Pattern Recognition 86 (2019) 48–61

Contents lists available at ScienceDirect

Pattern Recognition

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

OFS-Density: A novel online streaming feature selection method

Peng Zhou[a], Xuegang Hu[a,*], Peipei Li[a], Xindong Wu[b]

[a] Hefei University of Technology, Hefei 230009, China
[b] University of Louisiana, lafayette, LA 70504, USA

ARTICLE INFO

Article history:
Received 13 December 2017
Revised 20 June 2018
Accepted 27 August 2018
Available online 1 September 2018

Keywords:
Feature selection
Online feature selection
Streaming features
Neighborhood rough set

ABSTRACT

Online streaming feature selection which deals with streaming features in an online manner plays a critical role in big data problems. Many approaches have been proposed to handle this problem. However, most existing methods need domain information before learning and specify some parameters in advance. In real-world applications, we cannot always require the domain information and it is a big challenge to specify uniform parameters for all different types of data sets. Motivated by this, we propose a new online streaming feature selection method based on adaptive density neighborhood relation, named OFS-Density. More specifically, with the neighborhood rough set theory, OFS-Density does not require the domain information before learning. Meanwhile, we propose a new adaptive neighborhood relation using the density information of the surrounding instances, which does not need to specify any parameters in advance. By the fuzzy equal constraint, OFS-Density can select features with a low redundancy. Finally, experimental studies on fourteen datasets show that OFS-Density is superior to traditional feature selection methods with the same numbers of features and state-of-the-art online streaming feature selection algorithms in an online manner.

1. Introduction

Feature selection aims to select a subset of feature space from the original data set which is “as good as possible”. It plays an important role in machine learning and pattern recognition [1]. The main task of feature selection is to remove irrelevant and redundant features from the feature space. There are many benefits from feature selection, such as reducing storage requirements and training time, facilitating data visualization and improving predictive accuracy [2].

With the increase of the data volume and dimensionality, traditional feature selection methods cannot fit the demand in efficiency any more [3]. Online streaming feature selection which deals with streaming features has attracted much attention in recent years [3–11]. Streaming features are defined as features that flow in one by one over time whereas the number of training examples is fixed [6]. As the features flow in one by one over time, we must decide whether to keep or discard the new arriving feature at each time stamp and we do not know the information of whole feature space before learning. There are two major reasons for online streaming feature selection: 1) The feature space is unknown or even infinite and 2) the feature space is known but feature streaming offers many advantages [9]. However, most of existing online streaming feature selection methods need domain information before learning and need specifying some parameters in advance. For instance, Grafting [4] needs to specify the parameter λ before learning and a proper value of λ is important to the final predictive accuracy. However, in real-world applications, we can not require the domain information before learning. Meanwhile, it is a challenge to specify uniform parameters for all types of data sets and it is infeasible to specify different parameters for different data sets. Motivated by this, we design a new online streaming feature selection method which need not any domain information and does not need to specify any parameters before learning.

Rough set theory [12], as an effective tool for feature selection, rule extraction, and knowledge discovery can provide an important advantage, that is, rough set-based data mining does not require any domain knowledge other than the given dataset. For example, OS-NRRSARA-SA [9] is a classical rough set based online streaming feature selection method which need not specify any parameters in advance. However, OS-NRRSARA-SA cannot deal with numerical data directly, because the classical Rough Set theory is originally proposed to deal with categorical data. In real-world applications, there are many numerical features in the data sets. Thus, fuzzy rough set [13–18] and neighborhood rough set [19–21] which supports both continuous and discrete data was proposed to deal with this challenge. OFS-A3M [10] based on neighborhood rough

[*] Corresponding author.
E-mail addresses: doodzhou@hotmail.com (P. Zhou), jsjxhuxg@hfut.edu.cn (X. Hu), peipeili@hfut.edu.cn (P. Li), xwu@louisiana.edu (X. Wu).

https://doi.org/10.1016/j.patcog.2018.08.009
0031-3203/© 2018 Elsevier Ltd. All rights reserved.

set theory is a new method for online streaming feature selection. With a new GAP neighborhood relation, OFS-A3M does not require domain information before learning and need not specify any parameters in advance. Nevertheless, OFS-A3M uses exact dependency equal constraint for feature redundancy analysis, which is too strict for real data sets and leads to some redundant features in the selected feature subset.

Therefore, in this paper, we propose a new online streaming feature selection method, named OFS-Density. Our contributions are as follows:

- Based on neighborhood rough set, OFS-Density does not require domain information before learning. As we know that, rough set-based data mining does not require any domain knowledge.
- We propose a new neighborhood relation which using the density information of the surrounding instances. With this new density neighborhood relation, OFS-Density can automatically select a proper number of neighbors during online feature selection. Therefore, OFS-Density need not specify any parameters in advance.
- OFS-Density uses a fuzzy equal constraint for redundant analysis to make the selected feature subset with low redundancy. Rough set based feature selection methods always use the condition of feature significance equal to zero for feature redundant analysis. However, in real data sets, the exactly equal constraint is too strict. With fuzzy equal constraint, OFS-Density can consider more candidate features for feature redundancy analysis which makes the final selected feature subset small and discriminative. The parameter λ used for fuzzy equal constraint makes OFS-Density can consider more features for redundant analysis. Meanwhile, in Section 5.2, we conclude that the λ value is not the bigger the better and it is set to $\lambda = 0.05$ for OFS-Density.
- Extensive experimental studies of eight traditional feature selection methods and seven online streaming feature selection approaches show that our proposed algorithm can get better performance than traditional feature selection methods with the same number of features and state-of-the-art online streaming feature selection approaches in an online manner.

The remainder of the paper is organized as follows. Section 2 discusses related work. Section 3 gives a brief introduction to neighborhood rough set theory. Section 4 presents our new proposed method for streaming feature selection. Section 5 reports experimental results and Section 6 concludes the paper.

2. Related work

In this section, we give an introduction to some representative traditional feature selection methods and the state-of-the-art online streaming feature selection algorithms.

Feature selection can have numerous benefits such as faster model training, reduced susceptibility to overfitting, offsetting the pernicious effects of the curse of dimensionality, and reducing storage, memory, and processing requirements during data analysis [1]. According to how the label information is used, feature selection algorithms can be divided into supervised [22–24], unsupervised [25] and semi-supervised [26] ones. More specifically, Fisher Score [27] computes a score for each feature as the ratio of inter-class separation and intra-class variance. ReliefF [28] estimates the quality of the features according to how well their values differentiate data samples that are near to each other. MI [29] considers the mutual information between the distribution of the values of a given feature and the membership to a particular class. FSV [30] is a wrapper method, where the feature selection process is injected into the training of an SVM by a linear programming technique. Laplacian Score [31] does not use the class information of each instance, and the importance of a feature is evaluated by its power of locality preserving. In order to model the local geometric structure, this method constructs a nearest neighbor graph. Laplacian Score algorithm seeks those features that respect this graph structure. INF [32] is an unsupervised graph-based filter method. In the INF formulation, each feature is a node in the graph, a path is a selection of features. The higher the centrality score, the more important (or more different) the feature. It assigns a score of importance to each feature by taking into account all possible feature subsets as paths on a graph. LLC-FS [33] associates a weight to each feature or kernel and incorporates it into the built-in regularization of the LLC [34] algorithm to take into account the relevance of each feature or kernel for the clustering. Correspondingly, the weights are estimated iteratively in the clustering process. Then, the weights of those irrelevant features or kernels can be shrunk to zero.

Traditional feature selection methods assume that all features in the feature space are available before learning. However, in some real-world applications such as [35,36], features may exist in a streaming format. Online streaming feature selection which deals with feature streams in an online manner, has attracted much attention in recent years and played a critical role in dealing with high-dimensional problems [3–7,9].

More specifically, Perkins and Theiler [4] proposed the Grafting algorithm based on a stagewise gradient descent approach for online feature selection. Grafting treats feature selection as an integral part of learning a predictor within a regularized framework. If the improvement of adding a new arriving feature in the model is greater than a predefined threshold λ, this new arriving feature will be selected. Grafting needs the information of the global feature space to choose a good value for the important regularization parameter λ in advance. Zhou et al. [5] proposed the Alpha-investing algorithm based on streamwise regression for online streaming feature selection. Alpha-investing does not need a global model and it is one of the penalized likelihood ratio methods. Nevertheless, Alpha-investing requires prior knowledge of the structure of the feature space to heuristically control the choice of candidate feature selection. Wu et al. [6] presented an online streaming feature selection framework with two algorithms called OSFS (Online Streaming Feature Selection) and fast-OSFS. There are two major steps in OSFS, including online relevance analysis (discards irrelevant features) and online redundancy analysis (eliminates redundant features). OSFS uses the conditional independence test for feature selection which needs a large number of training instances. Thus, on the datasets with high dimensionality and small samples, this may lead to information missing. Yu et al. [3] proposed a Scalable and Accurate Online feature selection Approach (SAOLA) for extremely high dimensional datasets. SAOLA employs novel online pairwise comparison techniques and maintains a parsimonious model over time in an online manner. SAOLA needs to specify a relevance threshold α in advance to determine whether two features are relevant, although the relevance thresholds do not have a significant impact on the algorithm.

In addition, rough set theory, proposed by Pawlak [12], is an effective tool for feature selection, rule extraction, and knowledge discovery. Rough set based data mining does not require any domain knowledge before learning. There are some research works of online streaming feature selection by using Rough set theory. More specifically, Eskandari et al.[9] proposed a classical Rough Set based method (OS-NRRSARA-SA) for online streaming feature selection. OS-NRRSARA-SA uses classical significance analysis concepts in Rough Set theory to control an unknown feature space in online streaming feature selection problems. OS-NRRSARA-SA need not specify any parameters before learning. However, OS-NRRSARA-SA is a classical Rough Set based method which cannot handle numerical features directly. Zhou et al. [10] proposed a new online streaming feature selection method OFS-A3M based on a new neighborhood rough set relation with adapted neighbors. With the maximal-dependency, maximal-relevance and maximal-significance evaluation criteria, OFS-A3M can select features with high correlation, high dependency, and low redundancy. There are mainly two differences between this work and OFS-A3M. First, OFS-A3M proposed a new neighborhood relation named as Gap, which uses the gap information of the neighbors. In this paper, we proposed a new neighborhood relation called Density, which uses the density information of the neighbors. Density-based neighborhood relation uses the information of surrounding neighbors as a whole, while Gap just uses the distance information of the previous instance and next instance. Second, OFS-A3M uses exact dependency equal constraint for the analysis of feature redundancy. It is too strict for real-world data sets, and it probably leads to some redundant features in the selected feature subset. In this manuscript, we use a fuzzy equal constraint for redundant analysis, which makes the selected feature subset present lower redundancy than OFS-A3M. Zhou et al. [11] proposed a new online streaming feature selection method for high-dimensional and class-imbalanced data, called K-OFSD. K-OFSD uses the dependency between condition features and decision classes for feature selection. In terms of Neighborhood Rough Set theory, K-OFSD uses the information of K nearest neighbors to select relevant features which can get higher separability between the majority class and the minority class. K-OFSD is designed for class-imbalanced data and it needs to specify the parameter K in advance. Diao et al. [37] proposed new methods to carry out the online selection with incrementally changing on features or instances. Four possible dynamic selection scenarios (feature addition, feature removal, instance addition, instance removal) are considered, with algorithms proposed in order to handle such individual situations. Based on fuzzy-rough sets theory, the proposed methods need not specify any parameters before learning and are demonstrated to be effective in dealing with real-world data sets.

3. Neighborhood rough set

In the classical rough set model [12], the objects with the same feature values in terms of attributes B are drawn together and form an equivalence class, denoted by $[x]_B$. The family of elemental granules $\{[x_i]_B | x_i \in U\}$ builds a concept system to describe an arbitrary subset of the sample space, where $U = \{x_1, x_2, \ldots, x_n\}$ is a nonempty finite set of objects, called a universe. For subset X, two unions of elemental granules: lower approximation and upper approximation are defined as follow:

$$\underline{B}X = \{[x_i]_B \mid [x_i]_B \subseteq X, x_i \in U\} \tag{1}$$

$$\overline{B}X = \{[x_i]_B \mid [x_i]_B \cap X \neq \emptyset, x_i \in U\} \tag{2}$$

The lower approximation is the maximal union of elemental granules consistently contained in X, while the upper approximation is the minimal union of elemental granules containing X. The difference between lower approximation and upper approximation is called approximation boundary of X: $BN(X) = \overline{B}X - \underline{B}X$. The lower approximation is also called positive region. The positive region, negative region and the boundary region of X are shown as Fig. 1.

Classical rough sets are originally proposed to deal with categorical data. However, in real-world applications, there are many integer-valued and real-valued data. Thus, some extended models of the classical rough set were proposed to deal with this problem. Fuzzy rough set [13,15,17] and neighborhood rough set [38] are

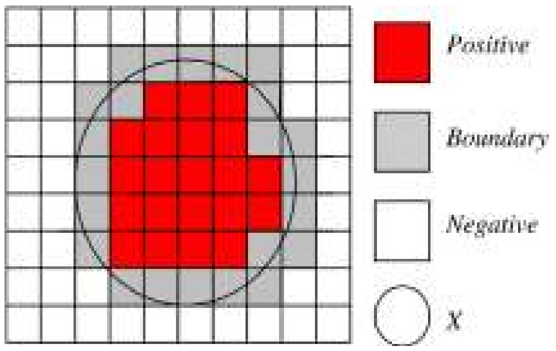

Fig. 1. Classical rough set.

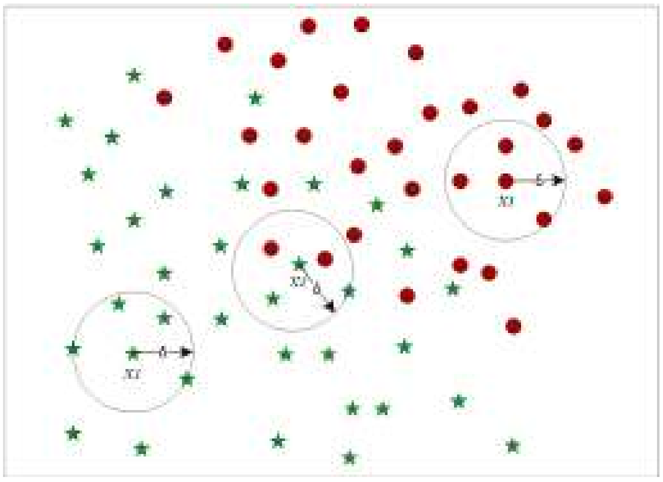

Fig. 2. δ neighborhood rough set.

two representative extensions of the classical rough set. Neighborhood Rough Set used neighborhood relation to replace the approximation based on equivalence relation of the traditional rough set model, which supports both continuous and discrete data sets [19,39]. In this section, we briefly review some basic concepts and notations of neighborhood rough set as follows.

An information system $S = (U, A)$, where $U = \{x_1, x_2, \ldots, x_n\}$ is a nonempty finite set of objects, called a universe. $A = \{a_1, a_2, \ldots, a_m\}$ is a nonempty finite set of attributes (features). More specifically, $S = (U, A, V, f)$ is called a decision table if $A = C \bigcup D$, where C is a set of condition attributes and D is a set of decision attributes, $C \bigcap D = \emptyset$. $V = \bigcup_{a \in A} V_a$, V_a is a domain of attribute a. f: $U \times A \to V$ is an information function such that $f(x, a) \in V_a$ for every $x \in U$, $a \in A$. $f(x_i, a_j)$ denotes the value of object x_i on the attributes a_j.

There are mainly two types of neighborhood relations: 1) neighborhood relation with a fixed distance (δ neighborhood), as shown in Fig. 2; 2) neighborhood relation with a fixed number of neighbors (k-nearest neighborhood), as shown in Fig. 3.

Definition 1. A metric Δ is a distance function from $R^N \times R^N \to R$, and $\Delta(x, y)$ denotes the distance between x and y. For $\forall x, y, z \in U$, it satisfies:

1) $\Delta(x, y) \geq 0$; $\Delta(x, y) = 0$ if and only if $x = y$;
2) $\Delta(x, y) = \Delta(y, x)$
3) $\Delta(x, z) \leq \Delta(x, y) + \Delta(y, z)$

Definition 2. Given U and C, let $B \subseteq C$ be a subset of attributes, $x \in U$. The neighborhood $\delta_B(x)$ of arbitrary object x on the feature subset B is defined as:

$$\delta_B(x) = \{y \mid \Delta(x, y) \leq \delta, y \in U\} \tag{3}$$

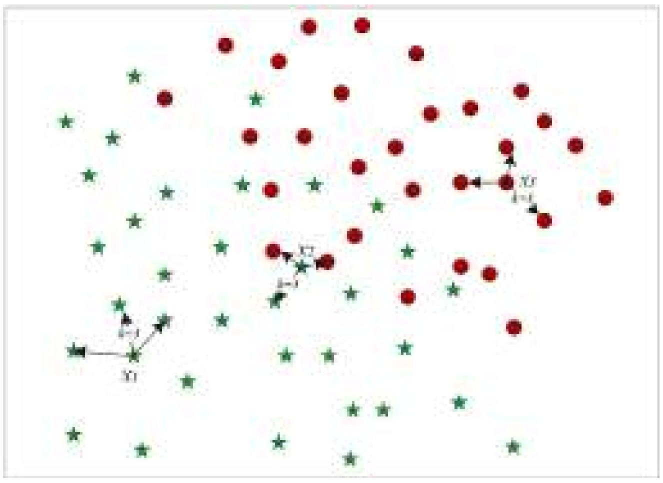

Fig. 3. k-nearest neighborhood rough set (k = 3).

Definition 3. Considering object x and given a set of numerical attributes B to describe the object, we call the k-nearest neighbors of x in terms of a k-nearest neighborhood information granule, denotes as $k_B(x)$.

$$k_B(x) = \{y \mid y \in Min_k\{Neighbors(x)\}, y \in U\} \tag{4}$$

Where $Neighbors(x)$ denotes all the neighbors of x, and $Min_k\{Neighbors(x)\}$ denotes the k nearest neighbors calculated on feature subset B.

Like classical rough set model, we give the lower and upper approximations of neighborhood relation R as follows.

Definition 4. Given a neighborhood approximation space $\Re_N = (U, R)$, for $\forall X \subseteq U$, two subsets of objects, called lower and upper approximations of X in terms of R neighborhood relation, are defined as

$$\underline{R}(X) = \{x_i \in U \mid R(x_i) \subseteq X, x_i \in U\} \tag{5}$$

$$\overline{R}(X) = \{x_i \in U \mid R(x_i) \cap X \neq \emptyset, x_i \in U\} \tag{6}$$

The boundary region of X in the approximation space is formulated as

$$BR(X) = \overline{R}(X) - \underline{R}(X) \tag{7}$$

As shown in Fig. 2, all the δ neighbor samples of x_1 have the same class label L_1 with mark "*" and the neighborhood samples of x_3 in a δ area are completely marked with "o" with another class label L_2. Meanwhile, the samples in the neighborhood of x_2 come from classes L_1 and L_2. We define the samples of x_2 as the boundary objects. Meanwhile, as shown in Fig 3, all the k-nearest neighbor (k = 3) samples of x_1 have the same class label L_1 and the neighborhood samples of x_3 have the same class label L_2. The neighbors of x_2 come from classes L_1 and L_2. In general, we need to find a feature subspace on which the boundary region is maintained as little as possible.

The size of the boundary region reflects the roughness degree of X in the approximation space. Usually, we hope that the boundary region of the decision is as little as possible for decreasing uncertain in the decision procedure. The lower approximation is also called positive region, denoted as $POS(x)$.

Definition 5. Let $B \subseteq C$, the dependency degree of B to D is defined as the ratio of consistent objects:

$$\gamma_B(D) = \frac{CARD(POS_B(D))}{CARD(U)}, \tag{8}$$

where $POS_B(D)$ denotes the lower approximation of D on feature subset B.

Thus, feature selection using neighborhood rough set aims to select a subset B from the feature set C that gets the maximal dependency degree of B to D.

4. Our new online streaming feature selection approach

In this section, we will introduce our new online streaming feature selection approach in detail. We first give a formal definition of online streaming feature selection. Then we introduce our new non-parameter neighborhood relation and the dependency calculation method. In terms of the new neighborhood relation and three evaluation criteria, we will present a new online streaming feature selection algorithm subsequently.

4.1. Definition of online streaming feature selection

Let $OSFS = (U, C \cup D, h, t)$ denote an online streaming feature selection framework, where U is a nonempty finite set of objects, C is the condition attribute set, and D is the decision attribute set. Let $C = [x_1, x_2, \ldots, x_n]^T \in R^{n\times d}$ consist of n samples over a d-dimensional feature space $F = [f_1, f_2, \ldots, f_d]^T \in R^d$. Let $D = [y_1, y_2, \ldots, y_n]^T \in R^{n\times 1}$ consist of n samples over the class label (decision feature space) $L = \{l_1, l_2, \ldots, l_m\}$, where l_i denotes the value of a class label. Given U, C and D, at each time stamp t, we get a new feature f_t of $C \cup D$ without knowing the exact number of d in advance. The problem of online streaming feature selection for mixed data is to derive a mapping h_t: $x_i \to L(x_i \in C)$ at each time stamp t, which is as good as possible using a subset of features that have arrived so far.

There are three challenges for online streaming feature selection. 1) Unlike traditional feature selection, we do not know the feature space before learning. Thus, we can not get any domain knowledge before selection. 2) Features are arriving randomly at each time. In order to decide whether detaining or discarding the new arriving features, we need to consider the new arriving feature and the selected feature subset as integration. 3) Although neighborhood rough set-based data mining does not require any domain knowledge, it is still a challenge to specify unified parameters δ for the δ neighborhood and k for the k-nearest neighborhood before learning. In the next, we will introduce a new neighborhood relation which need not specify any parameters before learning.

4.2. Our new neighborhood relation

Definition 6. Let $N_B(x_i)$ denote all of the neighbors of x_i sorted by the distance from the nearest to the farthest on feature subset B,

$$N_B(x_i) = < x_{(i,1)}, x_{(i,2)}, \ldots, x_{(i,j)}, \ldots, x_{(i,n-1)} > \tag{9}$$

where $\{x_i, x_{(i,1)}, x_{(i,2)}, \ldots, x_{(i,n-1)}\} = U$ and $\Delta(x_i, x_{(i,1)}) \leq \Delta(x_i, x_{(i,2)}) \leq \ldots \leq \Delta(x_i, x_{(i,n-1)})$.

We define the density of x_i to neighbor $x_{(i,\,k)}$ as $Density(x_i, x_{(i,k)}) = \frac{\Delta(x_i, x_{(i,k)})}{k}$, denoted as $d(k)$ for short. From $x_{(i,\,1)}$ to $x_{(i,n-1)}$, assuming the density value first decreases at neighbor $x_{(i,\,k)}$, then, we call $x_{(i,\,k)}$ the first **Inflection Point**, denoted as $IP(x_{(i,\,k)})$. We use the samples between x_i and the first Inflection Point as the nearest neighbors of x_i, shown as Fig. 4.

Based on this, we proposed a new neighborhood relation with adaptive neighbors using the Inflection Point, denoted as $IP_C(x)$ as shown in Eq. (10).

Definition 7. Given a set of finite and nonempty objects $U = \{x_1, x_2, \ldots, x_n\}$, the condition feature set C and a feature subset B

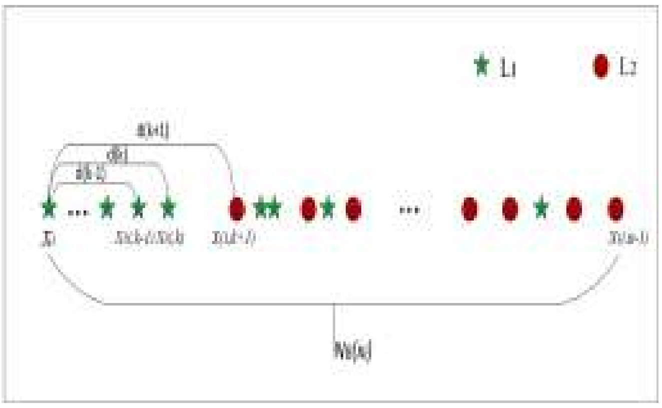

Fig. 4. our new neighborhood relation.

Table 1
An example dataset.

$x \in U$	f_1	f_2	f_3	f_4	d
x_1	3	5.6	1	1	-1
x_2	5	6.9	1	2	1
x_3	8	5.3	1	1	1
x_4	13	12.3	0	1	1
x_5	6	15.2	0	3	-1
x_6	5	2.6	0	2	1
x_7	9	6.8	1	2	-1
x_8	15	8.4	0	2	-1

($B \subseteq C$). For target object x_i, let $N_B(x_i) =< x_{(i,1)}, x_{(i,2)}, \ldots, x_{(i,n-1)} >$ denote all the neighbors of x_i from the nearest to the farthest on B. The adaptive neighborhood of arbitrary object $x_i \subseteq U$ on B is defined as:

$$IP_B(x_i) = \{x_{(i,1)}, x_{(i,2)}, \ldots, x_{(i,k-1)}\}, \tag{10}$$

where $IP(x_{(i,\,k)})$ is the first **Inflection Point** from $x_{(i,\,1)}$ to $x_{(i,n-1)}$.

Table 1shows an example dataset used to illustrate the definition of our new neighborhood relation, where x_1 to x_8 are the samples with four condition features (f_1 to f_4) and one decision feature (d). The distance function is calculated using Euclidean distance.

Let's take object x_3 and feature set $B = \{f_1, f_2\}$ as an example. First, we calculate all distances between x_3 and x_i ($i \neq 3$) on B namely: $\Delta_B(x_3, x_1) = \sqrt{(8-3)^2 + (5.3-5.6)^2} = 5.009$, $\Delta_B(x_3, x_2) = 3.4$, $\Delta_B(x_3, x_4) = 8.602$, $\Delta_B(x_3, x_5) = 10.1$, $\Delta_B(x_3, x_6) = 4.036$, $\Delta_B(x_3, x_7) = 1.803$, $\Delta_B(x_3, x_8) = 7.655$. All the neighbors of x_3 from the nearest to the farthest are $N_B(x_3) = \{x_7, x_2, x_6, x_1, x_8, x_4, x_5\}$.

For Density neighborhood, $Density(x_3, x_7) = \frac{\Delta(x_3, x_7)}{1} = 1.803$, $Density(x_3, x_2) = 1.7$. Thus, x_2 is the first Inflection Point and the Density neighborhood of x_3 is $IP_B(x_3) = \{x_7\}$.

Based on this new density neighborhood relation, we proposed the new dependency calculation method as follows.

In Algorithm 1, we calculate the CARD value of each instance x_i and get the sum for the final dependency degree. The CARD value ranges from 0 to 1, denoted as the consistency of x_i's class attribute with its neighbors' class attributes. In order to find the neighbors of x_i, we need to sort all neighbors of x_i by the distance. The time complexity of quicksort function is $O(n^*logn)$. Thus, the time complexity of **Dependency-Mixed** is $O(|X_S|^{2*}log|X_S|)$.

4.3. Our new algorithm

For online streaming feature selection, features flow in one by one over time. At time stamp t, we have the new arriving feature f_t and the selected candidate subset S_{t-1}. The aim of our new algorithm is to select features from $S_{t-1} \cup f_t$ with high correlation, high dependency, and low redundancy.

4.3.1. High correlation

For high correlation, it means the features selected in S_t at time stamp t should be maximal correlated to the decision attributes. For each feature f_i, we can calculate the dependency $\gamma_{f_i}(D)$ with Eq. (8). Thus, in order to get the high correlation, we should maximize the mean value of all dependency values between individual feature f_i and target class label D:

$$\mathbf{Max}\{\mathbb{R}(S, D)\}, \mathbb{R} = \frac{1}{|S_t|} \sum_{f_i \in S_t} \gamma_{f_i}(D). \tag{11}$$

For the new arriving features f_t at time stamp t, we calculate $\gamma_{f_t}(D)$ and compare it with $\mathbb{R}(S_{t-1}, D)$. If $\gamma_{f_t}(D) < \mathbb{R}(S_{t-1}, D)$, f_t will be discarded.

Theorem 1. *Suppose at time stamp $t-1$, the selected feature set is S_{t-1}. At time stamp t, the new arriving feature is f_t. If $\gamma_{f_t}(D) < \mathbb{R}(S_{t-1}, D)$ and we add f_t into S_{t-1}, then $\mathbb{R}(S_t, D) < \mathbb{R}(S_{t-1}, D)$.*

Proof. Let $|S_{t-1}| = N_{t-1}$ and $\mathbb{R}(S_{t-1}, D) = r_{t-1}$. It is obvious that $\sum_{f_i \in S_{t-1}} \gamma_{f_i}(D) = r_{t-1} \times N_{t-1}$. For $\gamma_{f_t}(D) < \mathbb{R}(S_{t-1}, D)$. If we add f_t into S, then $S_t = S_{t-1} \cup f_j$ and $|S_t| = N_{t-1} + 1$.

$$\begin{aligned}
&\mathbb{R}(S_t, D) = \tfrac{1}{|S_t|} \textstyle\sum_{f_i \in S_t} \gamma_{f_i}(D) \\
&= \tfrac{1}{N_{t-1}+1} (N_{t-1} \times r_{t-1} + \gamma_{f_j}(D)) \\
&= \tfrac{N_{t-1}}{N_{t-1}+1} r_{t-1} + \tfrac{1}{N_{t-1}+1} \gamma_{f_j}(D) \\
&= r_{t-1} + \tfrac{1}{N_{t-1}+1} (\gamma_{f_j}(D) - r_{t-1}). \\
&\because \gamma_{f_t}(D) < \mathbb{R}(S_{t-1}, D), \therefore \gamma_{f_j}(D) - r_{t-1} < 0, \therefore \mathbb{R}(S_t, D) < \mathbb{R}(S_{t-1}, D).
\end{aligned}$$

□

4.3.2. High dependency

For neighborhood rough set based feature selection, the final goal is to get a subset from feature space which can achieve the maximal dependency according to Eq. (8). In other words, at each time stamp t, for the selected candidate subset S_t, we should make sure that

$$\mathbf{Max}\{\mathbb{D}(S_t, D)\}, \mathbb{D} = \gamma_{S_t}(D). \tag{12}$$

For the new arriving features f_t at time stamp t, if $\gamma_{S_{t-1} \cup f_t}(D) \geq \gamma_{S_t}(D)$, we should add f_t into S_{t-1}. Otherwise, we will discard f_t.

Theorem 2. *[40] Suppose B is a subset of conditional features, f is an arbitrary conditional attribute that belongs to the dataset, and D is the set of decision attributes. Then $\gamma(B \cup f, D) \geq \gamma(B, D)$.*

Proof. The proof of this theorem is available in [40] page 90. □

With Theorem 2, we can find that, if we only use high dependency and high correlation for feature selection, there will be a lot of redundant features in the candidate subset ($\gamma_{S_{t-1} \cup f_t}(D) == \gamma_{S_t}(D)$). Thus, we need to consider the redundancy of the selected subset.

4.3.3. Low redundancy

In order to measure each feature's importance in the selected candidate subset, we need to define the significance of single feature to its feature set. The significance of a feature f to feature set B ($f \in B$) is defined as follows:

Definition 8. Given a condition attribute set B ($B \subseteq C$) and a decision attribute set D, a feature $f \in B$, the significance of the feature f to B is defined as:

$$\sigma_B(f, B) = \gamma_B(D) - \gamma_{\{B-f\}}(D) \tag{13}$$

In order to select the features with low redundancy, we should make the mean significance of each feature f_i in S achieve the maximal. That is

$$\mathbf{Max}\{\mathbb{S}(S_t, D)\}, \mathbb{S} = \frac{1}{|S_t|}\sum_{f_i \in S_t}\{\sigma_{S_t}(f_i, D)\}. \tag{14}$$

With the high dependency constraint and Theorem 2, we can see that $\forall f_i \in S_t$, $\sigma_{S_t}(f_i, D) \geq 0$ is satisfied. Thus, we should discard the features in S which satisfy the constraint $\sigma_{S_t}(f_i, D) = 0$.

However, in real data sets, we find it rare that the dependency of $S \cup f_i$ is exactly equal to the dependency of S. Thus, we relax the exactly equal restriction and change it to an interval restriction. That is, if

$$\left|\frac{Dep_S - \gamma_{S\cup f_i}}{Dep_S}\right| \leq \lambda, \tag{15}$$

then we will execute the redundancy analysis. With this new fuzzy equal constraint, more candidate features will be considered into the redundancy analysis step and this will make the final selected feature subset lower redundancy. The default value of λ is 0.05, more details refer to Section 5.2.

To sum up, we propose our new online streaming feature selection algorithm as Algorithm 2.

Algorithm 1 Dependency-Density .

```
Require: ~~
    $X_S$: sample values on feature set $S$;
    $R$: density neighborhood relation;
Ensure: ~~
    $dep_S$: dependency on feature set $S$;
1: $card_S$: the number of positive samples on $S$, initialized to 0;
2: $card_U$: the number of instances of $X_S$;
3: FOR each $x_i$ in $X_S$
4:     find the neighbor samples of $x_i$ on $R$ as $S_R(x_i)$;
5:     calculate the card value of $x_i$ as $Card(S_R(x_i))$;
6:     $card_S = card_S + Card(S_R(x_i))$;
7: END FOR
8: $dep_S = card_S/card_U$;
9: return $dep_S$;
```

More specifically, if a new feature f_i arrives at time stamp t_i, Step 7 calculates the dependency of f_i using the dependency calculation method **Dependency-Density**. Step 8 compares the dependency of f_i with the mean dependency of the selected feature set S. If γ_{f_i} is smaller than $Mean_{Dep_S}$, f_i is discarded. Step 11 compares the dependency of current feature set S with the dependency of the feature set $S \cup f_i$. If the dependency of $S \cup f_i$ is bigger than Dep_S, which means adding new feature f_i will increase the dependency of the selected feature set, then we add f_i into S. Otherwise, if the ratio of the difference between the dependency of $S \cup f_i$ and Dep_S with Dep_S is less than a fixed value λ, we will analyse the feature redundancy. For each feature in $S \cup f_i$, we randomly select a feature from the candidate feature set and calculate its significance according to Eq. (14). We will discard features whose significance equal to 0. In sum, with this new online streaming feature selection algorithm, we can select features with high correlation, high dependency, and low redundancy.

4.4. Time complexity of OFS-Density

The time complexity of OFS-Density mainly depends on the dependency function **Dependency-Density**.

Suppose the data set is $\mathbb{D}$, the number of instances in $\mathbb{D}$ is N and the number of features in $\mathbb{D}$ is F. According to Section 4.2, the time complexity of **Dependency-Density** is $O(N^2 logN)$. At time

Algorithm 2 OFS-Density.

```
Require: ~~
    $X$: the data samples with condition features;
    $Y$: the decision classes;
Ensure: ~~
    $S$: the selected feature set;
1: $S$: the selected feature set, initialized to {};
2: $\lambda$: the parameter control the fuzzy equal constraint(default value 0.05);
3: $Dep_S$:the dependency of $S$ to $Y$, initialized to 0;
4: $Mean_{Dep_S}$: the mean dependency of features in $S$, initialized to 0;
5: Repeat
6: Get a new feature $f_i$ of $X$ at time stamp $t_i$ as $X_{f_i}$;
7: Calculate the dependency of $X_{f_i}$ as $\gamma_{f_i}$ using Dependency-Density;
8: IF $\gamma_{f_i} < Mean_{Dep_S}$
9:     Discard feature $f_i$ and go to Step 24;
10: END IF
11: IF $\gamma_{S\cup f_i} > Dep_S$
12:     $S = S \cup f_i$;
13:     $Dep_S = \gamma_S, Mean_{Dep_S} = \frac{1}{|S|}\sum_{f_i \in S}\gamma_{f_i}(Y)$;
14: ELSE IF $|(Dep_S - \gamma_{S\cup f_i})/Dep_S| <= \lambda$
15:     $S = S \cup f_i$;
16:     FOR each feature in $S$
17:         Randomly select a feature $f'$ in $S$;
18:         Calculate $f'$ 's significance as $\sigma_S(f')$;
19:         IF $\sigma_S(f') == 0$
20:             Remove feature $f'$ from $S$;
21:         END IF
22:     END FOR
23: END IF
24: Until no more features are available;
25: return $S$;
```

stamp t_i, a new feature f_i is presented to the algorithm. Steps 6–8 calculate the dependency of f_i and compare it with $Mean_{Dep_S}$ (the mean dependency value of each feature in selected feature set S). The time complexity is $O(N^2 logN)$. If the dependency of f_i is smaller than $Mean_{Dep_S}$, f_i will be discarded. Otherwise, we calculate the dependency of $S \cup f_i$ and compare it with Dep_S (the dependency of currently selected feature set). This time complexity is also $O(N^2 logN)$. If the dependency of $S \cup f_i$ is bigger than Dep_S, we add f_i into S and go on to the next feature. If the dependency of $S \cup f_i$ is equal to or little smaller than Dep_S, we will calculate each features' significance and remove the redundant features from S. The time complexity of this phase is $O(|S|^* N^2 logN)$.

Thus, the worst time complexity of OFS-Density is $O(F^2 {*} N^2 logN)$.

5. Experimental

5.1. Experiment setup

In this section, we apply the proposed online feature selection algorithm on fourteen data sets, including four UCI data sets (WDBC, HILL VALLEY, IONOSPHERE,SONAR), nine DNA microarray data sets (PROSTATE-std, COLON, LYMPHOMA-std, DLBCL, GLIOMA, SRBCT-std, LUNG2, LEUKEMIA-std, MLL) [41,42] and one NIPS 2003 data set (ARCENE) [6] as shown in Table 2.

In our experiments, we use three basic classifiers, KNN, SVM, and CART in Matlab R2015b to evaluate a selected feature subset. We perform 10-fold cross-validation on each data set. Feature selection is training on 9/10 data samples and testing on the rest

Table 2
Experimental data sets.

Data set	Instances	Features	Classes
IONOSPHERE	351	34	2
WDBC	569	30	2
SONAR	208	60	2
HILL	606	100	2
COLON	62	2000	2
SRBCT	63	2308	4
LUNG2	203	3312	5
LYMPHOMA	62	4026	3
GLIOMA	50	4433	4
MLL	72	5848	3
PROSTATE	102	6033	2
DLBCL	77	6285	2
LEU	72	7129	2
ARCENE	200	10000	2

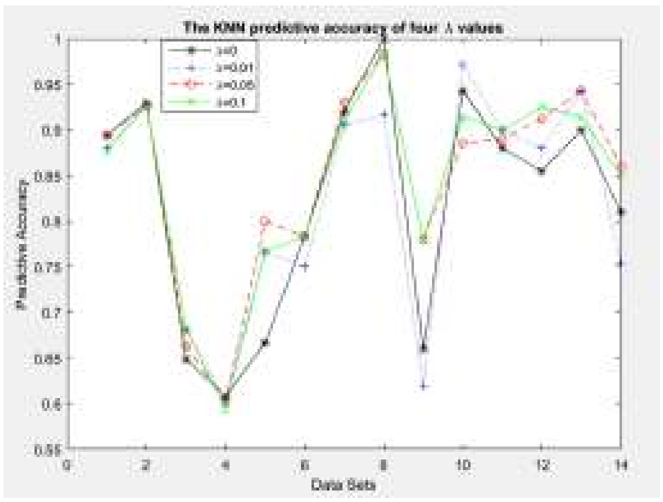

Fig. 5. Predictive accuracy in KNN varying with four different values of λ .

1/10 data. All competing algorithms use the same training and testing data for each fold. All experimental results are conducted on a PC with Intel(R) i5-3470S, 2.9 GHz CPU, and 8GB memory.

To validate whether OFS-Density and its rivals have significant differences in the predictive accuracy, we conduct the Friedman test at a 95% significance level [43], under the null-hypothesis. The performance of OFS-Density and its rivals has no significant difference if the null-hypothesis is accepted. When the null-hypothesis at the Friedman test is rejected, we continuously proceed with the Nemenyi test [43] as a post-hoc test. With the Nemenyi test, the performance of those two methods is significantly different if the corresponding average rankings differ by at least the critical difference (how to calculate the critical difference, please see [43]).

5.2. Analysis of λ in OFS-Density

In this subsection, we will analyse the influence of λ in OFS-Density. We select three values (0.01, 0.05 and 0.1) of λ and the exactly equal constraint ($\lambda = 0$) as compared ones.

Figs. 5–7 show the experimental results of our new algorithm with four different λ values (0, 0.01, 0.05 and 0.1) on these data sets (the data sets from 1 to 14 are IONOSPHERE, WDBC, SONAR, HILL, COLON, SRBCT, LUNG2, LYMPHOMA, GLIOMA, MLL, PROSTATE, DLBCL, LEU, ARCENE). Figs. 8 and 9 show the mean number of selected features and running time on these data sets. In these experiments, we select KNN, SVM, and CART as the basic classifiers, and the value of k in KNN is set to 1.

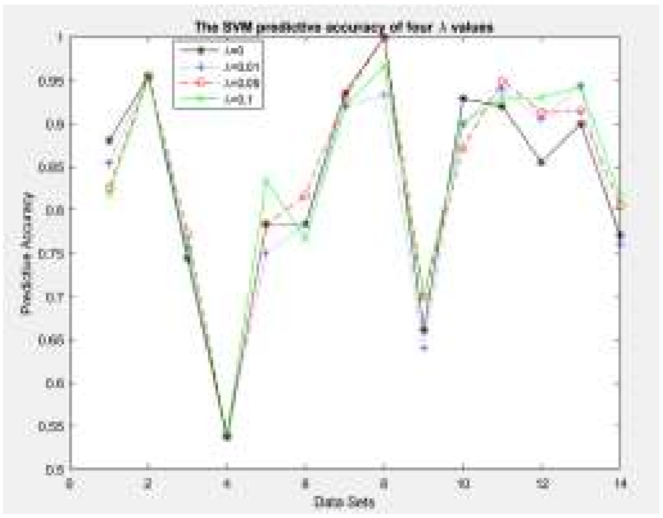

Fig. 6. Predictive accuracy in SVM varying with four different values of λ .

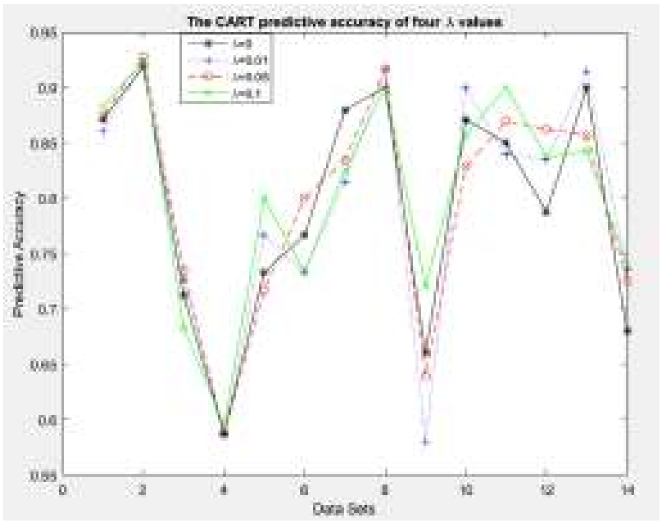

Fig. 7. Predictive accuracy in CART varying with four different values of λ .

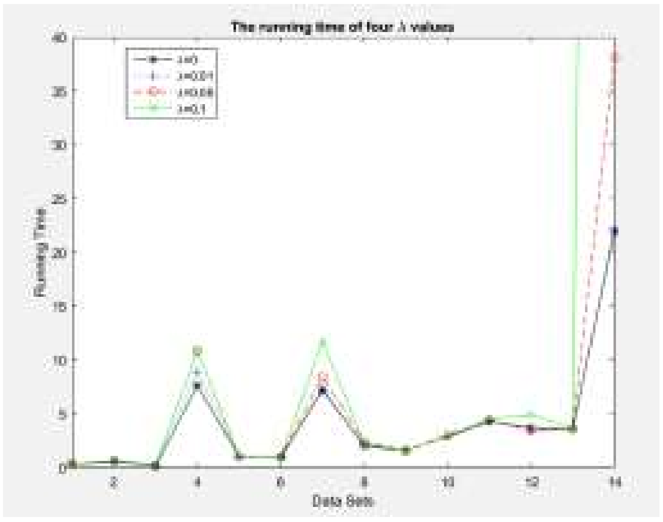

Fig. 8. Running Time varying with four different values of λ .

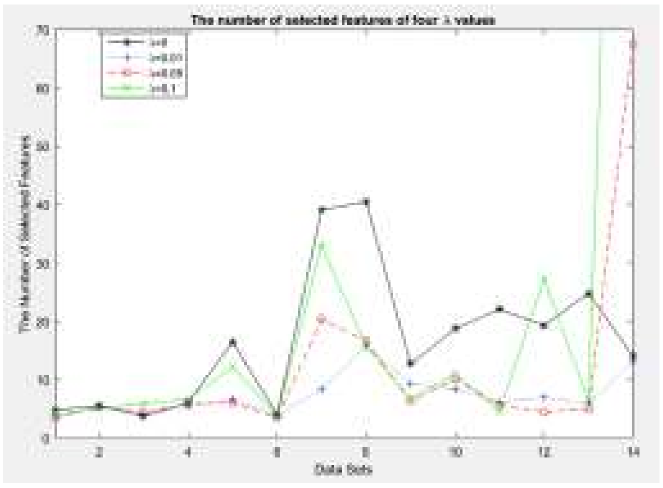

Fig. 9. The mean number of selected features on four different values of λ .

Table 3
The mean values of different λ on predictive accuracy, running time and number of selected features.

	$\lambda = 0$	$\lambda = 0.01$	$\lambda = 0.05$	$\lambda = 0.1$
KNN classifer	0.8213	0.8215	**0.8467**	0.8433
SVM classifer	0.8321	0.8272	**0.8411**	0.8405
CART classifer	0.7942	0.7950	0.7982	**0.8029**
Running time	**4.1152**	4.1929	5.6289	25.3083
Selected features	16.5785	**7.3571**	11.7785	26.1428

Table 4
P-values of $\lambda = 0$ VS. $\lambda = 0.01, 0.05, 0.1$.

	$\lambda = 0$	$\lambda = 0.01$	$\lambda = 0.05$	$\lambda = 0.1$
KNN classifer	–	0.5930	0.2482	0.7815
SVM classifer	–	0.7815	0.0833	0.5930
CART classifer	–	0.5930	0.4054	0.4054
Running time	–	0.5930	**0.0075**	**0.0075**
Selected features	–	**0.0002**	**0.0075**	0.1088

In Fig. 8, the running time of $\lambda = 0.1$ is 308.956. In Fig. 9, the number of selected features of $\lambda = 0.1$ is 223.9. Table 3 shows the mean value of predictive accuracy, running time and number of selected features with different values of λ.

Besides, with the Friedman test, the *p*-values of $\lambda = 0$ (exactly equal) vs. $\lambda = 0.01, 0.05, 0.1$ on predictive accuracy, running time and number of selected features can be seen in Table 4.

From Figs. 5–9 and Tables 3 and 4, we have the following observations.

- There is no significant difference in predictive accuracy with different values of λ. $\lambda = 0.05$ gets the best performance with KNN and SVM classifiers and $\lambda = 0.1$ gets the highest mean predictive accuracy with CART classifier.
- With the increasing of values of λ, the corresponding running time increases rapidly. This is because a bigger λ value means more times to run the feature redundancy analysis.
- On the number of selected features, $\lambda = 0$ selects more features than $\lambda = 0.01$ and $\lambda = 0.05$. This indicates that the exactly equal constraint can lead to some redundant features. However, $\lambda = 0.1$ selects the maximum number of features and consumes the maximum running time. Thus, bigger values of λ do not mean a better performance.

In sum, relaxing the exactly equal restriction can remove redundant features and get a promotion on the predictive accuracy. However, the λ value is not the bigger the better. In the next experiments, we will use $\lambda = 0.05$ for the OFS-Density algorithm.

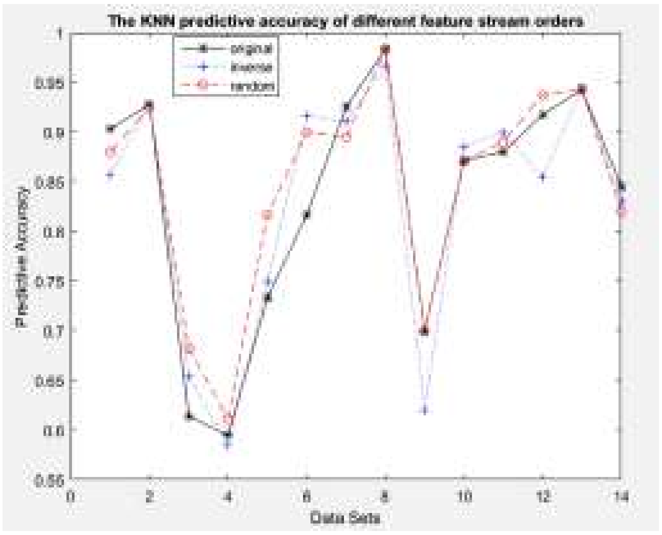

Fig. 10. Predictive accuracy using KNN on different feature stream orders.

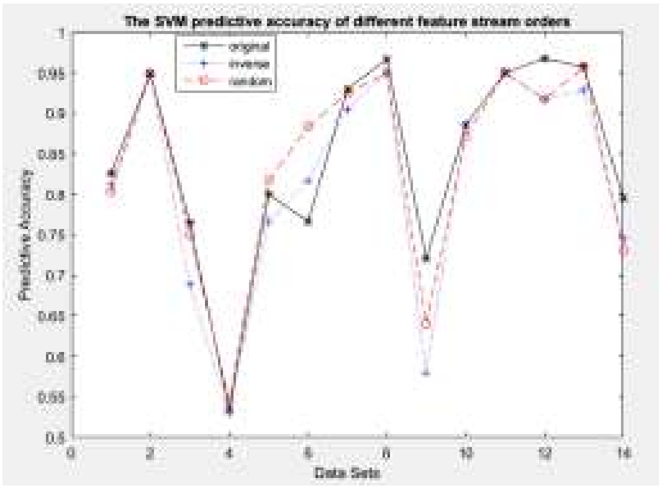

Fig. 11. predictive accuracy using SVM on different feature stream orders.

5.3. Influence of feature stream order

In this subsection, we will validate the influence of feature stream order on our new neighborhood relation and new online streaming feature selection algorithm. We compare three types of feature stream orders: original, inverse and random.

Figs. 10–12 show the experimental results of our new algorithm with three different feature stream orders on these data sets. Figs. 13 and 14 show the mean number of selected features and running time on these data sets. In these experiments, we select KNN, SVM, and CART as the basic classifiers and the value of *k* in KNN is set to 1.

The p-values of original vs. inverse and random on predictive accuracy, running time and number of selected features can be seen in Table 5.

From Figs. 10–14, we can see that there are minor fluctuations on predictive accuracy, running time and number of selected features varying with different feature stream orders. From Table 5, we can see that there is no significant difference among these three orders on predictive accuracy, running time and number of selected features, except in the cases of original vs. inverse with SVM classifier. The main reason is that SVM classifier is robust and the predictive accuracy basically increases with the number of se-

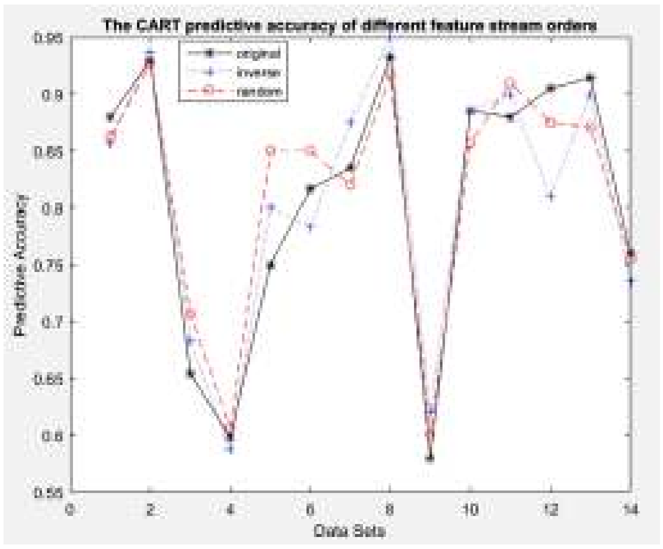

Fig. 12. Predictive accuracy using CART on different feature stream orders.

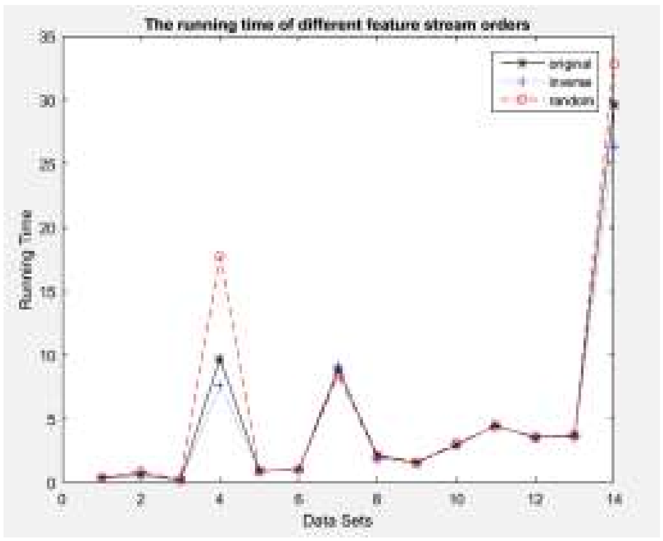

Fig. 13. Running time varying with different feature stream orders.

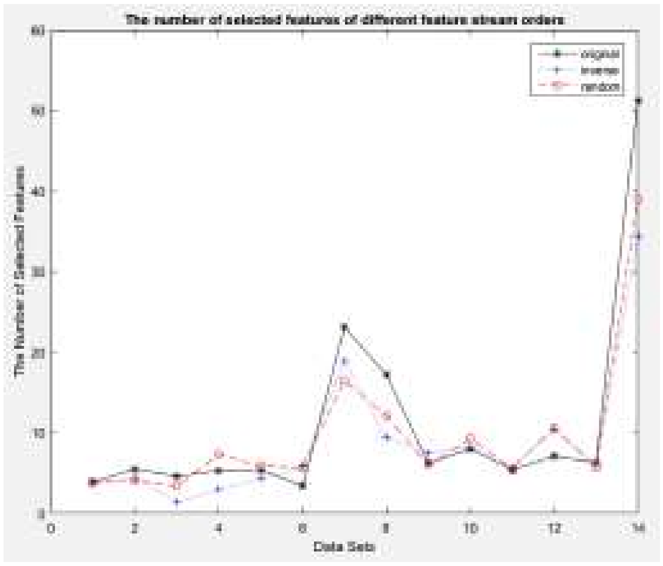

Fig. 14. The mean number of selected features on different feature stream orders.

Table 5
The *p*-values of original VS. inverse and random.

	Original	Inverse	Random
KNN classifer	–	0.5637	0.5271
SVM classifer	–	**0.0039**	0.1317
CART classifer	–	0.7815	0.5930
Running time	–	1.0000	0.2850
Selected features	–	0.2850	0.5930

lected features. Meanwhile, our new method selects few features and this makes the predictive performance with SVM is unstable. In sum, the feature stream order has little influence on our new online streaming feature selection method.

5.4. OFS-Density vs. traditional feature selection methods

In this subsection, we compare OFS-Density with eight representative traditional feature selection methods, including Fisher Score [27], ReliefF [28], PCC (Pearson Correlation Coefficient) [44], Laplacian Score [31], MI (mutual information) [29], INF [32], LLC-FS [33] and FSV [30].

All these algorithms are implemented in MATLAB [45]. The K value of ReliefF is set to 5 for the best performance. None of these eight traditional feature selection methods can handle the scenario of feature streaming in an online manner. Thus, we rank all the features evaluated by these traditional feature selection methods from high to low and select the same number of features as OFS-Density. We evaluate OFS-Density and all competing ones on the predictive accuracy with 10-fold cross-validation.

Tables 6–8 summarize the predictive accuracy of OFS-Density against the other eight competing algorithms using the basic classifiers of KNN ($k = 1$), SVM and CART. The p-values of Friedman test on KNN, SVM and CART are 2.9390e−10, 1.2618e−07, and 2.7607e−07. Thus, there is a significant difference between OFS-Density and other eight competing algorithms respectively on predictive accuracy. According to the Nemenyi test, the value of CD (critical difference) is 3.2132.

From Tables 6–8, we have the following observations.

- OFS-Density vs. Fisher. According to the values of average rankings and CD, there is no significant difference between OFS-Density and Fisher on predictive accuracy with these three classifiers. OFS-Density outperforms Fisher on ten of fourteen datasets in cases with KNN, SVM, and CART. This is because Fisher measures the features independently, and it can not consider the information of the selected feature set as an integral. In total, OFS-Density performs better than Fisher.
- OFS-Density vs. PCC. There is a significant difference between OFS-Density and PCC in predictive accuracy with KNN, and there is no significant difference between them with CART and SVM. OFS-A3M outperforms PCC on eleven of the fourteen datasets. For some data sets, such as SRBCT, DLBCL, and ARCENE, OFS-Density is higher PCC over 20% on predictive accuracy. PCC can not handle some datasets well. OFS-Density is superior to PPC.
- OFS-Density vs. ReliefF. There is no significant difference in predictive accuracy between OFS-A3M and ReliefF with KNN, SVM, and CART. OFS-Density gets the higher predictive accuracy than ReliefF on twelve of the fourteen datasets. ReliefF also uses the neighbors' information for feature selection. However, ReliefF does not discriminate redundant features which makes it performance bad on some data sets, such as GLIOMA.
- OFS-Density vs. MI. There is a significant difference between OFS-Density and MI with these three classifiers. OFS-A3M outperforms MI on thirteen of the fourteen datasets at least. The

Table 6
Predictive accuracy using the KNN classifier.

Data set	OFS-Density	Fisher	PCC	ReliefF	MI	Laplacian	INF	LLC-FS	FSV
IONOSPHERE	**0.8943**	0.88	0.8743	0.8229	0.7943	0.8343	0.8171	0.8543	0.7943
WDBC	**0.9385**	0.9332	0.9332	0.8962	0.9016	0.9244	0.9367	0.9086	0.9016
SONAR	0.6827	0.7108	0.7108	0.6779	0.5479	0.5955	0.6955	0.6684	0.5479
HILL	**0.595**	0.5314	0.5314	0.557	0.5545	0.4967	0.5033	0.5479	0.5545
COLON	**0.75**	0.7333	0.7333	0.7167	0.5167	0.5167	0.5667	0.6167	0.5167
SRBCT	**0.8833**	**0.8833**	0.5833	0.8667	0.6333	0.3167	0.2667	0.3833	0.6333
LUNG2	**0.93**	0.79	0.815	0.8	0.745	0.765	0.81	0.79	0.835
LYMPHOMA	**1**	0.9833	0.95	0.9667	0.7833	0.9333	0.6	0.95	0.7833
GLIOMA	**0.68**	0.64	0.66	0.26	0.56	0.42	0.4	0.48	0.62
MLL	**0.9286**	0.9	0.7857	0.9286	0.6429	0.8429	0.9143	0.8714	0.8286
PROSTATE	**0.93**	0.91	0.91	0.91	0.63	0.59	0.48	0.72	0.63
DLBCL	**0.95**	0.7925	0.4425	0.8125	0.605	0.685	0.7375	0.7625	0.7875
LEUKEMIA	**0.9571**	0.8714	0.8714	0.8429	0.6	0.5857	0.4714	0.4714	0.6
ARCENE	**0.86**	0.655	0.615	0.675	0.585	0.7	0.715	0.655	0.69
AVG. ACCURACY	**0.8556**	0.8010	0.7439	0.7666	0.6499	0.6575	0.6367	0.6913	0.6944
AVG. RANKS	**8.7143**	6.5714	5.5000	5.6071	2.8571	3.2857	3.9643	4.2857	4.2143

Table 7
Predictive accuracy using the SVM calssifier.

Data set	OFS-Density	Fisher	PCC	ReliefF	MI	Laplacian	INF	LLC-FS	FSV
IONOSPHERE	0.8143	**0.8571**	0.8343	0.6914	0.84	0.6971	0.6686	0.7771	0.84
WDBC	**0.9614**	0.9526	0.9526	0.9209	0.9244	0.9297	0.9526	0.9437	0.9244
SONAR	**0.7504**	0.7361	0.7361	0.7008	0.5674	0.6165	0.6456	0.6476	0.5674
HILL	**0.5339**	0.5074	0.5074	0.5207	0.5074	0.5099	0.5058	0.5149	0.5074
COLON	0.8167	0.85	0.85	**0.8667**	0.65	0.7	0.6833	0.7167	0.65
SRBCT	0.8	**0.8833**	0.6667	0.75	0.6833	0.3833	0.3667	0.3	0.6833
LUNG2	**0.935**	0.845	0.84	0.85	0.85	0.82	0.865	0.86	0.85
LYMPHOMA	**0.9833**	0.9333	0.9	0.9167	0.6833	0.9	0.65	0.9167	0.6833
GLIOMA	**0.6**	0.58	**0.6**	0.28	**0.6**	0.46	0.48	0.42	0.48
MLL	0.9286	**0.9429**	0.9	**0.9429**	0.7	0.9143	**0.9429**	0.8857	0.9
PROSTATE	**0.94**	0.92	0.92	**0.94**	0.59	0.6	0.47	0.73	0.59
DLBCL	**0.975**	0.8625	0.7875	0.825	0.6925	0.71	0.685	0.7625	0.8
LEUKEMIA	**0.9429**	0.9	0.9	0.8857	0.6143	0.6429	0.5857	0.6	0.6143
ARCENE	**0.805**	0.725	0.625	0.65	0.575	0.635	0.67	0.66	0.725
AVG. ACCURACY	**0.8418**	0.8210	0.7871	0.7672	0.6769	0.6799	0.6550	0.6953	0.7010
AVG. RANKS	**8.1071**	7.0000	5.3214	5.6429	3.3929	3.6786	3.4643	4.3214	4.0714

Table 8
Predictive accuracy using the CART calssifier.

Data set	OFS-Density	Fisher	PCC	ReliefF	MI	Laplacian	INF	LLC-FS	FSV
IONOSPHERE	**0.9114**	0.8829	0.86	0.7943	0.8371	0.8314	0.8029	0.8114	0.8371
WDBC	**0.9262**	0.9227	0.9227	0.891	0.9158	0.9209	0.9245	0.9121	0.9158
SONAR	0.6965	**0.7261**	**0.7261**	0.5807	0.514	0.5331	0.6436	0.5875	0.514
HILL	**0.5926**	0.5107	0.5107	0.5248	0.5091	0.4983	0.4835	0.5124	0.5091
COLON	**0.7833**	**0.7833**	**0.7833**	0.7667	0.6333	0.5333	0.6167	0.6667	0.6333
SRBCT	0.75	**0.9167**	0.65	0.8167	0.7	0.3	0.25	0.3167	0.7
LUNG2	**0.83**	0.795	0.785	0.82	0.715	0.82	0.8	0.82	0.8
LYMPHOMA	**0.9667**	0.85	0.8833	0.85	0.65	0.8833	0.7	0.9	0.65
GLIOMA	0.52	**0.62**	**0.62**	0.34	0.4	0.48	0.38	0.5	0.48
MLL	0.8571	0.8143	0.8571	0.8571	0.6	0.8286	0.8571	0.7714	**0.8714**
PROSTATE	0.89	0.91	0.91	**0.93**	0.65	0.59	0.48	0.65	0.65
DLBCL	**0.9125**	0.825	0.8125	0.7875	0.68	0.7225	0.6475	0.725	0.8
LEUKEMIA	**0.8714**	0.9286	0.9286	0.8429	0.5429	0.6714	0.6	0.5	0.5429
ARCENE	**0.805**	0.65	0.585	0.645	0.595	0.68	0.66	0.66	0.715
AVG. ACCURACY	**0.8080**	0.7953	0.7738	0.7461	0.6387	0.6637	0.6318	0.6666	0.6870
AVG. RANKS	**7.9643**	6.6071	6.2143	5.0000	2.8571	4.0000	3.3929	4.4643	4.5000

features are evaluated independently with MI which makes it performance inferior to OFS-Density.

- OFS-Density vs. Laplacian Score. There is a significant difference between OFS-Density and Laplacian in predictive accuracy with these three classifiers. OFS-Density outperforms Laplacian Score on all of these datasets. As an unsupervised method, Laplacian Score does not use the class information for feature selection. In general, Laplacian Score performance inferior to OFS-Density.
- OFS-Density vs. INF. INF gets the lowest mean predictive accuracy and there is a significant difference between OFS-Density and INF with these three classifiers. OFS-Density outperforms INF on thirteen of the fourteen datasets. On some data sets, such as SRBCT, LYMPHOMA, GLIOMA, PROSTATE, and LEUKEMIA, INF performs badly. INF is an unsupervised method and the performance is inferior to OFS-Density.
- OFS-Density vs. LLC-FS. There is a significant difference between OFS-Density and LLC-FS with these three classifiers. Meanwhile, OFS-Density performs better than LLC-FS on all these data sets.
- OFS-Density vs. FSV. There is a significant difference between OFS-Density and FSV in predictive accuracy. Although FSV is a wrapper method, OFS-Density performs better than FSV on thir-

Table 9
Predictive accuracy using KNN as the base classifier.

Data set	OFS-Density	Grafting	α-investing	OSFS	Fast-OSFS	SAOLA	OS-SA	OFS-A3M
IONOSPHERE	**0.9057**	0.8714	0.9	0.8686	0.8743	0.8714	0	0.8571
WDBC	0.9403	0.9403	0.9561	0.9614	**0.9632**	0.9104	0.186	0.9561
SONAR	0.6759	0.8599	0.7817	0.7256	0.7093	0.6293	0.7632	**0.8308**
HILL	0.5686	0.5397	0.638	0	0	0	0	**0.6182**
COLON	**0.85**	0.7167	0.4667	0.6667	0.75	0.8333	0.6333	0.8
SRBCT	**0.8833**	0.7167	0.5667	0.7167	0.7333	0.7667	0.6833	0.85
LUNG2	**0.925**	0.92	0.825	0.79	0.83	0.88	0	0.9
LYMPHOMA	**0.9833**	0.8833	0.7333	0.9333	0.9167	0.9667	0.9333	0.9
GLIOMA	0.56	0.5	0.46	0.54	0.54	0.66	0	**0.76**
MLL	0.9	**0.9714**	0.9571	0.7714	0.8143	0.9429	0	0.9286
PROSTATE	**0.94**	0.7	0.77	0.85	0.84	0.88	0.76	0.78
DLBCL	0.875	0.925	0.825	0.775	**0.9375**	0.925	0	0.8125
LEUKEMIA	**0.9143**	0.7	0.6	0.8857	**0.9143**	**0.9143**	0.8	0.8571
ARCENE	**0.865**	0.595	0.675	0.62	0.705	0.63	0	0.765
AVG. ACCURACY	**0.8418**	0.7742	0.7253	0.7217	0.7519	0.7721	0.3399	0.8296
AVG. RNAKS	**6.3929**	4.2857	3.8929	3.7857	5.0714	5.1786	2.0714	5.3214

Table 10
Predictive accuracy using SVM as the base classifier.

Data set	OFS-Density	Grafting	α-investing	OSFS	Fast-OSFS	SAOLA	OS-SA	OFS-A3M
IONOSPHERE	0.8229	0.8686	0.8657	0.8714	**0.8771**	0.8686	0	0.7771
WDBC	0.9614	0.9631	0.9737	0.9649	0.9631	0.9139	0.1912	**0.9684**
SONAR	**0.7637**	0.7398	0.7336	0.7546	0.7531	0.7236	0.7388	0.714
HILL	0.5364	0.5124	0.5554	0	0	0	0	**0.538**
COLON	**0.8833**	0.65	0.65	0.7833	0.7667	0.8667	0.6333	0.8333
SRBCT	0.8	0.7333	0.3667	0.7	0.7	0.8167	0.7167	**0.9167**
LUNG2	0.915	**0.94**	0.9	0.865	0.89	0.9	0	0.915
LYMPHOMA	**1**	0.8667	0.7667	0.9667	0.9	0.9667	0.9333	0.8667
GLIOMA	0.68	0.64	0.44	0.58	0.6	0.62	0	**0.7**
MLL	0.8857	0.9714	**0.9857**	0.8429	0.8571	0.9	0	0.9143
PROSTATE	**0.96**	0.66	0.84	0.9	0.91	0.87	0.86	0.78
DLBCL	0.8875	**0.9375**	0.855	0.85	0.925	0.925	0	0.85
LEUKEMIA	0.9143	0.6714	0.6857	0.9	**0.9714**	0.9429	0.8714	0.9
ARCENE	**0.82**	0.68	0.76	0.65	0.695	0.625	0	0.75
AVG. ACCURACY	**0.8450**	0.7738	0.7413	0.7592	0.7720	0.7813	0.3531	0.8159
AVG. RANKS	**6.1786**	4.6429	4.1429	4.2500	4.8571	4.8214	2.1786	4.9286

teen of the fourteen datasets. Thus, OFS-Density is superior to FSV.

In sum, OFS-Density provides best overall performance on these data sets with the same number of selected features and gets the highest mean predictive accuracy and ranks with KNN, SVM, and CART.

5.5. OFS-Density vs. online streaming feature selection methods

In this subsection, we compare our algorithm with seven state-of-the-art online feature selection methods: Grafting [4], Alpha-investing [5], OSFS [6], Fast-OSFS [6], SAOLA [3], OS-NRRSARA-SA [9] and OFS-A3M [10].

All aforementioned algorithms are implemented in MATLAB [46]. For we cannot get the source code of OS-NRRSARA-SA, we implemented it by ourselves. The significance level α is set to 0.01 for OSFS, Fast-OSFS, and SAOLA. For Grafting, the parameter λ is set to 0.5. For Alpha-investing, the parameters are set to the values used in [5]. For OS-NRRSARA-SA, it can not deal with real-valued data directly. In order to convert real-valued data to discrete value data, we used the method proposed by Guyon and Elisseeff [2].

Tables 9–11 summarize the predictive accuracy of OFS-Density against the other seven algorithms using the KNN ($k = 1$), SVM and CART classifiers. Tables 12 and 13 show the running time and the number of selected features of OFS-Density against other algorithms. If the algorithm selects all the features in data sets or selects none of the features, we set the predictive accuracy and the number of selected features to 0. The p-values of Friedman test on KNN, SVM, CART, running time and number of selected features are 1.3499e−04, 0.0056, 0.0179, 1.3351e−28 and 3.8769e−14 respectively. Thus, there is a significant difference between OFS-Density and other seven competing algorithms respectively on predictive accuracy, running time and number of selected features. According to the Nemenyi test, the value of CD (critical difference) is 2.8085.

From Tables 9–13 , we have the following observations.

- OFS-Density vs. Grafting. With the Friedman test and Nemenyi test, there is no significant difference between OFS-Density and Grafting on predictive accuracy with KNN, SVM, and CART, but there is a significant difference on the number of selected features. OFS-Density outperforms Grafting on nine of the fourteen datasets at least in predictive accuracy, while Grafting selects the most number of features among all these compared methods. Thus, there must a lot of redundant features in the selected feature subset with Grafting. Meanwhile, OFS-Density is faster than Grafting.
- OFS-Density vs. Alpha-investing. Alpha-investing is the fastest algorithm among all these compared algorithms. There is no significant difference between OFS-Density and Alpha-investing on predictive accuracy with KNN, SVM, and CART. OFS-Density outperforms Alpha-investing on ten of the fourteen datasets at least. In the same time, the features selected by Alpha-investing cannot fit for some datasets well. For instance, Alpha-investing only gets the predictive accuracy of around 0.3 and 0.4 on dataset SRBCT with KNN and SVM respectively. For some data sets, such as COLON, SRBCT, and LEUKEMIA, Alpha-investing only selects one or two features. The reason is that these data

Table 11
Predictive accuracy using CART as the base classifier.

Data set	OFS-Density	Grafting	α-investing	OSFS	Fast-OSFS	SAOLA	OS-SA	OFS-A3M
IONOSPHERE	**0.8971**	0.8886	0.8743	0.8743	0.8829	0.8829	0	0.8457
WDBC	0.9209	0.9263	0.9227	**0.9403**	0.9368	0.891	0.1737	0.9174
SONAR	0.6722	0.7113	0.7336	0.6351	0.6802	0.6817	0.7446	**0.7504**
HILL	0.5901	0.5066	0.5884	0	0	0	0	**0.5992**
COLON	0.7333	0.65	0.5333	0.7167	0.7667	**0.8333**	0.6667	0.7167
SRBCT	0.8167	0.5667	0.6333	0.7167	0.7333	**0.8833**	0.7	0.7833
LUNG2	0.855	**0.915**	0.81	0.78	0.795	0.83	0	0.815
LYMPHOMA	0.85	0.7833	0.7667	0.8667	0.8667	0.85	**0.95**	0.8833
GLIOMA	0.54	0.56	0.4	0.6	0.5	**0.62**	0	0.58
MLL	**0.9143**	0.8286	0.7429	0.7714	0.9	0.8571	0	0.7857
PROSTATE	**0.9**	0.78	0.84	0.89	0.85	0.83	0.82	0.76
DLBCL	0.8	0.6925	0.775	0.7975	0.81	0.8	0	**0.825**
LEUKEMIA	0.8857	0.7857	0.5714	**0.9143**	0.8714	0.8714	0.8571	0.8429
ARCENE	0.7	0.69	**0.76**	0.615	0.72	0.655	0	0.7
AVG. ACCURACY	**0.7910**	0.7346	0.7108	0.7227	0.7366	0.7489	0.3508	0.7717
AVG. RANKS	**5.8929**	4.0000	3.5357	4.4286	5.2857	5.1071	2.6786	5.0714

Table 12
Running time (seconds).

Data set	OFS-Density	Grafting	α-investing	OSFS	Fast-OSFS	SAOLA	OS-SA	OFS-A3M
IONOSPHERE	0.3373	0.2074	**0.0021**	0.1265	0.0162	0.0117	0.162	0.0117
WDBC	0.6073	3.6651	**0.0036**	0.1225	0.0584	0.0136	0.3312	0.0136
SONAR	0.1874	0.4152	**0.0037**	0.0503	0.0216	0.0159	3.2413	0.0159
HILL	10.8671	33.1627	**0.0109**	0.015	0.0149	0.0153	0.5784	0.0153
COLON	0.9031	4.1893	**0.0716**	0.4131	0.2945	0.3117	35.7356	0.3117
SRBCT	1.0019	7.5742	**0.0912**	1.8171	0.5312	0.8129	87.2487	0.8129
LUNG2	9.6493	10.607	**0.6714**	62.784	3.3015	2.172	47.6706	2.172
LYMPHOMA	3.4191	6.0938	**0.2181**	10.2915	2.0148	4.4564	68.2504	4.4564
GLIOMA	1.8031	2.9393	**0.2362**	5.0781	1.3078	2.3033	17.4906	2.3033
MLL	3.102	2.0967	**0.4509**	12.4407	2.0482	4.9122	39.708	4.9122
PROSTATE	4.51	9.8044	**0.3489**	2.4136	1.1444	1.4457	186.6152	1.4457
DLBCL	3.7305	2.6564	**0.4503**	3.0092	1.204	1.5468	52.3555	1.5468
LEUKEMIA	3.801	7.9587	**0.4426**	4.2189	1.3923	1.9461	121.5487	1.9461
ARCENE	36.2364	70.4346	**0.9856**	9.0093	2.0664	3.049	446.1125	3.049
AVG. ACCURACY	5.7253	11.5574	**0.2847**	7.9849	1.1011	1.6437	79.0749	1.6437
AVG. RANKS	5.5714	6.5000	1.0000	5.7143	2.5714	3.5714	7.5000	3.5714

Table 13
The number of selected features.

Data set	OFS-Density	Grafting	α-investing	OSFS	Fast-OSFS	SAOLA	OS-SA	OFS-A3M
IONOSPHERE	3.7	31.7	7.7	3.7	4	3.9	0	5.8
WDBC	5.7	15.7	18.8	3	4	2	3.1	15.2
SONAR	4.1	29.4	12.1	2.9	3	2.6	11.4	23
HILL	6	1	9.3	0	0	0	0	19.1
COLON	5.8	66.3	1	1.9	2.5	3.9	7.4	32.1
SRBCT	4.6	74.9	1	2.3	5.1	20.3	8.2	11.7
LUNG2	20.5	166.7	37.5	6	9.9	29.7	0	21.7
LYMPHOMA	25.8	71.3	3.4	3.2	5.6	37	4.1	6.9
GLIOMA	5.5	58.4	3.4	1.5	4	16.7	0	21.5
MLL	8.7	53.6	9.5	2.6	5.1	32.9	0	8.9
PROSTATE	5.4	114.2	2	1.7	3.5	11.7	6.9	50.1
DLBCL	10.4	61.2	7.3	2.3	5.1	20.2	0	13
LEUKEMIA	4.2	81.6	1.9	2.6	5.3	20.3	6.4	14.6
ARCENE	60.4	122.3	8.3	2.5	5.5	19.1	0	44.4
AVG. ACCURACY	12.2	67.7	8.8	2.5	4.4	15.7	3.3	20.5
AVG. RANKS	4.5357	7.7143	4.2143	1.9286	3.4643	5.0357	2.8929	6.2143

sets are very sparse and Alpha-investing can only select the first few features of these data sets.

- OFS-Density vs. OSFS. There is no significant difference between OFS-Density and OSFS on predictive accuracy, running time and number of selected features. OFS-Density outperforms OSFS on eleven of the fourteen datasets at least. On dataset HILL, OSFS cannot select any features and gets the prediction accuracy 0. In addition, OFS-Density is faster than OSFS in running time. OSFS selects the least number of features among all these compared algorithms. Thus, some important information is probably missed which causes the low predictive accuracy.
- OFS-Density vs. Fast-OSFS. There is no significant difference between OFS-A3M and Fast-OSFS on predictive accuracy. OFS-A3M performs better than Fast-OSFS on ten of the fourteen datasets. Meanwhile, Fast-OSFS is faster than OFS-A3M. However, as similar to OSFS, Fast-OSFS also selects very few features on data sets, which leads to the missing of some important information.
- OFS-Density vs. SAOLA. There is no significant difference between OFS-Density and SAOLA on predictive accuracy with KNN, SVM, and CART. SAOLA is faster than OFS-Density and selects more features than OFS-Density. However, OFS-A3M outperforms SAOLA on nine of the fourteen datasets at least in pre-

dictive accuracy. Thus, the features selected by OFS-Density is more discriminative. Meanwhile, on the data set HILL, SAOLA cannot select any features and get the predictive accuracy 0. This demonstrates that SAOLA cannot handle some types of data well.

- OFS-Density vs. OS-NRRSARA-SA. There is a significant difference between OFS-Density and OS-NRRSARA-SA on predictive accuracy. On seven of the fourteen datasets, OS-NRRSARA-SA selects all the features of the datasets and we set the predictive accuracy and number of selected features to 0. The main reason for this is OS-NRRSARA-SA cannot deal with continues features directly and it can not select discriminative features after data converted. Meanwhile, OS-NRRSARA-SA spends the maximum running time among all these compared methods. OS-NRRSARA-SA uses the classical rough set for feature selection which makes it need not set any parameters before learning. However, it cannot deal with real-valued data directly and cannot handle some datasets well.
- OFS-Density vs. OFS-A3M. There is no significant difference between OFS-Density and OFS-A3M on predictive accuracy. OFS-Density performs a little better than OFS-A3M. Meanwhile, OFS-A3M runs faster and selects more features. Similar to OFS-Density, OFS-A3M also uses adaptive neighborhood rough set relation for feature selection. However, OFS-A3M uses the exactly equal constraint for feature redundant analysis, which makes it select more features and cause more redundancy.

In our experiments, we have conducted 10-fold cross-validation on each data set. We randomly divided the instances of each data set into 10 folds. The instances used for feature selection in Sections 5.4 and 5.5 are different. Thus, the corresponding selected features in Sections 5.4 and 5.5 are probably different. For OFS-Density, there is not only one combination of features, which can make the dependency of the selected feature subset achieve the maximal. Thus, although given the same data set, if the training instances are different, the final selected feature subset will be probably different too.

In sum, OFS-Density is not faster than some compared methods, but it outperforms all competing algorithms on predictive accuracy.

6. Conclusion

In this paper, we proposed a new method for online streaming feature selection. Our new algorithm is based on neighborhood rough set theory which does not require domain information before learning. We proposed a new density neighborhood relation which automatically decides the number of neighbors during dependency calculation by the density information of the surrounding instances. With this new neighborhood relation, we need not specify any parameters in advance. Meanwhile, we use a fuzzy equal constraint for redundancy analysis which makes the selected feature subset small and discriminative. As compared to eight traditional feature selection methods and seven state-of-the-art online streaming feature selection algorithms, the proposed algorithm is superior to traditional feature selection methods with the same number of features and performs better than online streaming feature selection algorithms in an online manner. As we have known, neighborhood rough set is one of tolerance rough sets. In our future work, we will attempt to apply fuzzy rough sets in online streaming feature selection.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China under grant 2016YFB1000901, the National Natural Science Foundation of China under grants (61503112, 61673152) and the US National Science Foundation under grant IIS-1652107.

References

[1] H. Liu, H. Motoda, Computational Methods of Feature Selection, Chapman and Hall/CRC Press, 2007.
[2] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Mach. Learn. Res. 3 (2003) 1157–1182.
[3] K. Yu, X. Wu, W. Ding, J. Pei, Scalable and accurate online feature selection for big data, ACM Trans. Knowl. Discov. Data 11 (2) (2016).
[4] S. Perkins, J. Theiler, Online feature selection using grafting, in: Proceedings of the 20th International Conference on Machine Learning, 2003, pp. 592–599.
[5] J. Zhou, D.P. Foster, R.A. Stine, L.H. Ungar, Streamwise feature selection, J. Mach. Learn. Res. 3 (2) (2006) 1532–4435.
[6] X. Wu, K. Yu, W. Ding, H. Wang, X. Zhu, Online feature selection with streaming features, IEEE Trans. Pattern Anal. Mach. Intell. 35 (5) (2013) 1178–1192.
[7] J. Wang, P. Zhao, S.C. Hoi, R. Jing, Online feature selection and its applications, IEEE Trans. Knowl. Data Eng. 26 (3) (2013) 698–710.
[8] J. Wang, M. Wang, P. Li, L. Liu, Z. Zhao, X. Hu, X. Wu, Online feature selection with group structure analysis, IEEE Trans. Knowl. Data Eng. 27 (2015) 3029–3041.
[9] S. Eskandari, M. Javidi, Online streaming feature selection using rough sets, Int. J. Approximate Reasoning 69 (C) (2016) 35–57.
[10] P. Zhou, X. Hu, P. Li, A new online feature selection method using neighborhood rough set, in: Proceedings of the 8th IEEE International Conference on Big Knowledge, 2017.
[11] P. Zhou, X. Hu, P. Li, X. Wu, Online feature selection for high-dimensional class-imbalanced data, Knowl. Based Syst. 136 (2017) 187–199.
[12] Z. Pawlak, Rough Sets - Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht , Boston, 1991.
[13] R. Jensen, Q. Shen, Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches, IEEE Trans. Knowl. Data Eng. 16 (12) (2004) 1457–1471.
[14] X. Zhang, C. Mei, D. Chen, J. Li, Feature selection in mixed data: a method using a novel fuzzy rough set-based information entropy, Pattern Recognit. 56 (1) (2016) 1–15.
[15] R. Jensen, Q. Shen, Fuzzy-rough sets assisted attribute selection, IEEE Trans. Fuzzy Syst. 15 (1) (2007) 73–89.
[16] Y. Lin, Q. Hu, J. Liu, J. Li, Streaming feature selection for multi-label learning based on fuzzy mutual information, IEEE Trans. Fuzzy Syst. (2017).
[17] R. Jensen, Q. Shen, New approaches to fuzzy-rough feature selection, IEEE Trans. Fuzzy Syst. 17 (4) (2009) 824–838.
[18] S. Vluymans, A. Fernndez, Y. Saeys, C. Cornelis, F. Herrera, Dynamic affinity-based classification of multi-class imbalanced data with one-versus-one decomposition: a fuzzy rough set approach, Knowl. Inf. Syst. 56 (1) (2018) 55–84.
[19] Q. Hu, D. Yu, J. Liu, C. Wu, Neighborhood rough set based heterogeneous feature subset selection, Inf. Sci. 178 (18) (2008) 3577–3594.
[20] Q. Hu, J. Liu, D. Yu, Mixed feature selection based on granulation and approximation, Knowl. Based Syst. 21 (4) (2008) 294–304.
[21] Y. Lin, J. Li, P. Lin, G. Lin, J. Chen, Feature selection via neighborhood multi–granulation fusion, Knowl. Based Syst. 67 (2014) 162–168.
[22] Z. Zeng, H. Zhang, R. Zhang, Y. Zhang, A novel feature selection method considering feature interaction, Pattern Recognit. 48 (8) (2015) 2656–2666.
[23] S. Tabakhi, P. Moradi, Relevance - redundancy feature selection based on ant colony optimization, Pattern Recognit. 48 (9) (2015) 2798–2811.
[24] A. Pecli, M.C. Cavalcanti, R. Goldschmidt, Automatic feature selection for supervised learning in link prediction applications: a comparative study, Knowl. Inf. Syst. 56 (1) (2018) 85–121.
[25] JieFeng, L. Jiao, F. Liu, TaoSun, X. Zhang, Unsupervised feature selection based on maximum information and minimum redundancy for hyperspectral images, Pattern Recognit. 51 (C) (2016) 295–309.
[26] R. Sheikhpour, M. Sarram, S. Gharaghani, M. Chahooki, A survey on semi-supervised feature selection methods, Pattern Recognit. 64 (C) (2016) 141–158.
[27] Q. Gu, Z. Li, J. Han, Generalized fisher score for feature selection, in: Proceedings of Conference on Uai, 2011.
[28] M. Robnik-Sikonja, I. Kononenko, Theoretical and empirical analysis of relieff and rrelieff, Mach. Learn. 53 (1–2) (2003) 23–69.
[29] J.R. Vergara, P.A. Estvez, A review of feature selection methods based on mutual information, Neural Comput. Appl. 24 (1) (2014) 175–186.
[30] P.S. Bradley, O.L. Mangasarian, Feature selection via concave minimization and support vector machines, in: Proceedings of the Fifteenth International Conference on Machine Learning, 1998, pp. 82–90.
[31] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, Adv. Neural Inf. Process. Syst. 17 (2005) 507–514.
[32] G. Roffo, S. Melzi, M. Cristani, Infinite feature selection, in: Proceedings of IEEE International Conference on Computer Vision, 2015, pp. 4202–4210.
[33] H. Zeng, Y. ming Cheung, Feature selection and kernel learning for local learning-based clustering, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1532–1547.
[34] M. Wu, B. Scholkopf, A local learning approach for clustering, in: Proceedings of International Conference on Neural Information Processing Systems, 2006, pp. 1529–1536.
[35] M. Wang, H. Li, D. Tao, K. Lu, X. Wu, Multimodal graph-based reranking for web image search, IEEE Trans. Image Process. 21 (11) (2012) 4649–4661.

[36] W. Ding, T.F. Stepinski, Y. Mu, L. Bandeira, R. Ricardo, Y. Wu, Z. Lu, T. Cao, X. Wu, Subkilometer crater discovery with boosting and transfer learning, ACM Trans. Intell. Syst. Technol. 2 (4) (2011) 1–22.
[37] R. Diao, N.M. Parthalin, Q. Shen, Dynamic feature selection with fuzzy-rough sets, in: IEEE International Conference on Fuzzy Systems, 2013, pp. 1–7.
[38] L. T, G. Y, Computing on binary relations i: data mining and neighborhood systems, in: Proceedings of the Rough Sets in Knowledge Discovery, 1998, pp. 107–121.
[39] J. Zhang, T. Li, D. Ruan, D. Liu, Neighborhood rough sets for dynamic data mining, Int. J. Intell. Syst. 27 (4) (2012) 317–342.
[40] R. Jensen, Q. Shen, Computational intelligence and feature selection: rough and fuzzy approaches, IEEE Press Series on Computational Intelligence, JOHN WILEY & SONS, 2008.
[41] K. Yang, Z. Cai, J. Li, G. Lin, A stable gene selection in microarray data analysis, BMC Bioinform. 7 (2006) 228.
[42] L. Yu, C. Ding, S. Loscalzo, Stable feature selection via dense feature groups, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008.
[43] J. Demar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (1) (2006) 1–30.
[44] M. Wasikowski, X. Chen, Combating the small sample class imbalance problem using feature selection, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1388–1400.
[45] G. Roffo, Feature selection library (matlab toolbox), arXiv:1607.01327[cs.CV](2016).
[46] K. Yu, W. Ding, X. Wu, Lofs: library of online streaming feature selection, Knowl. Based Syst. 113 (1–3) (2016).

Peng Zhou is currently working toward the PhD degree at Hefei University of Technology, China. His research interests are in data mining and knowledge engineering.

Xuegang Hu received the BS degree from the Department of Mathematics at Shandong University, and the MS and PhD degrees at Hefei University of Technology. He is a professor in the School of Computer Science and Information Hefei University of Technology, China, and the directorgeneral of Association of Higher Education at Anhui Province. His research interests include data mining and knowledge engineering.

Peipei Li is currently a lecturer at Hefei University of Technology, China. She received her B.S., M.S. and Ph.D. degrees from Hefei University of Technology in 2005, 2008, 2013 respectively. She was a research fellow at Singapore Management University from 2008 to 2009. She was a student intern at Microsoft Research Asia between Aug. 2011 and Dec. 2012. Her research interests are in data mining and knowledge engineering.

Xindong Wu is currently the director of School of Computing and Informatics and professor at University of Louisiana at Lafayette. From 2001 to 2015, he was a Professor of Computer Science at the University of Vermont (USA). He is a Fellow of the IEEE and the AAAS. He holds a Ph.D. in Artificial Intelligence from the University of Edinburgh, Britain. He is the founder and current Steering Committee Chair of the IEEE International Conference on Data Mining and the founder and current Editor-in-Chief of Knowledge and Information Systems. He was the Editor-in-Chief of the IEEE Trans. on Knowledge and Data Eng. from 2005 to 2008. His research interests include data mining, Big Data analytics etc.