}_j \in n_B^\delta(\boldsymbol{x}_i) - n_B^{\delta_1}(\boldsymbol{x}_i)$, $\boldsymbol{x}_{j'} \in n_B^{\delta_2}(\boldsymbol{x}_i)$ and $n_B^{\delta_2}(\boldsymbol{x}_{j'}) \subseteq n_B^{\delta_2}(\boldsymbol{x}_j)$, there is $H_B^\delta(\boldsymbol{x}_i) \leq H_B^{\delta_1}(\boldsymbol{x}_i) + H_B^{\delta_2}(\boldsymbol{x}_i)$.*

Proof. The proof can be seen in Appendix C. □

Definition 12. Let $NI = \langle U, A, V, f, \delta\rangle$, $B \subseteq A$ be a neighborhood information system and $\boldsymbol{x}, \boldsymbol{y} \in U$, the neighborhood distance $d_B^\delta(\boldsymbol{x}, \boldsymbol{y})$ of $\boldsymbol{x}$ and $\boldsymbol{y}$ on B is as

$$d_B^\delta(\boldsymbol{x}, \boldsymbol{y}) = \left|\frac{1}{\left|n_B^\delta(\boldsymbol{x})\right|} \cdot H_B^\delta(\boldsymbol{x}) - \frac{1}{\left|n_B^\delta(\boldsymbol{y})\right|} \cdot H_B^\delta(\boldsymbol{y})\right| \tag{20}$$

Proposition 2. *For $\forall \boldsymbol{x}, \boldsymbol{y} \in U$, $d_B^\delta(\boldsymbol{x}, \boldsymbol{y})$ satisfies the following properties:*

1. *Nonnegativity: $d_B^\delta(\boldsymbol{x}, \boldsymbol{y}) \geq 0$; if $\boldsymbol{x} = \boldsymbol{y}$, $d_B^\delta(\boldsymbol{x}, \boldsymbol{y}) = 0$;*
2. *Symmetry: $d_B^\delta(\boldsymbol{x}, \boldsymbol{y}) = d_B^\delta(\boldsymbol{y}, \boldsymbol{x})$;*
3. *Triangle inequality: $d_B^\delta(\boldsymbol{x}, \boldsymbol{y}) \leq d_B^\delta(\boldsymbol{x}, \boldsymbol{z}) + d_B^\delta(\boldsymbol{z}, \boldsymbol{y})$ for $\forall z \in U$.*

Proof. The proof can be seen in Appendix D. □

It is obvious that $d_B^\delta(\boldsymbol{x}, \boldsymbol{y})$ is the distance of $\boldsymbol{x}$ and $\boldsymbol{y}$. It can utilize $d_B^\delta(\boldsymbol{x}, \boldsymbol{y})$ to measure the dissimilarity of $\boldsymbol{x}$ and $\boldsymbol{y}$. Different from Euclidean distance, $d_B^\delta(\boldsymbol{x}, \boldsymbol{y})$ is based on the uncertainty measure which does not depend on the geometric space.

4.2. Self-adaption density clustering algorithm

For data stream, concept drift is generated with the change of time-varying data streams. Therefore the latest data has a greater importance. Sample weighting mechanism is widely used in data stream. Let a data stream $\boldsymbol{S} = \{\dots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \dots\}$, where $\boldsymbol{x}_t$ is a data point generated at t moment and $\boldsymbol{x}_t \in \mathbb{R}^{1\times m}$, the weight of $\boldsymbol{x}_t$ is $f_w(\boldsymbol{x}_t) = 2^{-\lambda(T-t)}$ where T is the current time and λ is a predefined parameter. The weight value of $\boldsymbol{x}_t$ exponentially decays with time going on. In order to facilitate the description of the algorithm, in this paper, we deem time-step is equal to the sequence number of data which means $\boldsymbol{S} = \{\dots, \boldsymbol{x}_{i-1}, \boldsymbol{x}_i, \boldsymbol{x}_{i+1}, \dots\}$ where $\boldsymbol{x}_i$ is the ith data point and the size of a data block is equal to the size of a sliding window. Let β_0 be the lower limit of the weight of data point; if the weight of a data point is less than β_0, the data point is outdated and deleted from sliding window. For a data point, if it is deleted, the least time Δt is as

$$2^{-\lambda \Delta t} \leq \beta_0 \Rightarrow \Delta t \geq -\frac{\log \beta_0}{\lambda} \tag{21}$$

Theorem 4. *Let M be the size of a sliding window; assume that the probability of concept drift in a sliding window obeys a normal distribution:*

$$f(x) = \frac{\lambda}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \ (1 \leq x \leq M) \tag{22}$$

therefore the size of a sliding window M should be $M \leq \sqrt{-2\ln(\frac{\lambda^2 e^{-\frac{1}{2}} - 2}{\lambda^2})}$ and $\sqrt{\frac{2}{e^{-\frac{1}{2}}}} \leq \lambda \leq \sqrt{\frac{2}{e^{-\frac{1}{2}} - 1}}$.

Proof. The proof can be seen in Appendix E. □

For a data block $\boldsymbol{B}_i$ $(i = 1, 2, 3, \dots)$, how to determine the clustering center points is a crucial problem. A clustering center point should have a large density and a large distance from other center points. The possibility of a data point being a center point can be evaluated as

$$p(\boldsymbol{x}_j, \boldsymbol{c}_i) = f_w(\boldsymbol{x}_j) \frac{\left|n_{\boldsymbol{B}_i}^\delta(\boldsymbol{x}_j)\right|}{\left|\boldsymbol{B}_i\right|} + \beta \sum_{l=1}^{|c_i|} d(\boldsymbol{x}_j, \boldsymbol{c}_{il}), \boldsymbol{x}_j \in \boldsymbol{B}_i \text{ and } \boldsymbol{c}_l \in \boldsymbol{c}_i \tag{23}$$

where k is the number of clusters and δ is a predefined parameter. The first item of Eq. (23) is related to the density of $\boldsymbol{x}_j$. The greater the density of the neighborhood is, the larger the value of the first item will be. k can be determined by the rough set partition $\boldsymbol{B}_i/A^c$. k is set as $\left|\boldsymbol{B}_i/A^c\right|$ and A^c is the categorical attribute set of $\boldsymbol{B}_i$ before dimension reduction. In order to determine the parameter δ, we can execute a prior *k-means* clustering. The average radius of all clusters is as the value of δ and δ is calculated as

$$\delta = \frac{1}{k} \sum_{l=1}^{k} \max_{\boldsymbol{x}_j \in \boldsymbol{c}'_{il}} (d(\boldsymbol{x}_j, \boldsymbol{c}'_{il})) \tag{24}$$

$\boldsymbol{c}'_{il}$ is the clustering center points selected by *k-means* algorithm and $d(\boldsymbol{x}_j, \boldsymbol{c}'_{il})$ is the distance of $\boldsymbol{x}_j$ and the center point of the cluster $\boldsymbol{c}'_{il}$. After δ is computed, the k data points with the maximum values of Eq. (23) are selected as the initial center points. The method selecting the initial center points can be summarized as Algorithm 3.

Algorithm 3 The method of selecting the initial center points

```
Input: A data block set $\boldsymbol{B}_i$, $k$, $\delta$ and the categorical attributes $A^c$.
Output: The $k$ center points $\boldsymbol{C}$.
for each $\boldsymbol{x}_j \in \boldsymbol{B}_i$ do
    Obtain the neighborhood class $n^{\delta}_{A^c}(\boldsymbol{x}_j)$ of $\boldsymbol{x}_j$;
end for
Select a data point $\boldsymbol{x}$ with maximum density;
$\boldsymbol{C} \leftarrow \boldsymbol{C} \cup \boldsymbol{x}$, $\boldsymbol{B}_i \leftarrow \boldsymbol{B}_i - \boldsymbol{x}$ and $k' = 1$;
while $k' \leq k$ do
    for each $\boldsymbol{x}_j \in \boldsymbol{B}_i$ do
        Calculate $p(\boldsymbol{x}_j, \boldsymbol{C})$;
    end for
    $\boldsymbol{x} = \operatorname{argmax} p(\boldsymbol{x}_j, \boldsymbol{C})$;
    $\boldsymbol{C} \leftarrow \boldsymbol{C} \cup \boldsymbol{x}$, $\boldsymbol{B}_i \leftarrow \boldsymbol{B}_i - \boldsymbol{x}$ and $k' \leftarrow k' + 1$;
end while
```

After k center points are determined, each data point can be partitioned into a cluster by the neighborhood distance. Each data point belongs to the nearest cluster. The cluster of $\boldsymbol{x}_j$ is determined as

$$q = \underset{1 \leq l \leq k}{\operatorname{argmin}}\ d^{\delta}_{B}(\boldsymbol{x}_j, \boldsymbol{c}_{il}),\ \ \boldsymbol{x}_j \in \boldsymbol{B}_i \tag{25}$$

Therefore a clustering vector $\boldsymbol{v}_i \in \mathbb{R}^{1\times k}$ of $\boldsymbol{B}_i$ is generated from the clustering result and $\boldsymbol{v}_i = \left\{\left|\boldsymbol{c}_{i1}\right|, \left|\boldsymbol{c}_{i2}\right|, \ldots, \left|\boldsymbol{c}_{ik}\right|\right\}$.

After $\boldsymbol{B}_i$ is handled, $\boldsymbol{B}_i$ is deleted and the next data block $\boldsymbol{B}_{i+1}$ enters into the sliding window. If there is no concept drift in $\boldsymbol{B}_{i+1}$, the distributions of the two data block are the same or the difference of the distributions is small. Hence $\boldsymbol{v}_i$ can be as the clustering center points of $\boldsymbol{B}_{i+1}$. The cluster of each data point in $\boldsymbol{B}_{i+1}$ is determined as Eq. (25). The mean square error of the clustering result is as

$$E_{i+1} = \frac{1}{M}\sum_{j=1}^{M}\sum_{l=1}^{k} f_w(\boldsymbol{x}_j) w_{jl} \left\|\boldsymbol{x}_j - \boldsymbol{c}_{(i+1)l}\right\|_2^2,\quad \boldsymbol{x}_j \in \boldsymbol{B}_{i+1} \tag{26}$$

where w_{jl} is the ownership weight of $\boldsymbol{x}_j$; if $\boldsymbol{x}_j \in \boldsymbol{c}_{(i+1)l}$, $w_{jl} = 1$; else $w_{jl} = 0$. $f_w(\boldsymbol{x}_j)$ is the weight of $\boldsymbol{x}_j$ in $\boldsymbol{B}_{i+1}$.

In the data block $\boldsymbol{B}_{i+1}$, because of concept drift including gradual concept drift and abrupt concept drift, the distribution has a large dissimilarity between $\boldsymbol{B}_{i+1}$ and $\boldsymbol{B}_{i+1}$. We can adjust the clustering center points according to the clustering result. Let the final clustering center points of $\boldsymbol{B}_i$ be $\boldsymbol{c}_i$ and $\boldsymbol{c}_i \in \mathbb{R}^{k\times d}$, the initial centering center points $\boldsymbol{c}_{i+1}$ of $\boldsymbol{B}_{i+1}$ is set as $\boldsymbol{c}_i$. The partial derivative of E_{i+1} is as

$$\frac{\partial E_{i+1}}{\partial \boldsymbol{c}_{(i+1)l}} = -\frac{1}{M}\sum_{j=1}^{M} f_w(\boldsymbol{x}_j) w_{jl}(\boldsymbol{x}_j - \boldsymbol{c}_{(i+1)l}),\ (l = 1, 2, \ldots, k) \tag{27}$$

The $\boldsymbol{c}_{(i+1)l}$ can be updated from the gradient descent method (Kiwiel, 2001; Comaniciu and Meer, 2002) and it can be computed as

$$\boldsymbol{c}_{(i+1)l} \leftarrow \boldsymbol{c}_{(i+1)l} + \eta\frac{1}{M}\sum_{j=1}^{M} f_w(\boldsymbol{x}_j) w_{jl}(\boldsymbol{x}_j - \boldsymbol{c}_{(i+1)l}) \tag{28}$$

where η is a learning step which is predefined by prior knowledge. The clustering center points of $\boldsymbol{B}_{i+1}$ can be updated iteratively until the maximum number of iteration is reached or the change of the mean square error is small enough. The method of updating the clustering center points is as Algorithm 4.

Algorithm 4 The method of updating the clustering center points

```
Input: A data block set $\boldsymbol{B}_{i+1}$, the clustering center points $\boldsymbol{B}_i$: $\boldsymbol{c}_i$, the
    maximum number $N_{max}$, the threshold $\varepsilon$ and the learning step $\eta$;
Output: The final center points $\boldsymbol{c}_{i+1}$.
$j$=0, $\Delta E = +inf$;
$\boldsymbol{c}_0 \leftarrow \boldsymbol{c}_i$; $\boldsymbol{c}_{i+1} \leftarrow \boldsymbol{c}_i$;
while $j \leq N_{max} \&\& \Delta E > \varepsilon$ do
    $j \leftarrow j + 1$;
    Determine the cluster of each data point of $\boldsymbol{B}_{i+1}$ according to
  the clustering center points $\boldsymbol{c}_0$;
    for each $\boldsymbol{c}_{(i+1)l} \in \boldsymbol{c}_{i+1}$ do
        $\boldsymbol{c}_{(i+1)l} \leftarrow \boldsymbol{c}_{(i+1)l} + \eta\frac{1}{M}\sum_{j=1}^{M} f_w(\boldsymbol{x}_j) w_{jl}(\boldsymbol{x}_j - \boldsymbol{c}_{(i+1)l})$;
    end for
    $\Delta E = \left\|\boldsymbol{c}_0 - \boldsymbol{c}_{i+1}\right\|_2$;
    $\boldsymbol{c}_0 \leftarrow \boldsymbol{c}_{i+1}$;
end while
```

After the clustering center points is updated, a cluster vector is also generated for $\boldsymbol{B}_{i+1}$ and the cluster vector is as $\boldsymbol{v}_{i+1} = \left\{\left|\boldsymbol{c}_{(i+1)1}\right|, \left|\boldsymbol{c}_{(i+1)2}\right|, \ldots, \left|\boldsymbol{c}_{(i+1)k}\right|\right\}$ where $\left|\boldsymbol{c}_{(i+1)k}\right|$ is the number of data points in the cluster $\boldsymbol{c}_{(i+1)k}$. Therefore concept drift can be detected by comparing the similarity between $\boldsymbol{v}_i$ and $\boldsymbol{v}_{i+1}$. The similarity between $\boldsymbol{v}_i$ and $\boldsymbol{v}_{i+1}$ is as

$$\theta = \arccos\frac{\boldsymbol{v}_i \cdot \boldsymbol{v}_{i+1}}{\left|\boldsymbol{v}_i\right| \cdot \left|\boldsymbol{v}_{i+1}\right|} \tag{29}$$

If $\theta \leq \tau$ (τ is a threshold), it means that $\boldsymbol{v}_i$ is similar to $\boldsymbol{v}_{i+1}$ and there is no concept drift in data stream or the change is very small where $0 \leq \tau \leq 1$. If $\tau \geq \theta$, it shows that there is a great difference between $\boldsymbol{v}_i$ and $\boldsymbol{v}_{i+1}$ and concept drift has appeared in data stream; hence the clustering center points are not fit for the distribution of the current data; then Algorithm 3 is executed to search new clustering center points. The detail steps of *SDNC* are as Algorithm 5.

Algorithm 5 SNDC

```
Input: A data stream $\boldsymbol{S}$, $M$, $d$, $\beta$, $\eta$ and $\tau$.
Output: The clusters of each data block.
while $\boldsymbol{S} \neq NULL$ do
    Obtain $M$ data points to organize a data block $\boldsymbol{B}_i$;
    Obtain the weight of each data point in the data block $\boldsymbol{B}_i$;
    Execute Algorithm 1 to make the categorical attributes become
  the numeric attributes in $\boldsymbol{B}_i$;
    Execute Algorithm 2 to reduce the dimension of $\boldsymbol{B}_i$;
    if $i == 1$ then
        Execute Algorithm 3 to select the initial center points $\boldsymbol{c}_i$ and
  generate the cluster vector $\boldsymbol{v}_i$;
    else
        Utilize $\boldsymbol{c}_{i-1}$ to cluster $\boldsymbol{B}_i$ as Algorithm 4 and generate the
  cluster vector $\boldsymbol{v}_i$;
        Compute the similarity of $\boldsymbol{v}_{i-1}$ and $\boldsymbol{v}_i$ as Eq.(??);
        if $\theta \leq \tau$ then
            Obtain the cluster result of each data point in $\boldsymbol{B}_i$;
        else
            Execute Algorithm 3 to select the initial center points $\boldsymbol{c}_i$
  and generate the cluster vector $\boldsymbol{v}_i$;
            Obtain the cluster result of each data point in $\boldsymbol{B}_i$;
        end if
    end if
end while
```

From Algorithm 5, it is known that the algorithm can automatically adjust the center points if concept drift is appearing in data stream. In addition, the important parameter k and δ can be determined by rough set partition which does not need prior knowledge. The range of τ, λ and M can also be determined by inequality constraints which reduces the range of the parameters. The only parameter which is absolutely needing prior knowledge to be determined is η. It is obvious that η can affect the convergence of Algorithm 4 and the result of Algorithm 4 is a local optimal solution. τ is a parameter which can affect the sensitivity of *SNDC* to concept drift. A large value of τ makes *SNDC* sensitive to the change of the concept in data stream, but it is also sensitive to noise; a small value of τ makes *SNDC* relatively insensitive to the change of the concept in data stream, but concept drift may be also be ignored.

Table 1
The details of the experimental data sets.

Data set	Attributes	Numerical	Categorical	Samples	Classes	Type	Concept drift
Hyperplane	41	40	1	5,000	2	Mixed	Gradual
waveform	22	21	1	5,000	3	Mixed	Gradual
adult	14	10	4	5,000	2	Mixed	Gradual
student	34	1	33	5,000	3	Mixed	Abrupt
magic	11	1	10	19,020	2	Mixed	Abrupt
eye	15	14	1	5,000	2	Mixed	Abrupt
winequality	12	11	1	4,898	8	Mixed	Gradual
occupancy	6	5	1	8,143	2	Mixed	Abrupt
biodeg	42	25	17	1,055	2	Mixed	Abrupt
Thoracic	17	14	3	470	2	Mixed	Gradual
Germany	25	20	5	1,000	2	Mixed	Gradual
ionosphere	35	32	3	351	2	Mixed	Gradual

Table 2
The J and R of the algorithms testing on the experimental data sets.

Data set	J				R				M
	SNDC	k-means	k-service	DenStream	SNDC	k-means	k-service	DenStream	
Hyperplane	**0.4972 ± 0.0032**	0.3607 ± 0.0975	0.2998 ± 0.0404	0.1289 ± 0.0704	0.4996 ± 0.0028	**0.5966 ± 0.0822**	0.5221 ± 0.0207	0.5241 ± 0.0322	200
waveform	0.3316 ± 0.0032	0.3628 ± 0.0688	**0.3801 ± 0.0563**	0.2604 ± 0.0674	0.3401 ± 0.0034	**0.6838 ± 0.0417**	0.5851 ± 0.1049	0.6635 ± 0.0446	200
adult	**0.6457 ± 0.0457**	0.5218 ± 0.1421	0.4998 ± 0.0981	0.6251 ± 0.0611	0.6462 ± 0.0458	**0.6608 ± 0.1181**	0.5715 ± 0.0827	0.6534 ± 0.0441	200
student	**0.9712 ± 0.0679**	0.4015 ± 0.0832	0.4961 ± 0.1508	0.1891 ± 0.1012	**0.9712 ± 0.0679**	0.4062 ± 0.0847	0.5032 ± 0.1436	0.2019 ± 0.1023	200
magic	**0.9882 ± 0.0725**	0.3841 ± 0.0521	0.6253 ± 0.1272	0.7269 ± 0.3661	**0.9882 ± 0.0725**	0.3883 ± 0.0564	0.6268 ± 0.1251	0.7274 ± 0.3659	500
eye	**0.8867 ± 0.1679**	0.4446 ± 0.1593	0.6303 ± 0.1609	0.6561 ± 0.1831	**0.8867 ± 0.1679**	0.4815 ± 0.1946	0.6473 ± 0.1453	0.6562 ± 0.1831	200
winequality	**0.3255 ± 0.0265**	0.2391 ± 0.0301	0.2987 ± 0.0440	0.2631 ± 0.0578	0.3266 ± 0.0265	**0.5726 ± 0.0331**	0.4545 ± 0.0634	0.4850 ± 0.1373	500
occupancy	0.8292 ± 0.2215	0.5229 ± 0.1648	0.8254 ± 0.1964	**0.8294 ± 0.2214**	0.8294 ± 0.2213	0.5707 ± 0.2117	**0.8502 ± 0.1702**	0.8294 ± 0.2214	500
biodeg	**0.7994 ± 0.3048**	0.5173 ± 0.1485	0.5944 ± 0.1865	0.3061 ± 0.1505	**0.8078 ± 0.2897**	0.5267 ± 0.1441	0.6144 ± 0.1686	0.3478 ± 0.1492	50
Thoracic	**0.6744 ± 0.1482**	0.4845 ± 0.1663	0.5272 ± 0.1072	0.5744 ± 0.2129	**0.6931 ± 0.1116**	0.5581 ± 0.1456	0.5762 ± 0.0911	0.6190 ± 0.1547	50
Germany	**0.5368 ± 0.1001**	0.3707 ± 0.1019	0.4256 ± 0.0636	0.4142 ± 0.1292	**0.5637 ± 0.0616**	0.5346 ± 0.0879	0.5277 ± 0.0435	0.5339 ± 0.0671	40
ionosphere	**0.5691 ± 0.1923**	0.3781 ± 0.0669	0.4055 ± 0.0413	0.2647 ± 0.1164	**0.5724 ± 0.1904**	0.5196 ± 0.1035	0.5229 ± 0.0511	0.5189 ± 0.0563	45

5. The experimental results and analyses

In order to test the performance of *SNDC*, a series of data sets are chosen as the experimental data sets. *waveform* and *hyperplane* are generated by MOA[1] (Bifet et al., 2010) and the other data sets are from UCI Machine Learning Repository.[2] *DenStream* (Cao et al., 2006), *Streaming k-service* (Braverman et al., 2011) and *k-means* (Jiawei et al., 2012) with sliding window mechanism which is also denoted as *k-means* are chosen as comparison algorithms. All algorithms are executed on MATLAB 2017Ra. The parameters of *SNDC* are set as: $\tau = 0.5$, $\lambda = 2.3$ and $\eta = 3\tau$; M is changing with different data sets. The parameters of *Denstream* are set as: $\epsilon = 0.5, \beta = 0.2, \mu = 0.8$ and $\lambda = 0.2$. The parameters of *Streaming k-service* are as: $k = 3, \lambda = 2$ and $\beta = 1.5$. The details of the data sets are in Table 1.

5.1. The evaluation criteria

In this paper, four evaluation criteria are employed to evaluate the performance of the experimental algorithms. The evaluation criteria are Jaccard coefficient, Rand statistic, Fowlkes and Mallows index and Mean Square Error. The Jaccard coefficient is defined as

$$J = \frac{SS}{SS + SD + DS} \tag{30}$$

The Rand statistic is defined as

$$R = \frac{SS + DD}{SS + SD + DS + DD} \tag{31}$$

The Fowlkes and Mallows index is defined as

$$FM = \sqrt{\frac{SS}{SS + SD}} \cdot \sqrt{\frac{SS}{SS + DS}} \tag{32}$$

where *SS* is the number of the samples that they belong to the same class and they are in the same cluster; *SD* is the number of the samples that they belong to the same class but they are in the different cluster; *DS* is the number of the samples that they belong to different classes but they are in the same cluster; *DD* is the number of the samples that they belong to different classes and they are also in different clusters.

The mean square error is defined as

$$MSE = \sqrt{\sum_{i=1}^{N} \sum_{l=1}^{k} w_{il} (\boldsymbol{x}_i - \boldsymbol{c}_l)^2} \tag{33}$$

where N is the number of samples, k is the number of clusters, $\boldsymbol{c}_l$ is the center point of the lth cluster and w_{il} is defined as Eq. (27).

For a data set, the larger values of *J*, *R* and *FM* indicate better results; a small value of *MSE* means that the data is aggregated more compactly and the clustering result will be better.

5.2. The experimental results

In order to test the performance of *SNDC*, we chose 12 data sets as the experimental data sets. d is set as the half dimensionality of the raw data set. All algorithm are executed on the data sets. The results are showed as Tables 2–4 and Figs. 1–12.

Tables 2–3 are the results of the evaluation criteria. It is known that the *J* of *SNDC* is better than the comparison algorithms on 11 data sets and *SNDC* obtains *7* best results for *R*; the *FM* of *SNDC* is the best of all on 11 data sets and the *MSE* of *SNDC* is better than the comparison algorithms on 12 data sets. It can conclude that the clustering effect of *SNDC* is better than the comparison algorithms on the most data sets. Figs. 1–12 are the test results of the algorithms on each data block of the data sets. It can see that the performance of *SNDC* is stable and each performance of the comparison algorithms has a large fluctuation. The reason for the fluctuation is that there are concept drifts on data stream. It is obvious the test results of *SNDC* are better than the other algorithms from Tables 2–3. It means that the response mechanism of *SNDC* for concept drift is effective and the mechanism can improve the performance of *SNDC*. The time overhead of all algorithms is showed in Table 4. It can see that the time overhead of *SNDC* is much more than the comparison algorithms except for DenStream. Therefore *SNDC* is a time-consuming algorithm.

In order to test the effect of sliding window size on the performance of *SNDC*, we chose *Hyperplane, waveform, adult, student, magic, eye, winequality* and *occupancy* as the experimental data sets with different M values; the test results are showed as Tables 5–9.

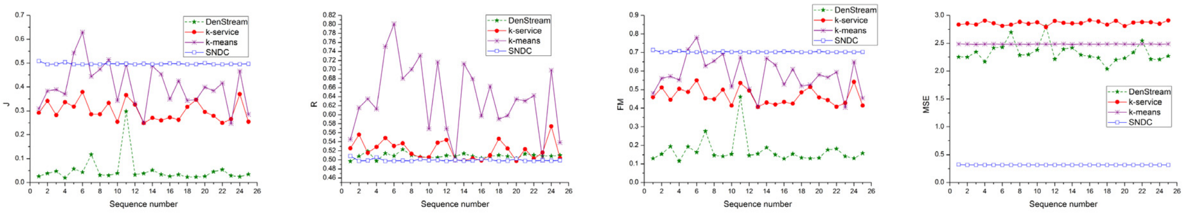
From Tables 5–8, it is known that there is a trend for the four evaluation criteria becoming better with M increasing on the some

[1] https://moa.cms.waikato.ac.nz/

[2] http://archive.ics.uci.edu/ml/datasets.php

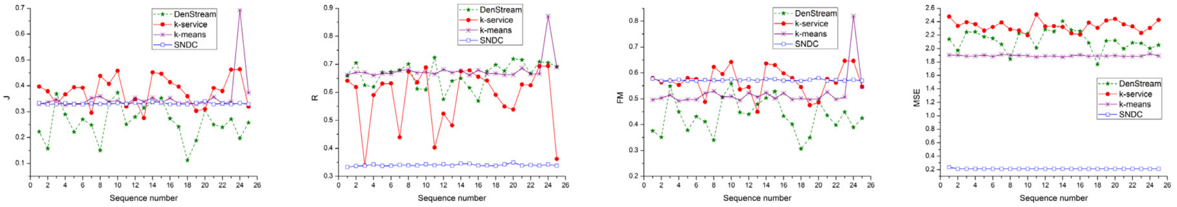
Table 3
The *FM* and *MSE* of the algorithms testing on the experimental data sets.

Data set	FM				MSE				M
	SNDC	k-means	k-service	DenStream	SNDC	k-means	k-service	DenStream	
Hyperplane	**0.7034 ± 0.0025**	0.5324 ± 0.0975	0.4646 ± 0.0445	0.2886 ± 0.0882	**0.3165 ± 0.0015**	2.4853 ± 0.0027	2.8620 ± 0.0331	2.3114 ± 0.3103	200
waveform	**0.5721 ± 0.0031**	0.5294 ± 0.0651	0.5692 ± 0.0541	0.4331 ± 0.0658	**0.2131 ± 0.0044**	1.8951 ± 0.0134	2.3373 ± 0.0807	2.1232 ± 0.1451	200
adult	**0.8025 ± 0.0284**	0.6922 ± 0.1146	0.6653 ± 0.0843	0.7797 ± 0.0507	**0.1454 ± 0.0437**	1.7829 ± 0.0153	2.0703 ± 0.0833	2.1546 ± 0.1637	200
student	**0.9848 ± 0.0357**	0.6285 ± 0.0603	0.6942 ± 0.1066	0.4173 ± 0.1208	**0.2682 ± 0.0807**	1.8448 ± 0.0828	2.1156 ± 0.1988	1.6794 ± 0.2276	200
magic	**0.9932 ± 0.0415**	0.6166 ± 0.0421	0.7861 ± 0.0804	0.8152 ± 0.2527	**0.1095 ± 0.0000**	1.4843 ± 0.0231	1.7136 ± 0.1976	1.6677 ± 0.1481	500
eye	**0.9371 ± 0.0945**	0.6553 ± 0.1084	0.7792 ± 0.1107	0.8038 ± 0.1047	**0.1994 ± 0.0025**	1.5771 ± 0.1647	1.1364 ± 0.2992	1.6841 ± 0.2171	200
winequality	**0.5697 ± 0.0231**	0.3863 ± 0.0385	0.4920 ± 0.0619	0.4507 ± 0.0971	**0.1023 ± 0.0216**	1.6637 ± 0.0156	1.9654 ± 0.1045	1.9319 ± 0.1942	500
occupancy	0.9021 ± 0.1278	0.7126 ± 0.1081	0.8961 ± 0.1224	**0.9023 ± 0.1276**	**0.0747 ± 0.0057**	0.9348 ± 0.1236	1.1282 ± 0.4456	1.1445 ± 0.4401	500
biodeg	**0.8687 ± 0.2073**	0.7057 ± 0.1055	0.7541 ± 0.1340	0.5333 ± 0.1377	**0.4937 ± 0.2828**	2.2017 ± 0.0678	2.3880 ± 0.2558	1.9488 ± 0.1212	500
Thoracic	**0.8056 ± 0.1197**	0.6492 ± 0.1346	0.6900 ± 0.0847	0.7242 ± 0.1771	**0.4381 ± 0.2267**	1.7351 ± 0.0585	2.0228 ± 0.1562	2.1278 ± 0.3009	500
Germany	**0.7142 ± 0.0966**	0.5439 ± 0.0959	0.5983 ± 0.0644	0.5851 ± 0.1273	**0.5028 ± 0.1256**	2.1025 ± 0.0137	2.4301 ± 0.0653	2.4247 ± 0.1168	40
ionosphere	**0.7447 ± 0.1151**	0.5636 ± 0.0681	0.5889 ± 0.0549	0.4416 ± 0.1389	**0.6272 ± 0.0136**	2.0507 ± 0.1685	2.3526 ± 0.2186	1.9358 ± 0.1594	45



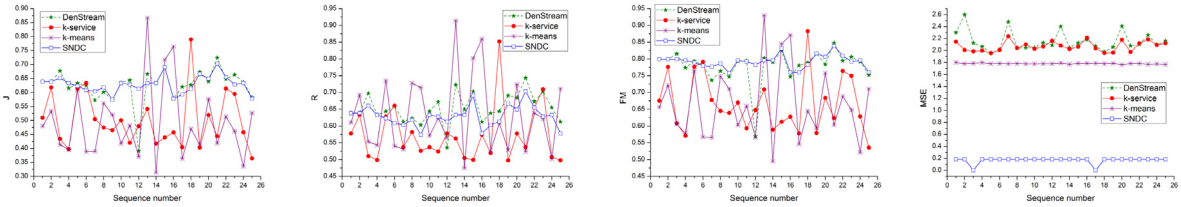
(a) J (b) R (c) FM (d) MSE

Fig. 1. The evaluation criteria of the algorithms on *Hyperplane* data set.



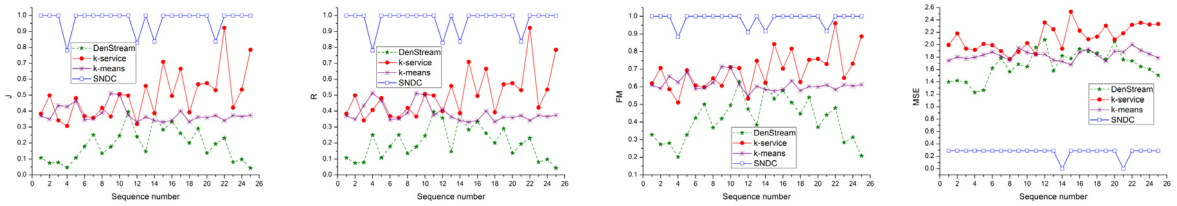
(a) J (b) R (c) FM (d) MSE

Fig. 2. The evaluation criteria of the algorithms on *waveform* data set.



(a) J (b) R (c) FM (d) MSE

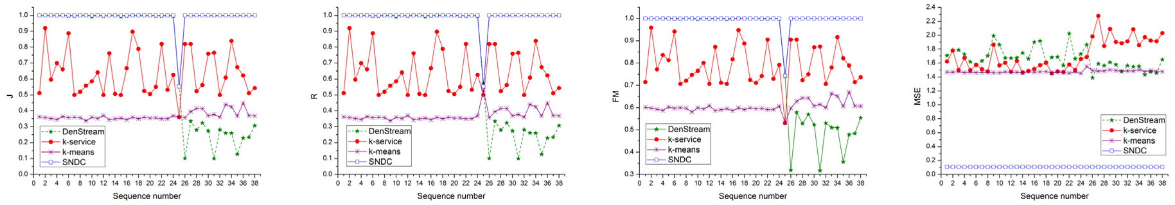
Fig. 3. The evaluation criteria of the algorithms on *adult* data set.



(a) J (b) R (c) FM (d) MSE

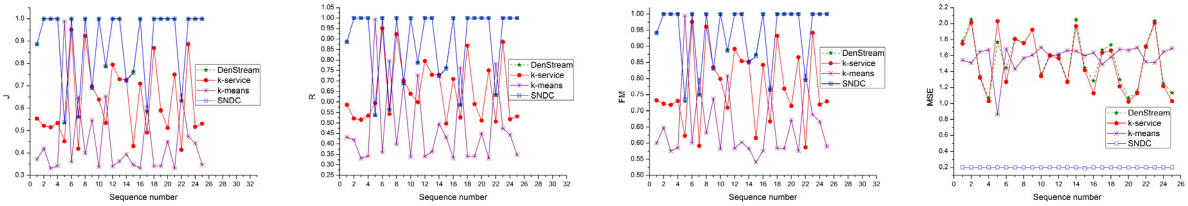
Fig. 4. The evaluation criteria of the algorithms on *student* data set.

data sets. The reason is that a large *M* leads to more samples and the center points will be selected more exactly for those data sets, therefore the evaluation criteria becomes better. If *M* exceeds the best value, the clustering performance of *SNDC* keeps stable or decreasing which can be seen on *magic, eye, winequality* and *occupancy* data sets. The reason for the decrease of the evaluation criteria is that a large *M* decreases the sensitivity of *SNDC* for concept drift. In addition, from Table 9, *SNDC* with a larger *M* spends more time on the clustering



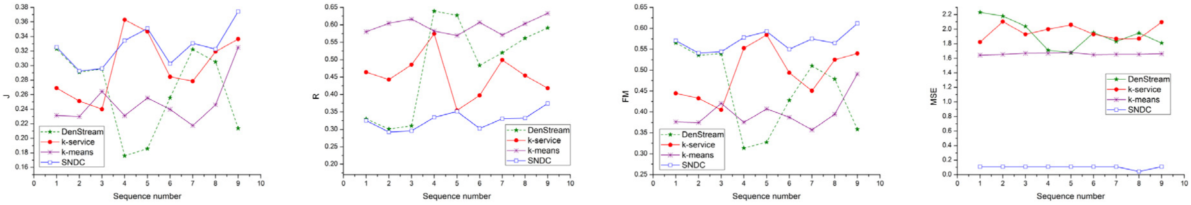
(a) J (b) R (c) FM (d) MSE

Fig. 5. The evaluation criteria of the algorithms on *magic* data set.



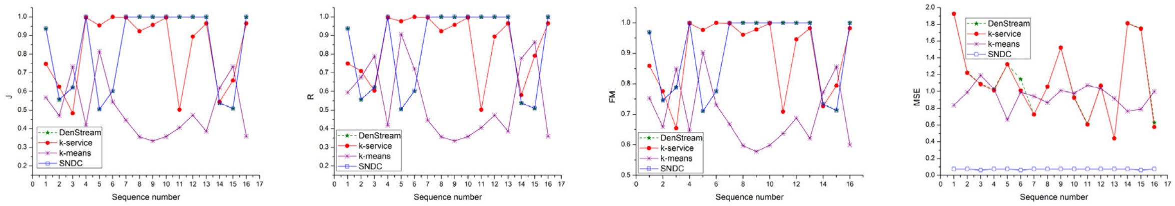
(a) J (b) R (c) FM (d) MSE

Fig. 6. The evaluation criteria of the algorithms on *eye* data set.



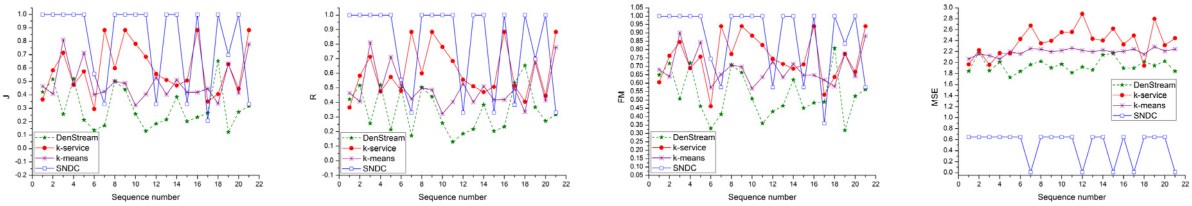
(a) J (b) R (c) FM (d) MSE

Fig. 7. The evaluation criteria of the algorithms on *winequality* data set.



(a) J (b) R (c) FM (d) MSE

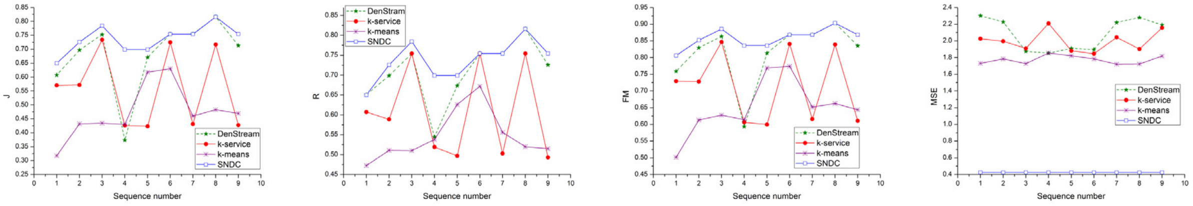
Fig. 8. The evaluation criteria of the algorithms on *occupancy* data set.



(a) J (b) R (c) FM (d) MSE

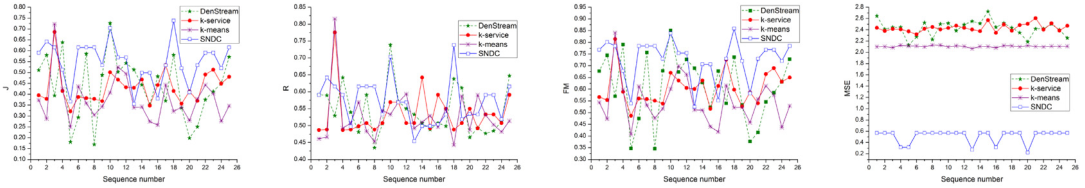
Fig. 9. The evaluation criteria of the algorithms on *biodeg* data set.

procedure and the growth rate of time overhead is greater than that of the evaluation criteria with M increasing. It can conclude that M can affect the performance of *SNDC*. Both a large M and a small M are not conducive to the approximation ability of *SNDC*; it can also infer that *SNDC* is sensitive to the scale of processing data and the size of data block can affect the speed of *SNDC*. The above results present that a large M can effect the sensibility for concept drift detection. In



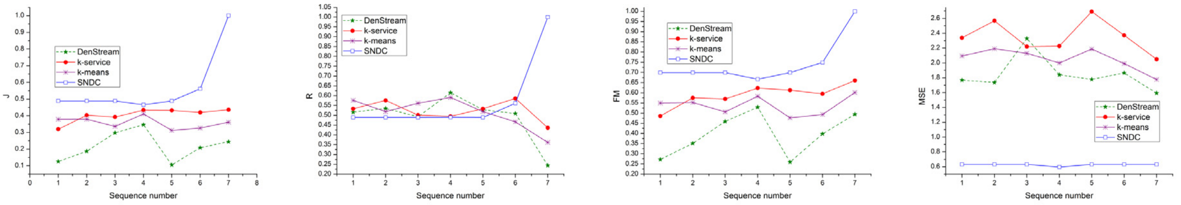
(a) J (b) R (c) FM (d) MSE

Fig. 10. The evaluation criteria of the algorithms on *Thoracic* data set.



(a) J (b) R (c) FM (d) MSE

Fig. 11. The evaluation criteria of the algorithms on *Germany* data set.



(a) J (b) R (c) FM (d) MSE

Fig. 12. The evaluation criteria of the algorithms on *ionosphere* data set.

addition, it is known from Table 9 that a large M will cost more time. Therefore an appropriate M is important for concept drift detection.

In order to test the effectiveness of dimension reduction of *SNDC*, *Hyperplane*, *waveform*, *magic*, *winequality*, *occupancy* and *adult* are selected as the experimental data sets. The *SNDC* and the *SNDC* without dimension reduction (denoted as *USNDC*) is executed on the data sets. The parameter of the algorithm is set as M=70 and the other parameters are as the experiment in Table 2. The results are showed as Tables 10–11.

From Table 10, it can be seen that the J of *SNDC* obtains the best results on 6 data sets. However *USNDC* obtains the best results only on *magic* data set. For the R index, both *SNDC* and *USNDC* obtains 4 best results on the experimental data sets. For the FM and MSE, *SNDC* is better than *USNDC* on 5 data sets. It indicates that the dimension reduction of *SNDC* improves the performance of *SNDC*. By analyzing the standard deviation of the data in Table 10, it is known that the standard deviation of *SNDC* is less than *USNDC* on the most data set and it means that the dimension reduction leads to a more stable performance for *SNDC*. Table 11 is the time overhead of the algorithms testing on the experimental data sets. It can be seen that *SNDC* has a larger cost than *USNDC*, but the difference of the time overheads is small. Therefore it can conclude that the dimension reduction mechanism is effective.

In order to test the effect of the parameter τ on the performance of *SNDC*, we choose *biodeg* as the experimental data set. We set different values for τ and *winsize*=40; the other parameters are set as the experiment in Table 2. The test result is showed as Table 12.

τ can also effect the sensibility for concept drift detection. For the two adjacent data blocks $\boldsymbol{B}_i$ and $\boldsymbol{B}_{i+1}$, if τ is a large value, abrupt concept drift can be detected only when the difference between $\boldsymbol{B}_i$ and $\boldsymbol{B}_{i+1}$ is large enough. If τ is a very small value, a small difference of two data blocks will be seen as abrupt concept drift; the proposed algorithm will detect many abrupt concept drifts in data stream and many of them are false alarms and it is not conducive to the robustness. Table 12 is the result of SNDC with different τ values. In *biodeg* data set, concept drift is abrupt concept drift. From the result, it is known that the performance of *SNDC* decreases after τ becomes a large value; it means that a large τ value can decrease the sensibility of *SNDC* for concept drift which results in many concept drifts cannot be detected. Therefore it can conclude that τ can affect the ability of concept drift detection.

Table 4
The time overhead of the algorithms on the experimental data sets.

Data set	SNDC	k-means	k-service	DenStream
Hyperplane	0.4821	0.0737	0.3177	4.8245
waveform	0.1853	0.0605	0.3374	4.7934
adult	0.1197	0.0502	0.2837	24.8349
student	0.6117	0.0578	0.2968	299.4424
magic	0.8264	0.2143	0.5131	487.2687
eye	0.0975	0.1571	0.2882	6.0726
winequality	0.1365	0.0434	0.2283	64.8211
occupancy	0.1823	0.0781	0.2743	95.4994
biodeg	0.0801	0.0201	0.2426	24.7372
Thoracic	0.0264	0.0121	0.2087	2.0852
Germany	0.0404	0.0255	0.2616	1.0018
ionosphere	0.0235	0.0069	0.2235	0.5625

In order to future test the performance of *SNDC* algorithm, we choose *PHT* algorithm as comparison algorithm (Sakamoto et al.,

Table 5
The *J* of *SNDC* testing on the experimental data sets with different *M*.

M	50	100	150	200	250	300	350	400
Hyperplane	0.4913 ± 0.0159	0.4948 ± 0.0050	0.4970 ± 0.0046	0.4972 ± 0.0031	0.4979 ± 0.0032	0.4982 ± 0.0020	0.4987 ± 0.0019	0.4990 ± 0.0018
waveform	0.3241 ± 0.0137	0.3290 ± 0.0059	0.3298 ± 0.0028	0.3316 ± 0.0032	0.3317 ± 0.0026	0.3317 ± 0.0019	0.3321 ± 0.0012	0.3329 ± 0.0017
adult	0.5739 ± 0.0869	0.6245 ± 0.0401	0.6287 ± 0.0373	0.6287 ± 0.0315	0.6291 ± 0.0300	0.6296 ± 0.0306	0.6321 ± 0.0228	0.6331 ± 0.0222
student	0.9062 ± 0.2095	0.9862 ± 0.0686	0.9750 ± 0.0815	0.9712 ± 0.0679	0.9800 ± 0.0616	0.9395 ± 0.1653	0.9437 ± 0.1447	0.9488 ± 0.1443
magic	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
eye	0.9652 ± 0.1180	0.9243 ± 0.1669	0.9168 ± 0.1601	0.8867 ± 0.1679	0.8050 ± 0.2299	0.7945 ± 0.2129	0.7339 ± 0.2279	0.7336 ± 0.2070
winequality	0.3394 ± 0.0629	0.3335 ± 0.0489	0.3311 ± 0.0415	0.3304 ± 0.0399	0.3313 ± 0.0359	0.3294 ± 0.0333	0.3259 ± 0.0301	0.3294 ± 0.0320
occupancy	0.9636 ± 0.1123	0.9445 ± 0.1308	0.9243 ± 0.1489	0.9133 ± 0.1549	0.9008 ± 0.1642	0.8925 ± 0.1610	0.8742 ± 0.1859	0.8741 ± 0.1818

Table 6
The *R* of *SNDC* testing on the experimental data sets with different *M*.

M	50	100	150	200	250	300	350	400
Hyperplane	0.5013 ± 0.0161	0.4998 ± 0.0050	0.5003 ± 0.0043	0.4996 ± 0.0027	0.4998 ± 0.0029	0.4997 ± 0.0017	0.5000 ± 0.0016	0.5002 ± 0.0016
waveform	0.3596 ± 0.0132	0.3465 ± 0.0060	0.3413 ± 0.0030	0.3401 ± 0.0034	0.3385 ± 0.0030	0.3372 ± 0.0021	0.3367 ± 0.0015	0.3370 ± 0.0020
adult	0.5969 ± 0.0719	0.6272 ± 0.0392	0.6306 ± 0.0375	0.6295 ± 0.0314	0.6298 ± 0.0310	0.6307 ± 0.0299	0.6321 ± 0.0228	0.6331 ± 0.0222
student	0.9076 ± 0.2084	0.9862 ± 0.0686	0.9750 ± 0.0815	0.9712 ± 0.0679	0.9800 ± 0.0616	0.9395 ± 0.1653	0.9437 ± 0.1447	0.9488 ± 0.1443
magic	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
eye	0.9652 ± 0.1180	0.9243 ± 0.1669	0.9168 ± 0.1601	0.8867 ± 0.1679	0.8051 ± 0.2298	0.7947 ± 0.2127	0.7342 ± 0.2276	0.7341 ± 0.2065
winequality	0.3455 ± 0.0632	0.3418 ± 0.0484	0.3397 ± 0.0409	0.3378 ± 0.0372	0.3363 ± 0.0368	0.3353 ± 0.0334	0.3303 ± 0.0312	0.3327±0.0297
occupancy	0.9636 ± 0.1123	0.9445 ± 0.1308	0.9244 ± 0.1487	0.9135 ± 0.1546	0.9009 ± 0.1640	0.8926 ± 0.1608	0.8743 ± 0.1857	0.8742 ± 0.1815

Table 7
The *FM* of *SNDC* testing on the experimental data sets with different *M*.

M	50	100	150	200	250	300	350	400
Hyperplane	0.6939 ± 0.0117	0.7000 ± 0.0039	0.7027 ± 0.0036	0.7034 ± 0.0025	0.7043 ± 0.0025	0.7047 ± 0.0016	0.7052 ± 0.0016	0.7056 ± 0.0014
waveform	0.5542 ± 0.0128	0.5661 ± 0.0055	0.5693 ± 0.0027	0.5721 ± 0.0030	0.5731 ± 0.0025	0.5735 ± 0.0019	0.5741 ± 0.0012	0.5752 ± 0.0016
adult	0.7378 ± 0.0725	0.7879 ± 0.0269	0.7913 ± 0.0245	0.7920 ± 0.0202	0.7923 ± 0.0184	0.7919 ± 0.0201	0.7949 ± 0.0143	0.7956 ± 0.0138
student	0.9435 ± 0.1262	0.9923 ± 0.0380	0.9864 ± 0.0442	0.9848 ± 0.0357	0.9894 ± 0.0325	0.9647 ± 0.0962	0.9682 ± 0.0819	0.9707 ± 0.0841
magic	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
eye	0.9801 ± 0.0679	0.9567 ± 0.0959	0.9532 ± 0.0914	0.9370 ± 0.0945	0.8876 ± 0.1333	0.8831 ± 0.1236	0.8467 ± 0.1333	0.8483 ± 0.1217
winequality	0.5779 ± 0.0548	0.5720 ± 0.0429	0.5704 ± 0.0370	0.5709 ± 0.0355	0.5727 ± 0.0309	0.5704 ± 0.0288	0.5681 ± 0.0254	0.5721 ± 0.0288
occupancy	0.9795 ± 0.0643	0.9690 ± 0.0742	0.9576 ± 0.0851	0.9515 ± 0.0884	0.9444 ± 0.0939	0.9403 ± 0.0912	0.9290 ± 0.1069	0.9293 ± 0.1041

Table 8
The *MSE* of *SNDC* testing on the experimental data sets with different *M*.

M	50	100	150	200	250	300	350	400
Hyperplane	0.6326 ± 0.0015	0.4474 ± 0.0015	0.3654 ± 0.0015	0.3165 ± 0.0015	0.2831 ± 0.0015	0.2586 ± 0.0015	0.2394 ± 0.0015	0.2240 ± 0.0015
waveform	0.4247 ± 0.0044	0.3006 ± 0.0044	0.2457 ± 0.0045	0.2130 ± 0.0044	0.1907 ± 0.0044	0.1743 ± 0.0045	0.1615 ± 0.0045	0.1513 ± 0.0045
adult	0.1908 ± 0.1879	0.2328 ± 0.0868	0.1898 ± 0.0715	0.1721 ± 0.0518	0.1506 ± 0.0515	0.1241 ± 0.0615	0.1414 ± 0.0000	0.1322 ± 0.0000
student	0.4898 ± 0.2148	0.4123 ± 0.0000	0.3264 ± 0.0586	0.2682 ± 0.0807	0.2607 ± 0.0000	0.2380 ± 0.0000	0.2203 ± 0.0000	0.2061 ± 0.0000
magic	0.3464 ± 0.0000	0.2449 ± 0.0000	0.2000 ± 0.0000	0.1732 ± 0.0000	0.1549 ± 0.0000	0.1414 ± 0.0000	0.1309 ± 0.0000	0.1224 ± 0.0000
eye	0.4000 ± 0.0000	0.2828 ± 0.0000	0.2309 ± 0.0000	0.1994 ± 0.0025	0.1777 ± 0.0035	0.1613 ± 0.0042	0.1477 ± 0.0048	0.1368 ± 0.0047
winequality	0.3305 ± 0.0644	0.2181 ± 0.0519	0.1675 ± 0.0470	0.1397 ± 0.0516	0.1301 ± 0.0382	0.1078 ± 0.0400	0.1028 ± 0.0374	0.1001 ± 0.0421
occupancy	0.2404 ± 0.0331	0.1685 ± 0.0272	0.1364 ± 0.0203	0.1191 ± 0.0081	0.0989 ± 0.0271	0.0966 ± 0.0072	0.0896 ± 0.0065	0.0834 ± 0.0065

Table 9
The time of *SNDC* testing on the experimental data sets with different *M*.

M	50	100	150	200	250	300	350	400
Hyperplane	2.4239	2.7614	3.4434	4.6501	5.7719	7.2583	10.6742	20.0493
waveform	2.5041	2.7767	3.4822	4.4630	5.6507	7.4006	10.6949	20.0654
adult	69.9086	109.3227	130.9888	204.4531	280.0690	317.7491	430.4740	472.2332
student	108.79937	175.4111	242.8118	296.7513	375.9375	430.0833	520.7356	585.2599
magic	2.3414	2.7805	3.4390	4.6427	5.7846	7.5179	10.8842	20.3370
eye	2.64	3.0259	4.4348	6.1585	8.2105	10.9931	15.2326	27.3210
winequality	9.8683	3.7547	5.6855	6.4614	8.0672	10.5668	15.0665	29.2985
occupancy	15.8801	12.704	15.69	15.0143	17.3377	21.6926	29.8374	49.5741

Table 10
The results of the algorithms testing on the experimental data sets.

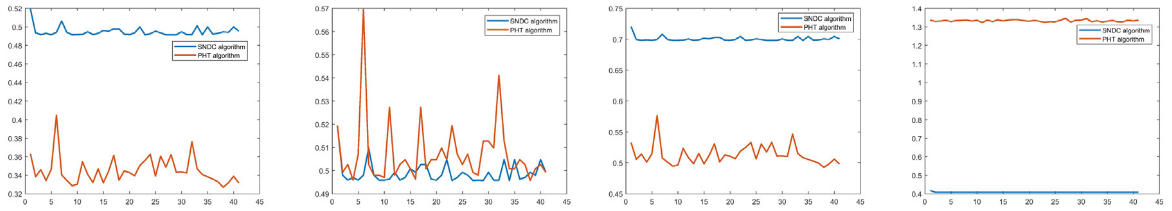
	J		R		FM		MSE	
	USNDC	SNDC	USNDC	SNDC	USNDC	SNDC	USNDC	SNDC
Hyperplane	0.4019 ± 0.0221	**0.4935 ± 0.0106**	**0.5009 ± 0.0103**	**0.5009 ± 0.0108**	0.5798 ± 0.0264	**0.6976 ± 0.0079**	2.9719 ± 0.0060	**0.5347 ± 0.0015**
waveform	0.3081 ± 0.0409	**0.3262 ± 0.0096**	**0.5146 ± 0.1025**	0.3512 ± 0.0092	0.4913 ± 0.0590	**0.5601 ± 0.0092**	2.3942 ± 0.1105	**0.3591 ± 0.0044**
magic	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	2.0471 ± 0.1260	**0.2927 ± 0.0000**
winequality	0.3250 ± 0.0547	**0.3350 ± 0.0517**	**0.3911 ± 0.1038**	0.3502 ± 0.0535	0.5485 ± 0.0671	**0.5711 ± 0.0454**	2.2188 ± 0.2265	**0.2543 ± 0.0623**
occupancy	0.9355 ± 0.1722	**0.9540 ± 0.1218**	0.9355 ± 0.1722	**0.9542 ± 0.1212**	0.9607 ± 0.1119	**0.9741 ± 0.0702**	1.9391 ± 0.2070	**0.1998 ± 0.0379**
adult	0.5130 ± 0.2297	**0.6077 ± 0.0578**	0.5771 ± 0.1097	**0.6156 ± 0.0523**	0.6731 ± 0.2378	**0.7718 ± 0.0443**	2.3261 ± 0.3600	**0.2316 ± 0.1409**

Table 11
The time overhead of the algorithms testing on the experimental data sets.

	Hyperplane	waveform	magic	winequality	occupancy	adult
USNDC	**2.0630**	**2.1749**	**1.9723**	**3.3095**	**11.4468**	**95.5186**
SNDC	2.6975	2.6759	2.5042	4.0385	13.0019	96.1279

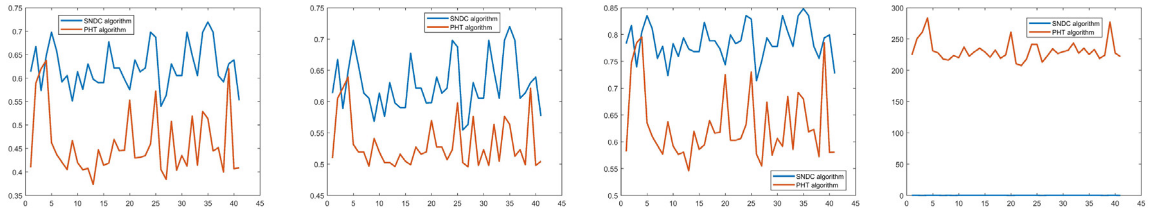
2015). *SNDC* algorithm and *PHT* algorithm are executed on *Hyperplane* and *adult* data sets. For the parameters, $M = 120$; the *PHT* algorithm's parameters are set as $\delta = 0.05$ and $\lambda = 0.2$ and the other parameters are set as the experiment in Table 2. The test results are showed in Table 13 and Figs. 13–14.

Figs. 13–14 show the test results of *SNDC* algorithm and *PHT* algorithm on each data block. Table 13 shows the mean results of *SNDC* algorithm and *PHT* algorithm on *Hyperplane* and *adult* data sets. From the results, it is known that *SNDC* algorithm outperforms *PHT* algorithm. It means that *SNDC* algorithm can effectively deal with data stream clustering with concept drift.



(a) J (b) R (c) FM (d) MSE

Fig. 13. The test result of *SNDC* algorithm and *PHT* algorithm on *Hyperplane* data set.



(a) J (b) R (c) FM (d) MSE

Fig. 14. The test results of *SNDC* algorithm and *PHT* algorithm on *adult* data set.

Table 12
The result of *SNDC* with different τ values.

	J	R	FM
τ = 0.05	0.7317 ± 0.4037	0.7317 ± 0.4037	0.7983 ± 0.3118
τ = 0.10	0.7317 ± 0.4037	0.7317 ± 0.4037	0.7983 ± 0.3118
τ = 0.30	0.7317 ± 0.4037	0.7317 ± 0.4037	0.7983 ± 0.3118
τ = 0.60	0.6007 ± 0.4434	0.6007 ± 0.4434	0.6965 ± 0.3449
τ = 1.00	0.6007 ± 0.4434	0.6007 ± 0.4434	0.6965 ± 0.3449
τ = 1.50	0.6007 ± 0.4434	0.6007 ± 0.4434	0.6965 ± 0.3449
τ = 2.00	0.5904 ± 0.4502	0.6028 ± 0.4430	0.6858 ± 0.3506
τ = 2.50	0.5904 ± 0.4502	0.6028 ± 0.4430	0.6858 ± 0.3506
τ = 3.00	0.5904 ± 0.4502	0.6028 ± 0.4430	0.6858 ± 0.3506

6. Conclusions

In this paper, a self-adaption neighborhood density clustering method for fixed data stream (*SNDC*) is proposed. For the *SNDC*, the categorical attributes can be mapped to numeric attributes; then a non-linear dimension reduction method based on neighborhood similarity is presented to decrease the dimension of data set. In the process of clustering, a neighborhood distance is defined to measure the similarity of data points. *SNDC* can automatically adjust centering points according to clustering result and concept drift can also be detected. The experimental results show that *SNDC* is effective. *SNDC* can improve the performance of the algorithm and produce a more stable result. Currently, medical data mining has an important application value and many medical data sets are mixed. In the future, we will applied *SNDC* in analyzing the clinical characteristics of diseases.

CRediT authorship contribution statement

Shuliang Xu: Methodology, Data curation, Formal analysis, Writing - original draft, Writing - review & editing. **Lin Feng:** Methodology, Supervision, Formal analysis, Funding acquisition. **Shenglan Liu:** Formal analysis, Data curation, Supervision, Validation. **Hong Qiao:** Formal analysis, Data curation, Supervision, Validation.

Acknowledgments

This work was supported by National Key Research and Development Program of China (Nos. 2017YFB1300200, 2017YFB1300203), National Natural Science Fund of China (Nos. 61972064, 61672130, 61602082, 61627808, 91648205), Open Program of State Key Laboratory of Software Architecture, China (No. SKLSAOP1701), Liaoning Revitalization Talents Program, China (No. XLYC1806006), Fundamental Research Funds for the Central Universities, China (Nos. DUT19RC(3)012, DUT17RC(3)071) and the development of science and technology of Guangdong province special fund project, China (No. 2016B090910001).

Appendix A. The proof of Theorem 1

Proof. $C \subseteq B \subseteq A \Rightarrow$ for $\forall \boldsymbol{x}_i \in U$, $n_B^{\delta}(\boldsymbol{x}_i) \subseteq n_C^{\delta}(\boldsymbol{x}_i) \Rightarrow$ for $\forall \boldsymbol{x}_j \in n_B^{\delta}(\boldsymbol{x}_i)$, $n_B^{\delta}(\boldsymbol{x}_j) \subseteq n_C^{\delta}(\boldsymbol{x}_j)$

$\Rightarrow \frac{|n_B^{\delta}(\boldsymbol{x}_j)|}{|U|} \leq \frac{|n_C^{\delta}(\boldsymbol{x}_j)|}{|U|}$ and $|U - n_B^{\delta}(\boldsymbol{x}_j)| \geq |U - n_C^{\delta}(\boldsymbol{x}_j)|$

$\Rightarrow \frac{|n_B^{\delta}(\boldsymbol{x}_j)|}{|U|} \log(\frac{|n_B^{\delta}(\boldsymbol{x}_j)|}{|U|} \cdot \frac{1}{|U-n_B^{\delta}(\boldsymbol{x}_j)|}) \leq \frac{|n_C^{\delta}(\boldsymbol{x}_j)|}{|U|} \log(\frac{|n_C^{\delta}(\boldsymbol{x}_j)|}{|U|} \cdot \frac{1}{|U-n_C^{\delta}(\boldsymbol{x}_j)|})$

$\because n_B^{\delta}(\boldsymbol{x}_i) \subseteq n_C^{\delta}(\boldsymbol{x}_i) \subseteq U \therefore |U - n_B^{\delta}(\boldsymbol{x}_j)| = |U| - |n_B^{\delta}(\boldsymbol{x}_j)| \geq |U| - |n_B^{\delta}(\boldsymbol{x}_i)| = |U - n_B^{\delta}(\boldsymbol{x}_i)|$

$\therefore -\sum_{\boldsymbol{x}_j \in n_B^{\delta}(\boldsymbol{x}_i)} \frac{|n_B^{\delta}(\boldsymbol{x}_j)|}{|U|} \log(\frac{|n_B^{\delta}(\boldsymbol{x}_j)|}{|U|} \cdot \frac{1}{|U-n_B^{\delta}(\boldsymbol{x}_j)|}) \geq -\sum_{\boldsymbol{x}_j \in n_C^{\delta}(\boldsymbol{x}_i)} \frac{|n_C^{\delta}(\boldsymbol{x}_j)|}{|U|} \log(\frac{|n_C^{\delta}(\boldsymbol{x}_j)|}{|U|} \cdot \frac{1}{|U-n_C^{\delta}(\boldsymbol{x}_j)|})$

Therefore $H_B^{\delta}(\boldsymbol{x}_i) \geq H_C^{\delta}(\boldsymbol{x}_i)$. □

Appendix B. The proof of Theorem 2

Proof. $\because \delta_1 \leq \delta_2 \therefore$ for $\forall \boldsymbol{x}_i \in U$, $n_B^{\delta_1}(\boldsymbol{x}_i) \subseteq n_B^{\delta_2}(\boldsymbol{x}_i)$

$\therefore$ for $\forall \boldsymbol{x}_j \in n_B^{\delta_1}(\boldsymbol{x}_i)$ or $\boldsymbol{x}_j \in n_B^{\delta_2}(\boldsymbol{x}_i)$, there is $n_B^{\delta_1}(\boldsymbol{x}_j) \subseteq n_B^{\delta_2}(\boldsymbol{x}_j)$

$\Rightarrow \frac{|n_B^{\delta_1}(\boldsymbol{x}_j)|}{|U|} \leq \frac{|n_B^{\delta_2}(\boldsymbol{x}_j)|}{|U|}$ and $|U - n_B^{\delta_1}(\boldsymbol{x}_j)| = |U| - |n_B^{\delta_1}(\boldsymbol{x}_j)| \geq |U| - |n_B^{\delta_2}(\boldsymbol{x}_j)| = |U - n_B^{\delta_2}(\boldsymbol{x}_j)|$

$\Rightarrow -\sum_{\boldsymbol{x}_j \in n_B^{\delta_1}(\boldsymbol{x}_i)} \frac{|n_B^{\delta_1}(\boldsymbol{x}_j)|}{|U|} \log(\frac{|n_B^{\delta_1}(\boldsymbol{x}_j)|}{|U|} \cdot \frac{1}{|U-n_B^{\delta_1}(\boldsymbol{x}_j)|}) \geq -\sum_{\boldsymbol{x}_j \in n_B^{\delta_2}(\boldsymbol{x}_i)} \frac{|n_B^{\delta_1}(\boldsymbol{x}_j)|}{|U|} \log(\frac{|n_B^{\delta_2}(\boldsymbol{x}_j)|}{|U|} \cdot \frac{1}{|U-n_B^{\delta_2}(\boldsymbol{x}_j)|})$

Therefore it can conclude $H_B^{\delta_1}(\boldsymbol{x}_i) \geq H_B^{\delta_2}(\boldsymbol{x}_i)$ □

Table 13
The mean results of *SNDC* algorithm and *PHT* algorithm on *Hyperplane* and *adult* data sets.

Data set	Algorithm	J	R	FM	MSE
Hyperplane	SNDC	**0.49450 ± 0.0051**	0.4991 ± 0.0047	**0.7007 ± 0.0040**	**0.4085 ± 0.0016**
	PHT	0.3500 ± 0.0249	**0.5097 ± 0.0195**	0.5185 ± 0.0275	1.334 ± 0.0049
adult	SNDC	**0.6243 ± 0.0458**	**0.6276 ± 0.0423**	**0.7868 ± 0.0330**	**0.2003 ± 0.0920**
	PHT	0.4596 ± 0.0689	0.5318 ± 0.0394	0.6297 ± 0.0651	231.4124 ± 16.1612

Appendix C. The proof of Theorem 3

Proof. $\because \delta = \delta_1 + \delta_2$ and $\delta_1 \geq \delta_2$ $\therefore$ for $\forall \boldsymbol{x}_i \in U$, there is $n_B^{\delta_2}(\boldsymbol{x}_i) \subseteq n_B^{\delta_1}(\boldsymbol{x}_i) \subseteq n_B^{\delta}(\boldsymbol{x}_i)$

$\therefore$ for $\forall \boldsymbol{x}_j \in n_B^{\delta}(\boldsymbol{x}_i)$, there are $n_B^{\delta_2}(\boldsymbol{x}_j) \subseteq n_B^{\delta_1}(\boldsymbol{x}_j) \subseteq n_B^{\delta}(\boldsymbol{x}_j)$.

$\therefore \frac{\left|n_B^{\delta}(\boldsymbol{x}_j)\right|}{|U|} \geq \frac{\left|n_B^{\delta_1}(\boldsymbol{x}_j)\right|}{|U|} \geq \frac{\left|n_B^{\delta_2}(\boldsymbol{x}_j)\right|}{|U|}$

$\therefore \left|U - n_B^{\delta}(\boldsymbol{x}_j)\right| = |U| - \left|n_B^{\delta}(\boldsymbol{x}_j)\right| \geq \left|U - n_B^{\delta_1}(\boldsymbol{x}_j)\right| = |U| - \left|n_B^{\delta_1}(\boldsymbol{x}_j)\right| \geq \left|U - n_B^{\delta_2}(\boldsymbol{x}_j)\right| = |U| - \left|n_B^{\delta_2}(\boldsymbol{x}_j)\right|$

Let $n_B^{\delta}(\boldsymbol{x}_i) = n_B^{\delta_1}(\boldsymbol{x}_i) \cup \Delta x$ and $n_B^{\delta_1}(\boldsymbol{x}_i) \cap \Delta x = \varnothing$ $\therefore$ $\left|n_B^{\delta_2}(\boldsymbol{x}_i)\right| \leq |\Delta x|$.

$\because \boldsymbol{x}_j \in n_B^{\delta}(\boldsymbol{x}_i) - n_B^{\delta_1}(\boldsymbol{x}_i)$, $\boldsymbol{x}_{j'} \in n_B^{\delta_2}(\boldsymbol{x}_i)$ and $n_B^{\delta_2}(\boldsymbol{x}_{j'}) \subseteq n_B^{\delta_2}(\boldsymbol{x}_j)$

$$\therefore \sum_{\boldsymbol{x}_j \in n_B^{\delta}(\boldsymbol{x}_i)} \frac{\left|n_B^{\delta}(\boldsymbol{x}_j)\right|}{|U|} \log\Big(\frac{\left|n_B^{\delta}(\boldsymbol{x}_j)\right|}{|U|} \cdot \frac{1}{\left|U - n_B^{\delta}(\boldsymbol{x}_j)\right|}\Big) = \sum_{\boldsymbol{x}_j \in n_B^{\delta_1}(\boldsymbol{x}_i) \cup \Delta x} \frac{\left|n_B^{\delta}(\boldsymbol{x}_j)\right|}{|U|} \log\Big(\frac{\left|n_B^{\delta}(\boldsymbol{x}_j)\right|}{|U|} \cdot \frac{1}{\left|U - n_B^{\delta}(\boldsymbol{x}_j)\right|}\Big)$$

$$\geq \sum_{\boldsymbol{x}_j \in n_B^{\delta_1}(\boldsymbol{x}_i)} \frac{\left|n_B^{\delta_1}(\boldsymbol{x}_j)\right|}{|U|} \log\Big(\frac{\left|n_B^{\delta_1}(\boldsymbol{x}_j)\right|}{|U|} \cdot \frac{1}{\left|U - n_B^{\delta_1}(\boldsymbol{x}_j)\right|}\Big) + \sum_{\boldsymbol{x}_j \in \Delta x} \frac{\left|n_B^{\delta_1}(\boldsymbol{x}_j)\right|}{|U|} \log\Big(\frac{\left|n_B^{\delta_1}(\boldsymbol{x}_j)\right|}{|U|} \cdot \frac{1}{\left|U - n_B^{\delta_1}(\boldsymbol{x}_j)\right|}\Big)$$

$$\geq \sum_{\boldsymbol{x}_j \in n_B^{\delta_1}(\boldsymbol{x}_i)} \frac{\left|n_B^{\delta_1}(\boldsymbol{x}_j)\right|}{|U|} \log\Big(\frac{\left|n_B^{\delta_1}(\boldsymbol{x}_j)\right|}{|U|} \cdot \frac{1}{\left|U - n_B^{\delta_1}(\boldsymbol{x}_j)\right|}\Big) + \sum_{\boldsymbol{x}_j \in \Delta x} \frac{\left|n_B^{\delta_2}(\boldsymbol{x}_j)\right|}{|U|} \log\Big(\frac{\left|n_B^{\delta_2}(\boldsymbol{x}_j)\right|}{|U|} \cdot \frac{1}{\left|U - n_B^{\delta_2}(\boldsymbol{x}_j)\right|}\Big)$$

$$\geq \sum_{\boldsymbol{x}_j \in n_B^{\delta_1}(\boldsymbol{x}_i)} \frac{\left|n_B^{\delta_1}(\boldsymbol{x}_j)\right|}{|U|} \log\Big(\frac{\left|n_B^{\delta_1}(\boldsymbol{x}_j)\right|}{|U|} \cdot \frac{1}{\left|U - n_B^{\delta_1}(\boldsymbol{x}_j)\right|}\Big) + \sum_{\boldsymbol{x}_j \in n_B^{\delta_2}(\boldsymbol{x}_i)} \frac{\left|n_B^{\delta_2}(\boldsymbol{x}_j)\right|}{|U|} \log\Big(\frac{\left|n_B^{\delta_2}(\boldsymbol{x}_j)\right|}{|U|} \cdot \frac{1}{\left|U - n_B^{\delta_2}(\boldsymbol{x}_j)\right|}\Big)$$

$$\therefore -\sum_{\boldsymbol{x}_j \in n_B^{\delta}(\boldsymbol{x}_i)} \frac{\left|n_B^{\delta}(\boldsymbol{x}_j)\right|}{|U|} \log\Big(\frac{\left|n_B^{\delta}(\boldsymbol{x}_j)\right|}{|U|} \cdot \frac{1}{\left|U - n_B^{\delta}(\boldsymbol{x}_j)\right|}\Big) \leq -\sum_{\boldsymbol{x}_j \in n_B^{\delta_1}(\boldsymbol{x}_i)} \frac{\left|n_B^{\delta_1}(\boldsymbol{x}_j)\right|}{|U|} \log\Big(\frac{\left|n_B^{\delta_1}(\boldsymbol{x}_j)\right|}{|U|} \cdot \frac{1}{\left|U - n_B^{\delta_1}(\boldsymbol{x}_j)\right|}\Big)$$

$$-\sum_{\boldsymbol{x}_j \in n_B^{\delta_2}(\boldsymbol{x}_i)} \frac{\left|n_B^{\delta_2}(\boldsymbol{x}_j)\right|}{|U|} \log\Big(\frac{\left|n_B^{\delta_2}(\boldsymbol{x}_j)\right|}{|U|} \cdot \frac{1}{\left|U - n_B^{\delta_2}(\boldsymbol{x}_j)\right|}\Big)$$

$\therefore H_B^{\delta}(\boldsymbol{x}_i) \leq H_B^{\delta_1}(\boldsymbol{x}_i) + H_B^{\delta_2}(\boldsymbol{x}_i)$. □

Appendix D. The proof of Proposition 2

Proof. Property 1 and 2 are obviously established.

$d_B^{\delta}(\boldsymbol{x}, \boldsymbol{y}) = \left|\frac{1}{\left|n_B^{\delta}(\boldsymbol{x})\right|} \cdot H_B^{\delta}(\boldsymbol{x}) - \frac{1}{\left|n_B^{\delta}(\boldsymbol{y})\right|} \cdot H_B^{\delta}(\boldsymbol{y})\right|$

It is known from the absolute value inequality: $|a| + |b| \geq |a - b|$

$$d_B^{\delta}(\boldsymbol{x}, \boldsymbol{z}) + d_B^{\delta}(\boldsymbol{y}, \boldsymbol{z}) = \left|\frac{1}{\left|n_B^{\delta}(\boldsymbol{x})\right|} \cdot H_B^{\delta}(\boldsymbol{x}) - \frac{1}{\left|n_B^{\delta}(\boldsymbol{z})\right|} \cdot H_B^{\delta}(\boldsymbol{z})\right| + \left|\frac{1}{\left|n_B^{\delta}(\boldsymbol{z})\right|} \cdot H_B^{\delta}(\boldsymbol{z}) - \frac{1}{\left|n_B^{\delta}(\boldsymbol{y})\right|} \cdot H_B^{\delta}(\boldsymbol{y})\right|$$

$$= \left|\frac{1}{\left|n_B^{\delta}(\boldsymbol{x})\right|} \cdot H_B^{\delta}(\boldsymbol{x}) - \frac{1}{\left|n_B^{\delta}(\boldsymbol{z})\right|} \cdot H_B^{\delta}(\boldsymbol{z})\right| + \left|\frac{1}{\left|n_B^{\delta}(\boldsymbol{y})\right|} \cdot H_B^{\delta}(\boldsymbol{y}) - \frac{1}{\left|n_B^{\delta}(\boldsymbol{z})\right|} \cdot H_B^{\delta}(\boldsymbol{z})\right|$$

$$\geq \left|\frac{1}{\left|n_B^{\delta}(\boldsymbol{x})\right|} \cdot H_B^{\delta}(\boldsymbol{x}) - \frac{1}{\left|n_B^{\delta}(\boldsymbol{z})\right|} \cdot H_B^{\delta}(\boldsymbol{z}) + \frac{1}{\left|n_B^{\delta}(\boldsymbol{z})\right|} \cdot H_B^{\delta}(\boldsymbol{z}) - \frac{1}{\left|n_B^{\delta}(\boldsymbol{y})\right|} \cdot H_B^{\delta}(\boldsymbol{y})\right|$$

$$= \left|\frac{1}{\left|n_B^{\delta}(\boldsymbol{x})\right|} \cdot H_B^{\delta}(\boldsymbol{x}) - \frac{1}{\left|n_B^{\delta}(\boldsymbol{y})\right|} \cdot H_B^{\delta}(\boldsymbol{y})\right| = d_B^{\delta}(\boldsymbol{x}, \boldsymbol{y}) \Rightarrow d_B^{\delta}(\boldsymbol{x}, \boldsymbol{y}) \leq d_B^{\delta}(\boldsymbol{x}, \boldsymbol{z}) + d_B^{\delta}(\boldsymbol{z}, \boldsymbol{y}).$$ □

Appendix E. The proof of Theorem 4

Proof. For a sliding window, let $I = \int_1^M \frac{\lambda}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \mathrm{d}x = \int_1^M \frac{\lambda}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \mathrm{d}y$

$\Rightarrow I^2 = \int_1^M \int_1^M \frac{\lambda^2}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dxdy$

$= \int_1^M \int_1^M \frac{\lambda^2}{2\pi} e^{-\frac{x^2}{2}} e^{-\frac{y^2}{2}} \mathrm{d}x\mathrm{d}y = \int_1^M \int_1^M \frac{\lambda^2}{2\pi} e^{-\frac{x^2+y^2}{2}} \mathrm{d}x\mathrm{d}y$

Let $x = \rho\cos\theta$, $y = \rho\sin\theta \Rightarrow \mathrm{d}x\mathrm{d}y = \rho\mathrm{d}\rho\mathrm{d}\theta$, $0 \leq \rho \leq M$ and $0 \leq \theta \leq \pi$;

$I^2 = \frac{\lambda^2}{2\pi} \int_0^{\pi} \int_1^M e^{\frac{-r^2}{2}} r\mathrm{d}\theta\mathrm{d}r = \frac{\lambda^2}{2\pi} \cdot \pi \int_1^M e^{\frac{-r^2}{2}} r\mathrm{d}r = \frac{\lambda^2}{2} \int_1^M e^{\frac{-r^2}{2}} r\ \mathrm{d}r = \frac{\lambda^2}{2}(e^{-\frac{1}{2}} - e^{-\frac{M^2}{2}})$ $\because I^2 \leq 1$

$\therefore \lambda^2 e^{-\frac{1}{2}} - \lambda^2 e^{-\frac{M^2}{2}} \leq 2 \Rightarrow e^{-\frac{M^2}{2}} \geq \frac{\lambda^2 e^{-\frac{1}{2}} - 2}{\lambda^2} \Rightarrow \frac{M^2}{2} \geq \ln(\frac{\lambda^2 e^{-\frac{1}{2}} - 2}{\lambda^2}) \Rightarrow M \leq \sqrt{-2\ln(\frac{\lambda^2 e^{-\frac{1}{2}} - 2}{\lambda^2})}$

$\therefore \frac{\lambda^2 e^{-\frac{1}{2}} - 2}{\lambda^2} \leq 1$ and $\lambda^2 e^{-\frac{1}{2}} - 2 \geq 0 \Rightarrow \sqrt{\frac{2}{e^{-\frac{1}{2}}}} \leq \lambda \leq \sqrt{\frac{2}{e^{-\frac{1}{2}} - 1}}$. □

References

Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J., 2002. Models and issues in data stream systems. In: ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. pp. 1–16.

Bai, L., Cheng, X., Liang, J., Shen, H., 2016. An optimization model for clustering categorical data streams with drifting concepts. IEEE Trans. Knowl. Data Eng. 28 (11), 2871–2883.

Belkin, M., Niyogi, P., 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. Adv. Neural Inf. Process. Syst. 14 (6), 585–591.

Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B., 2010. MOA: Massive online analysis. J. Mach. Learn. Res. 11 (2), 1601–1604.

Braverman, V., Meyerson, A., Ostrovsky, R., Roytman, A., Shindler, M., Tagiku, B., 2011. Streaming k-means on well-clusterable data. In: ACM-SIAM Symposium on Discrete Algorithms. pp. 26–40.

Cao, F., Ester, M., Qian, W., Zhou, A., 2006. Density-based clustering over an evolving data stream with noise. In: SIAM International Conference on Data Mining, April 20-22, 2006, Bethesda, MD, USA. pp. 328–339.

Cao, F., Liang, J., Bai, L., Zhao, X., Dang, C., 2010. A framework for clustering categorical time-evolving data. IEEE Trans. Fuzzy Syst. 18 (5), 872–882.

Chen, Y., Tu, L., 2007. Density-based clustering for real-time stream data. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August. pp. 133–142.

Chen, Y., Wu, K., Chen, X., Tang, C., Zhu, Q., 2014. An entropy-based uncertainty measurement approach in neighborhood systems. Inform. Sci. 279, 239–250.

Comaniciu, D., Meer, P., 2002. Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24 (5), 603–619.

Dijkstra, E.W., 1959. A note on two problems in connection with graphs. Numer. Math. 1 (1), 269–271.

Ding, S., Wu, F., Qian, J., Jia, H., Jin, F., 2015. Research on data stream clustering algorithms. Artif. Intell. Rev. 43 (4), 593–600.

Frey, B.J., Dueck, D., 2007. Clustering by passing messages between data points. Science 315 (5814), 972–976.

Hahsler, M., Bolanos, M., 2016. Clustering data streams based on shared density between micro-clusters. IEEE Trans. Knowl. Data Eng. 28 (6), 1449–1461.

Hu, Q., Yu, D., Liu, J., Wu, C., 2008. Neighborhood rough set based heterogeneous feature subset selection. Inform. Sci. 178, 3577–3594.

Jiang, W., Brice, P., 2009. Data stream clustering and modeling using context-trees. In: International Conference on Service Systems and Service Management. pp. 932–937.

Jiawei, H., Micheline, K., Jian, P., 2012. Data Mining: Concepts and Techniques, third ed. Morgan Kaufmann Publishers.

Kaur, S., Bhatnagar, V., Chakravarthy, S., 2015. Stream Clustering Algorithms: A Primer. Springer International Publishing, pp. 105–145.

Kiwiel, K.C., 2001. Convergence and efficiency of subgradient methods for quasiconvex minimization. Math. Program. 90 (1), 1–25.

Krishnaswamy, S., 2005. Mining data streams: A review. ACM SIGMOD Rec. 34 (2), 18–26.

Li, Y., Li, D., Wang, S., Zhai, Y., 2014. Incremental entropy-based clustering on categorical data streams with concept drift. Knowl.-Based Syst. 59 (2), 33–47.

Liang, J., Qian, Y., 2008. Information granules and entropy theory in information systems. Sci. China Inf. Sci. 51 (10), 1427–1444.

Liang, J., Wang, J., Qian, Y., 2009. A new measure of uncertainty based on knowledge granulation for rough sets. Inform. Sci. 179 (4), 458–470.

Maji, P., Pal, S.K., 2007. Rough-fuzzy c-medoids algorithm and selection of bio-basis for amino acid sequence analysis. IEEE Trans. Knowl. Data Eng. 19 (6), 859–872.

Mi, J.S., Wu, W.Z., Zhang, W.X., 2004. Approaches to knowledge reduction based on variable precision rough set model. Inform. Sci. 159, 255–272.

Sakamoto, Y., Fukui, K.-I., Gama, J., Nicklas, D., Moriyama, K., Numao, M., 2015. Concept drift detection with clustering via statistical change detection methods. In: 2015 Seventh International Conference on Knowledge and Systems Engineering, KSE. IEEE, pp. 37–42.

Shindler, M., Wong, A., Meyerson, A., 2011. Fast and accurate k-means for large datasets. In: International Conference on Neural Information Processing Systems. pp. 2375–2383.

Silva, J.A., Faria, E.R., Barros, R.C., Hruschka, E.R., 2013. Data stream clustering: A survey. ACM Comput. Surv. 46 (1), 1–31.

Song, J., Tsang, E.C.C., Chen, D., Yang, X., 2017. Minimal decision cost reduct in fuzzy decision-theoretic rough set model. Knowl.-Based Syst. 126, 104–112.

Swiniarski, R.W., Skowron, A., 2003. Rough set methods in feature selection and recognition. Pattern Recognit. Lett. 24 (6), 833–849.

Xu, S., Wang, J., 2016. A fast incremental extreme learning machine algorithm for data streams classification. Expert Syst. Appl. 65, 332–344.

Xu, S., Wang, J., 2017. Dynamic extreme learning machine for data stream classification. Neurocomputing 238, 433–449.

Yao, Y., Yao, B., 2012. Covering based rough set approximations. Inform. Sci. 200, 91–107.

Zhang, X., 2013. Matrix Analysis and Application, second ed. Tsinghua University Press.

Zhang, X., Furtlehner, C., Germainrenaud, C., Sebag, M., 2014a. Data stream clustering with affinity propagation. IEEE Trans. Knowl. Data Eng. 26 (7), 1644–1656.

Zhang, J., Li, T., Da, R., Liu, D., 2014b. Neighborhood rough sets for dynamic data mining. Inform. Sci. 257, 81–100.

Zhang, J., Zhu, Y., Pan, Y., Li, T., 2016. Efficient parallel boolean matrix based algorithms for computing composite rough set approximations. Inform. Sci. 329, 287–302.