Information Sciences 481 (2019) 258–279

Contents lists available at ScienceDirect

Information Sciences

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

Online streaming feature selection using adapted Neighborhood Rough Set

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

[a] Hefei University of Technology, Hefei 230009, China
[b] Anhui Province Key Laboratory of Industry Safety and Emergency Technology, Hefei 230009, China
[c] University of Louisiana, Lafayette, LA 70504, USA

ARTICLE INFO

Article history:
Received 29 August 2017
Revised 27 December 2018
Accepted 29 December 2018
Available online 31 December 2018

Keywords:
Feature selection
Online streaming feature selection
Feature streams
Neighborhood rough set
Adapted neighbors

ABSTRACT

Online streaming feature selection, as a new approach 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. However, most of the existing online streaming feature selection methods need the domain information before learning and specifying the parameters in advance. It is hence a challenge to select unified and optimal parameters before learning for all different types of data sets. In this paper, we define a new Neighborhood Rough Set relation with adapted neighbors named the Gap relation and propose a new online streaming feature selection method based on this relation, named OFS-A3M. OFS-A3M does not require any domain knowledge and does not need to specify any parameters in advance. With the "maximal-dependency, maximal-relevance and maximal-significance" evaluation criteria, OFS-A3M can select features with high correlation, high dependency and low redundancy. Experimental studies on fifteen different types of data sets show that OFS-A3M 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

Given a nonempty finite set of attributes (features) $A = C \cup D$, where C is the condition feature set and D is the decision feature set, the classical feature selection problem aims to select a subset of C, which can be used to derive a mapping function from samples to classes that is "as good as possible" according to some criterion [14,18]. A variety of feature selection approaches have been developed during the last three decades [6,8,19,22,26,27].

All these traditional feature selection methods assume that all the features in C are available before learning. However, in some real-world applications [31], features are generated dynamically and arrive in order. Streaming features are defined as features that flow in one by one over time whereas the number of training examples remains fixed [33]. A real-world example is the Mars crater detection from high-resolution planetary images [4]. It is infeasible to acquire the entire feature set which would have a near-global coverage of the Martian surface. Meanwhile, with the increasing of the scale of data, traditional batch feature selection methods cannot meet the demand for efficiency anymore [1,16,34]. An example for this

* Corresponding author.
E-mail addresses: doodzhou@hotmail.com (P. Zhou), jsjxhuxg@hfut.edu.cn (X. Hu).

https://doi.org/10.1016/j.ins.2018.12.074
0020-0255/

is the Web Spam Corpus 2011, a collection of approximately 330,000 spam web pages and 16,000,000 features (attributes) [28].

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 [20,21,30,33,37,40]. For example, OSFS (Online Streaming Feature Selection) [33], a method of selecting strongly relevant and non-redundant features on the fly and SAOLA (a Scalable and Accurate OnLine Approach) [37], which employs novel online pairwise comparison techniques to address the extremely high dimensionality and highly scalable challenges in an online manner. However, both OSFS and SAOLA need to specify the parameter α in advance, which affects the dependence and independence test. All aforementioned algorithms need the domain information to specify some parameters in advance. It is hence a challenge to select unified and optimal parameters before learning for different datasets.

Rough Set Theory, proposed by Pawlak, has been proven to be an effective tool for feature selection, rule extraction, and knowledge discovery [17]. One of the most important advantages of Rough Set based data mining is that they do not require any domain knowledge other than the given dataset. There are many works using rough sets for feature selection [5,12,15,24]. For example, OS-NRRSARA-SA [5] is a classical Rough Set based online streaming feature selection method, which needn't specify any parameters in advance. However, the classical Rough Set was originally proposed to deal with categorical data. In real-world applications, there are many numerical features in data sets and classical Rough Set based feature selection algorithms cannot handle numerical data directly. Thus, Neighborhood Rough Set which supports both continuous and discrete data was proposed to deal with this challenge [9–11,13,39]. Nevertheless, all these methods mentioned above were proposed for traditional feature selection and they cannot handle online streaming feature selection directly. Meanwhile, most of them need to specify some parameter values in advance and it is always difficult to select unified and optimal values for all different types of data sets.

Motivated by this, we define a new Neighborhood Rough Set relation named Gap which automatically selects the number of neighbors for each target object by its surrounding instance distribution. In terms of this relation, a new online streaming feature selection algorithm (OFS-A3M) is proposed to handle feature selection in an online manner. Our contributions are as follows:

- Most of existing online streaming feature selection methods need domain information to set parameters before learning. However, in real-world applications, we cannot always require the domain knowledge in advance. Based on Neighborhood Rough Set Theory, our proposed OFS-A3M algorithm does not need domain knowledge before learning.
- It is a challenge to select unified and optimal parameters in advance for those methods which need to specify parameters before learning. In order to design a non-parametric feature selection method, we propose a new neighborhood relation using the distribution information of the surrounding instances, named Gap. With this new neighborhood relation, OFS-A3M can automatically select a proper number of neighbors during online feature selection and needn't specify any parameters in advance.
- In terms of three maximal evaluation criteria (maximal-dependency, maximal-relevance, and maximal-significance), OFS-A3M can select features with high correlation, high dependency and low redundancy.
- In order to validate the effectiveness of our new neighborhood relation, we conduct a comparison between δ neighborhood and k-nearest neighborhood in detail. The results of Friedman test show that there is no significant difference between Gap and the other two relations with the optimal parameter values. Meanwhile, extensive experimental studies compared with seven traditional feature selection methods and four 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.

This new adapted Neighborhood Rough Set method was first introduced in our conference paper [41]. In comparison with the preliminary version, we have improvements in the following aspects: (1) we have performed a more comprehensive survey of existing related works; (2) we have provided more detailed descriptions of the theory of Neighborhood Rough Set and our new method; and (3) we have conducted more experiments, discussions and analysis.

The rest 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 defined Neighborhood Rough Set relation and a new online streaming feature selection approach based on this relation. Section 5 reports experimental results and analyzes all experimental algorithms. Section 6 concludes the paper.

2. Related work

Feature Selection is an important technology for machine learning and data mining. There are many representative algorithms for traditional feature selection, such as ReliefF [22], Fisher Score [6], MI (Mutual Information) [27], mRMR (minimal Redundancy and Maximal Relevance) [19], Laplacian Score [8], LASSO (least absolute shrinkage and selection operator) [26] and so on. 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 [14].

All aforementioned feature selection methods assume that all features in the feature space are available before learning. However, features may exist in a streaming format for some real-world applications [4,31]. Online feature selection with

streaming features has attracted much attention in recent years and played a critical role in dealing with high-dimensional and big data problems [5,20,30,33,37,40]. There are two major reasons for online streaming feature selection [5]: (1) The feature space is unknown or even infinite and (2) the feature space is known but feature streaming offers many advantages.

More specifically, Perkins and Theiler [20] considered the problem of online feature selection and proposed the Grafting algorithm based on a stagewise gradient descent approach. Grafting treats feature selection as an integral part of learning a predictor within a regularized framework. A new arriving feature is added to the selected features if the improvement in the model accuracy is greater than a predefined threshold λ. Grafting needs the information of the global feature space to choose a good value for the important regularization parameter λ in advance, and it is hence weak in the handling of streaming features.

Zhou et al. [40] proposed two algorithms of information-investing and alpha-investing, based on streamwise regression for online feature selection. Alpha-investing does not need a global model and it is one of the penalized likelihood ratio methods. However, these two algorithms require prior knowledge about the structure of the feature space to heuristically control the choice of candidate feature selection.

Wu et al. [33] presented an online streaming feature selection framework with two algorithms called OSFS (Online Streaming Feature Selection) and fast-OSFS. OSFS contains two major steps, including online relevance analysis (discards irrelevant features) and online redundancy analysis (eliminates redundant features). Although these two algorithms can select features with high relevancy and low redundancy, it uses a conditional independence test which needs a large number of training instances. This may lead to information missing during online feature selection on the datasets with high dimensionality and small samples.

Yu et al. [37] proposed the SAOLA approach (a Scalable and Accurate Online feature selection Approach) for high dimensional data. SAOLA employs novel online pairwise comparison techniques and maintains a parsimonious model over time in an online manner. SAOLA addresses two challenges in big data applications: extremely high dimensionality and its highly scalable requirement of feature selection. Nevertheless, SAOLA needs to set the parameter α in advance which affects the dependence and independence test.

Javidi and Eskandari [12] considered the problem of streamwise feature selection and proposed a method from the Rough Set perspective. The main motivation for this consideration is that Rough Set based data mining does not require any domain knowledge other than the given dataset. This new algorithm (SFS-RS) uses the significance analysis concepts in Rough Set Theory to control the unknown feature space. SFS-RS needn't specify any parameters before learning. However, SFS-RS is a classical Rough Set based method which cannot handle numerical features directly.

Eskandari and Javidi [5] proposed a Rough Set based method for online streaming feature selection, named OS-NRRSARA-SA. Unlike the classical Rough Set-based attribute reduction methods which only use the information contained in the positive region, OS-NRRSARA-SA considers the boundary and positive regions. Meanwhile, OS-NRRSARA-SA uses a noise resistant dependency measure to search for reduces. By using Rough Set Theory, OS-NRRSARA-SA does not need to specify any parameters before learning. However, like SFS-RS, OS-NRRSARA-SA cannot deal with numerical features directly.

Wang et al. [29] considered online streaming feature selection from the Rough Set perspective and proposed an uncertainty measure framework to address this issue. By specifying the uncertainty measure with conditional information entropy (CIE), a new algorithm was proposed based on the framework, called CIE-OSFS. CIE-OSFS does not need prior knowledge to deliver credible results and is robust to the changing of streaming orders. Nevertheless, similar to other Rough Set based methods, CIE-OSFS cannot handle numerical features directly.

Rahmaninia and Moradi [21] considered the challenges of high computational cost, the stability of the generated results and the size of the final feature subset in online streaming feature selection and proposed two feature selection methods called OSFSMI and OSFSMI-k. These two methods employ mutual information in a streaming manner to evaluate the relevancy and redundancy of features. However, OSFSMI and OSFSMI-k need to set the parameter β which controls redundancy penalty in advance. In order to choose an optimal value of this parameter, these new methods need domain knowledge before learning.

Zhou et al. [42] 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 was designed for class-imbalanced data and it needs to specify the parameter K in advance.

Rough Set Theory has attracted widespread attention after it was proposed [17]. The classical Rough Set Theory was originally proposed to deal with categorical data. However, in real-world applications, there are many numerical features in the data sets. Thus, Neighborhood Rough Set which supports both continuous and discrete data was proposed to deal with this challenge [9–11,13,39].

More specifically, Hu et al. [9] proposed a feature selection method for numerical and categorical mixed attributes by generalizing Pawlak's Rough Set model into δ Neighborhood Rough Set model and k-nearest-neighbor Rough Set model. The objects with numerical attributes are granulated with δ neighborhood relations or k-nearest-neighbor relations, while objects with categorical features are granulated with equivalence relations. Paper [10] presented a Neighborhood Rough Set model to deal with the problem of heterogeneous feature subset selection. This neighborhood model is used to reduce numerical and categorical features by assigning different thresholds for different kinds of attributes. In paper [13], hybridization of particle swarm optimization (PSO)-based Rough Set feature selection technique was proposed for achieving a minimal set

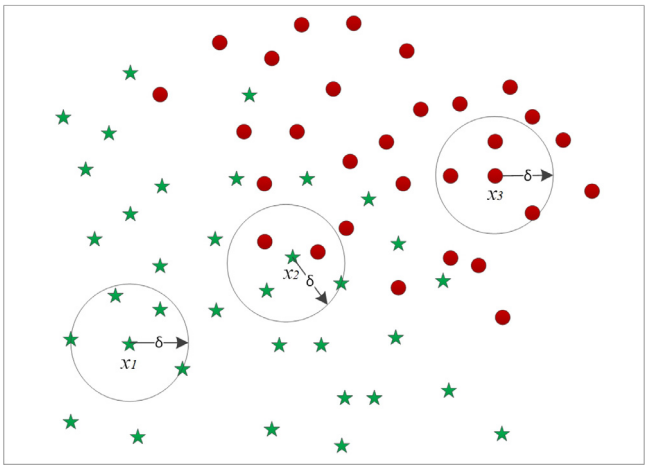

Fig. 1. δ Neighborhood Rough Set.

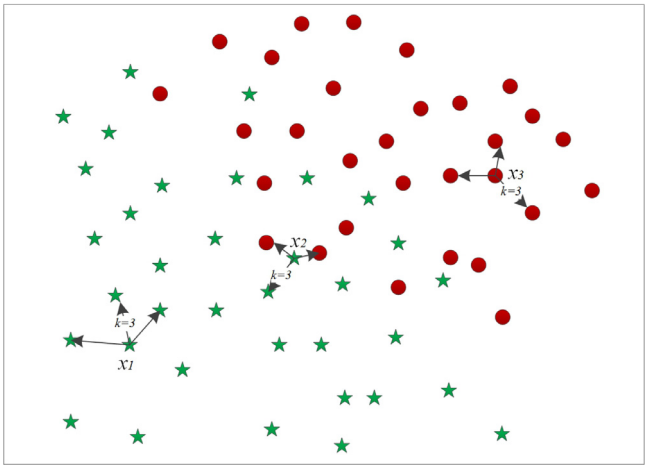

Fig. 2. k-nearest Neighborhood Rough Set (k=3).

of relevant features from extracted features. The selected features are applied to the proposed novel Neighborhood Rough Set classifier (NRSC) method for classification of multi-class motor imagery.

Nevertheless, all of these methods mentioned above are proposed for the traditional feature selection problem. Meanwhile, all these Neighborhood Rough Set based feature selection algorithms need to specify some parameters before learning. It is hence a challenge to select unified and optimal parameters before learning for different datasets. Motivated by this, we proposed a new neighborhood relation for online streaming feature selection and this new approach does not need to specify any parameters in advance. First, we give a brief introduction to the Neighborhood Rough Set Theory as follows.

3. Neighborhood Rough Set

Classical Rough Set is originally proposed to deal with categorical data [17,23]. However, in real-world applications, there are many integers-valued and real-valued data. Thus, Neighborhood Rough Set is used to replace the approximation based on the equivalence relation of the traditional Rough Set model with the neighborhood relation, which supports both continuous and discrete data sets [11,25,39]. In this section, we briefly review some basic concepts and notations of Neighborhood Rough Set as follows.

$DT = (U, A, V, h)$ is called a decision table [43], where $U = \{x_1, x_2, \ldots, x_n\}$ is a nonempty finite set of n objects, $A = C \bigcup D$, 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 the domain of attribute a. h: $U \times A \rightarrow V$ is an information function such that $h(x, a) \in V_a$ for every $x \in U$, $a \in A$. $h(x_i, a_j)$ denotes the value of object x_i on the attribute a_j. In this paper, h denotes the specific value of an object on a certain feature.

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

Definition 1. Given *DT*, 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 *DT*, let $B \subseteq C$ be a subset of attributes, $x \in U$. The neighborhood $\delta_B(x)$ of an arbitrary object x on the feature subset B is defined as:

$$\delta_B^r(x) = \{y \mid \Delta_B(x, y) \leq r, y \in U\}, \tag{1}$$

where Δ_B denotes the distance function calculating on subset B, and r is the distance radius.

Definition 3. Given *DT*, 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, denoted as $K_B(x)$.

$$K_B^k(x) = \{y \mid y \in Min_k\{\Delta(x, y)\}, y \in U, y \neq x\}, \tag{2}$$

where $Min_k\{\Delta(x, y)\}$ denotes the k nearest neighbors of x calculated on subset B.

Suppose a data set S has N instances. For each object x in S, it has $N-1$ neighbors. K-nearest neighborhood mainly considers the class information of k nearest neighbors.

Like Pawlak's Rough Set model, we give the lower and upper approximations of δ neighborhood and k-nearest neighborhood as follows.

Definition 4. Given *DT* and their neighborhood relations R over U. For $\forall X \subseteq U$, two subsets of objects, called lower and upper approximations of X in terms of a δ neighborhood relation, are defined as

$$\underline{R_\delta} X = \{x_i \mid \delta(x_i) \subseteq X, x_i \in U\} \tag{3}$$

$$\overline{R_\delta} X = \{x_i \mid \delta(x_i) \cap X \neq \emptyset, x_i \in U\} \tag{4}$$

Definition 5. Given *DT*, an arbitrary subset X of the sample space U and a family of k-nearest neighbor information granules $K(x)$, we define the lower and upper approximations in terms of k-nearest neighborhood relations as

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

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

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. 1, 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 are the boundary objects. Meanwhile, as shown in Fig 2, 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 6. Given *DT*, $B \subseteq C$, the dependency degree of B to D is defined as the ratio of consistent objects:

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

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.

For online streaming feature selection, features flow in one by one over time. 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 7. Given *DT*, a condition attribute set B ($B \subseteq C$) and a decision attribute set D, the significance of a feature f ($f \in B$) to B is defined as:

$$\sigma_B(D, f) = \gamma_B(D) - \gamma_{B \setminus \{f\}}(D) \tag{9}$$

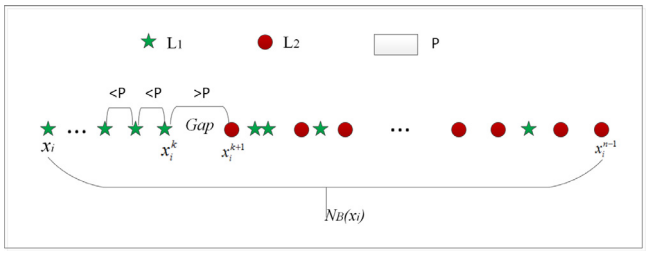

Fig. 3. Gap neighborhood.

Both δ neighborhood relation and k-nearest neighborhood relation need to specify parameters δ and k before learning. It is hence a challenge to select unified and optimal parameters before learning for different data sets. Motivated by this, we propose a new neighborhood relation which automatically selects the number of neighbors for each target object by its surrounding instances distribution. This new neighborhood relation does not need to specify any parameters before learning. More details can be seen in Section 4.

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 neighborhood relation with adapted neighbors. Three evaluation criteria of "maximal-dependency, maximal-relevance, and maximal-significance" based on the dependency between condition features and decision classes will be used for selecting features with high correlation and low redundancy. 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 $OSFS = (C, D, h', t)$ denote an online streaming feature selection framework, where 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 $D = [y_1, y_2, \ldots, y_n]^T \in R^n$ 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 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 is to derive a mapping function $h'_t : x_i \rightarrow L$ $(x_i \in C)$ at each time stamp t, which is as good as possible using a subset of features that have arrived so far.

Unlike traditional feature selection methods, we do not know the feature space before learning. Although Rough Set based data mining does not require any domain knowledge, it is still a challenge to specify unified parameters δ for δ neighborhood or k for k-nearest neighborhood before learning for all different types of data sets. Thus, we introduce a new neighborhood relation which does not need to specify any parameters before learning as follows.

4.2. Our new neighborhood relations

Definition 2 introduces the neighborhood relation with a fixed distance δ of the nearest neighbors to the target object, called δ neighborhood. Definition 3 presents the neighborhood relation with a fixed number of the nearest neighbors to the target object, called k-nearest neighborhood. However, for different datasets, the distribution of samples is asymmetrical. It is difficult to select a uniform δ for all types of data. Meanwhile, it is also a challenge for k-nearest neighborhood to select a uniform k for different types of datasets. When calculating the dependency value, it will be good to determine the number of neighbors for each target object by its surrounding instances distribution. Motivated by this, we define a new neighborhood relation which automatically selects the number of neighbors for each target object by its surrounding instances distribution as shown in Fig. 3.

Definition 8. Given $OSFS$, 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,

$$N_B(x_i) = < x_i^1, x_i^2, \ldots, x_i^j, \ldots, x_i^{n-1} > \tag{10}$$

where $\Delta_B(x_i, x_i^1) \le \Delta_B(x_i, x_i^2) \le \cdots \le \Delta_B(x_i, x_i^{n-1})$.

From x_i^1 to x_i^{n-1}, assume it is evenly distributed. We divided $\Delta_B(x_i^1, x_i^{n-1})$ into $n-1$ parts $P_1, P_2, \ldots, P_{n-1}$, where $Width_{P_1} = Width_{P_2} = \ldots = Width_{P_{n-1}} = p$. Then, each part contains one sample. Certainly, it is always a non-uniform distribution from x_i^1 to x_i^{n-1}. From x_i^1 to x_i^{n-1}, if the distance between two instances x_i^k and x_i^{k+1} is bigger than p, it is called a

Table 1
An example dataset.

$x \in U$	f_1	f_2	f_3	f_4	d
x_1	3	5.6	−66	3.05	−1
x_2	5	6.9	95	4.84	1
x_3	8	5.3	−28	5.89	1
x_4	13	12.3	−35	6.14	1
x_5	6	15.2	72	6.55	−1
x_6	5	2.6	42	10.94	1
x_7	9	6.8	−33	23.85	−1
x_8	15	8.4	15	23.85	−1

Gap between x_i^k and x_i^{k+1}, denoted as $Gap(x_i^k, x_i^{k+1})$. Thus, we use the samples between x_i and the first Gap as the nearest neighbors of x_i.

If there exist more than one neighbor with the same distance to x_i, there will be a set of many permutations of $N_B(x_i)$. However, this does not affect the final selected neighbors in terms of the Gap neighborhood relation.

Theorem 1. *Different permutations of $N_B(x_i)$ with the same distance to x_i do not affect the final selected neighbors in terms of the Gap neighborhood relation.*

Proof 1. Suppose $N_B(x_i) =< x_i^1, x_i^2, \ldots, x_i^k, x_i^{k+1}, \ldots x_i^{k+m}, \ldots, x_i^{n-1} >$, where $\Delta_B(x_i, x_i^k) = \Delta_B(x_i, x_i^{k+1}) = \ldots = \Delta_B(x_i, x_i^{k+m})$. Then, there are $m+1$ objects having the same distance to x_i. Suppose the first Gap from x_i^1 to x_i^{n-1} is $Gap(x_i^j, x_i^{j+1})$. Due to $\Delta_B(x_i^k, x_i^{k+1}) = \Delta_B(x_i^{k+1}, x_i^{k+2}) = \ldots = \Delta_B(x_i^{k+m-1}, x_i^{k+m}) = 0$, there should be $j+1 <= k$ or $j >= k+m$. If $j+1 <= k$, then $\{x_i^k, x_i^{k+1}, \ldots, x_i^{k+m}\}$ will not be considered as the nearest neighbors in terms of Gap neighborhood relation. If $j >= k+m$, then all the objects of $\{x_i^k, x_i^{k+1}, \ldots, x_i^{k+m}\}$ will be considered as the nearest neighbors together. Thus, the objects with equal distance will be considered together. The order of objects with equal distance will not affect the set of the final considered neighbors.

Based on this, we proposed a new neighborhood relation with adapted neighbors by using the Gap, denoted as $GAP(x)$ as shown in Eq. (11).

Definition 9. Given *OSFS* and a feature subset B ($B \subseteq C$), for target object x_i, let $N_B(x_i) =< x_i^1, x_i^2, \ldots, x_i^{n-1} >$ denotes all the neighbors of x_i from the nearest to the farthest on B. The width of Gap is p. The adapted neighborhood of arbitrary object $x_i \in U$ on B is defined as:

$$GAP_B(x_i) = \{x \mid x \in \{x_i^1, x_i^2, \ldots, x_i^k\}\}, \tag{11}$$

where $Gap(x_i^{m-1}, x_i^m) < p$, and $2 \le m \le k$.

More specifically, assume $D_{\max} = \Delta(x_i, x_i^{n-1})$ denotes the maximum distance from x_i to its neighbors and $D_{\min} = \Delta(x_i, x_i^1)$ denotes the minimum distance in $N_B(x_i)$. Thus, the average width of gaps for each elements in $N_B(x_i)$ is $G_{mean} = \frac{D_{\max} - D_{\min}}{n-1}$. We define the width of **Gap** as $W_{Gap} = 1.5 \times G_{mean}$(1.5 is an empirical value). From x_i^1 to x_i^{n-1}, if $Gap(x_i^k, x_i^{k+1})$ is the first Gap, which means $\Delta(x_i, x_i^{k+1}) - \Delta(x_i, x_i^k) \ge W_{Gap}$ and all the neighbors $\{x_i^j \mid 2 \le j \le k\}$ have $\Delta(x_i, x_i^j) - \Delta(x_i, x_i^{j-1}) < W_{Gap}$, we will consider $\{x_i^1, x_i^2, \ldots, x_i^k\}$ as the adapted neighbors of x_i.

Table 1 shows an example dataset used to illustrate the definition of Gap neighborhood, where x_1 – x_8 are the samples with four condition features (f_1 to f_4) and one decision features (d). The distance function is calculated with Standardized Euclidean distance (each coordinate difference between rows in data matrix X is scaled by dividing by the corresponding element of the standard deviation).

Take object x_3 and feature set $B = \{f_1, f_2\}$ as an example. First, we calculate Standardized Euclidean distances (the standard deviations of f_1 and f_2 are 4.1748 and 4.0576) between x_3 and x_i ($i \neq 3$) on B namely: $\Delta_B(x_3, x_1) = \sqrt{((8-3)/4.1748)^2 + ((5.3-5.6)/4.0576)^2} = 1.2$, $\Delta_B(x_3, x_2) = 0.8197$, $\Delta_B(x_3, x_4) = 2.1$, $\Delta_B(x_3, x_5) = 2.4865$, $\Delta_B(x_3, x_6) = 0.9794$, $\Delta_B(x_3, x_7) = 0.4405$, $\Delta_B(x_3, x_8) = 1.8426$. All the neighbors of x_3 from the nearest to the farthest are denoted as $N_B(x_3) =< x_7, x_2, x_6, x_1, x_8, x_4, x_5 >$.

For Gap neighborhood, $D_{\max} = \Delta_B(x_3, x_5) = 2.4865$, $D_{\min} = \Delta_B(x_3, x_7) = 0.4405$, and $G_{mean} = \frac{D_{\max} - D_{\min}}{7} = \frac{2.4865 - 0.4405}{7} = 0.2923$. Then $W_{Gap} = 1.5 \times G_{mean} = 1.5 \times 0.2923 = 0.4385$. From x_7 to x_5, $\Delta_B(x_7, x_2) = 0.3792$, $\Delta_B(x_2, x_6) = 0.1597$, $\Delta_B(x_6, x_1) = 0.2206$, $\Delta_B(x_1, x_8) = 0.6426$. Thus, the first GAP is $Gap(x_1, x_8) = 0.6426 > 0.4385$, and the Gap neighborhood of x_3 is $GAP_B(x_3) = \{x_7, x_2, x_6, x_1\}$.

Unlike traditional feature selection methods, online feature selection with feature streams gets features one by one over time. At the *jth* time stamp, we must decide the new arriving feature f_j whether to maintain or discard. Online streaming feature selection mainly aims to select features with high correlation and low redundancy. Thus, we introduce three evaluation criteria as follows.

4.3. Evaluation criteria of maximal-dependency, maximal-relevance and maximal-significance

For high-dimensional data sets, there are always many irrelevant and redundant features. In order to remove these features in the process of feature selection, we introduce three evaluation criteria for Rough Set based approaches as follows [15,19].

4.3.1. Maximal-dependency

Let C denote the set of m condition features of a given data set. The task of feature selection is to find a feature subset $B \subseteq C$ with d features ($d < m$) which have the largest dependency $\mathbb{D}$ on the decision attributes set D and the minimal value of d.

$$\text{Max}\{\mathbb{D}\} \quad \& \quad \text{Min}\{d\} \quad s.t. \quad \mathbb{D} = \gamma_B(D), \tag{12}$$

where $\mathbb{D} = \gamma_B(D)$ represents the dependency between the feature subset B and the decision attributes set D as shown in Eq. (8).

Theoretically, the maximal-dependency is the best evaluation criterion for feature selection with Neighborhood Rough Set. However, it is difficult to generate the resultant equivalence classes by using the maximal-dependency in the high-dimensional space. Reasons are analyzed below. First, the number of samples is often insufficient. Second, the generation of resultant equivalence classes is usually an ill-posed problem [19]. Meanwhile, the slow computational speed is another drawback of maximal-dependency. In addition, it is not suitable for online streaming feature selection because we just get one feature at each time stamp and we do not know the whole feature space in advance.

4.3.2. Maximal-relevance

Maximal-relevance is to search feature with approximates $\mathbb{D}$ using Eq. (12) with the mean value of all dependency values between individual feature f_i and target class label D:

$$\text{Max}\{\mathbb{R}\} \quad s.t. \quad \mathbb{R} = \frac{1}{|S|}\sum_{f_i \in S} \gamma_{f_i}(D), \tag{13}$$

where S is the selected feature subset.

The dependency among features which have been selected according to maximal-relevance could have rich redundancy. For instance, if two features f_i and f_j highly depend on each other, and both of them are in the candidate feature subset. The respective class discriminative power would not change a lot after we remove one of them. Thus, "maximal-relevance" can select features with high dependency to the condition classes, but it can not remove redundancy in the selected feature subset.

4.3.3. Maximal-significance

The significance of a feature f to feature set S ($f \in S$) is defined as follows:

Definition 10. Given a condition attribute set S and a decision attribute set D, the significance of a feature f ($f \in S$) to S is defined as:

$$\sigma_S(D, f) = \gamma_S(D) - \gamma_{S\setminus\{f\}}(D) \tag{14}$$

With the significance of the feature to its feature set, we can measure each feature's importance in the selected candidate subset. The maximal-significance condition can select mutually exclusive features as follows:

$$\text{Max}\{\mathbb{S}\} \quad s.t. \quad \mathbb{S} = \frac{1}{|S|}\sum_{f_i \in S}\{\sigma_S(D, f_i)\}. \tag{15}$$

In online streaming feature selection, we can not test all combinations of candidate features to maximize the dependency of the selected feature set as Eq. (12). Thus, we use the "maximal-relevance" criterion to select relevant features and discard irrelevant features at first. If a new arriving feature can increase the dependency of the selected subset, it will be selected by the "maximal-dependency" criterion. Otherwise, if the dependency of adding the new arriving feature equals to the currently selected subset, we first combine the new arriving feature and currently selected subset and then use the "maximal-significance" criterion to remove non-significant features. More details are given in Section 4.4.

4.4. Our new online streaming feature selection algorithm

$GAP_B(x)$ use different numbers of neighbors for dependency calculation which makes it competent to handle different kind of data. Given *OSFS*, $S \subseteq C$, S_R denotes the set of neighbors in terms of neighborhood relation R on the feature set S. The dependency calculation method using this new neighborhood relation denoted as *Dep-Adapted* is given in Algorithm 1. However, in terms of Gap neighborhood relation, our new algorithm deals with data sets that contain only numerical values. For categorical features, it can be handled by Classical Rough Set relation.

Algorithm 1 Dep–Adapted.

Require:
S: The target feature set;
R: Neighborhood relation,
Ensure:
$\mathbb{D}_S$: Dependency on feature set S
S_{card}: The number of positive samples on S, initialized to 0;
$||S||$: The number of instances in S;
FOREACH x_i in S
    Calculate the distance between x_i and its neighbors;
    Sort all the neighbors of x_i from the nearest to the farthest;
    Find the neighbors of x_i in term of R as $S_R(x_i)$;
    Calculate the card value of x_i as $S_R(x_i)_{card}$;
    $S_{card} = S_{card} + S_R(x_i)_{card}$;
END
$\mathbb{D}_S = S_{card}/||S||$;
return $\mathbb{D}_S$;

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. For each object x_i, we calculate the distance between x_i and its neighbors. The time complexity of this step is $O(||S||)$. In order to find the neighbors of x_i in terms of neighborhood relation R, we need to sort all the neighbors of x_i by distance. The time complexity of quicksort function is $O(n^*logn)$. Thus, the time complexity of *Dep-Adapted* is $O(||S||^{2*}log||S||)$.

Based on the dependency calculating method, we introduce our new online streaming feature selection algorithm by using the "maximal-dependency, maximal-relevance and maximal-significance" evaluation criteria mentioned above, called "OFS-A3M" as shown in Algorithm 2. The main goal of this online feature selection algorithm is to maximize $\mathbb{D}_S$ with the

Algorithm 2 OFS-A3M.

Require:
C: the condition feature set;
D: the decision classes;
Ensure:
S: the selected feature set
S: the selected feature set, initialized to {};
$\mathbb{D}_S$: the dependency of S to D, initialized to 0;
$Mean_{\mathbb{D}_S}$: the mean dependency of features in S, initialized to 0;
Repeat
Get a new feature f_i of C at time stamp t_i as C_{f_i};
Calculate the dependency of C_{f_i} as $\gamma_{f_i} = \mathbf{Dep-Adapted}(\mathbf{C_{f_i}}, \mathbf{GAP})$;
IF $\gamma_{f_i} < Mean_{\mathbb{D}_S}$
    discard feature f_i; and go to Step 23;
END IF
IF $\gamma_{S\cup f_i} > \mathbb{D}_S$
    $S = S \cup f_i$;
    $\mathbb{D}_S = \gamma_S$, $Mean_{\mathbb{D}_S} = \frac{1}{|S|}\sum_{f_i\in S}\gamma_{f_i}(D)$;
ELSE IF $\gamma_{S\cup f_i} == \mathbb{D}_S$
        $S = S \cup f_i$;
        random the feature order in S;
        FOREACH feature f' in S
            calculate f' 's significance as $\sigma_S(D, f')$;
            IF $\sigma_S(Y, f') == 0$
                $S = S\backslash\{f'\}$, $Mean_{\mathbb{D}_S} = \frac{1}{|S|}\sum_{f_i\in S}\gamma_{f_i}(D)$;
            END IF
        END FOR
END IF
Until no more features are available;
return S;

feature streams.

More specifically, if a new feature f_i arrives at time stamp t_i, Step 6 calculates the dependency of f_i using the dependency calculation method *Dep-Adapted*. The dependency function *Dep-Adapted* can be calculated on either only one feature or a feature set. All the following dependency computation calls from step 6 to step 18 use Algorithm 1 (*Dep-Adapted*) as their calculating method for dependency degree. Step 7 compares the dependency of f_i with the mean dependency of the selected feature set S. If γ_{f_i} is smaller than $Mean_{\mathbb{D}_S}$ and we add f_i into S, the $Mean_{\mathbb{D}_S}$ will decrease. Thus, with the "maximal-relevance" constraint, f_i is discarded when its dependency is smaller than $Mean_{\mathbb{D}_S}$.

If f_i satisfies the "maximal-relevance" constraint, step 10 compares the dependency of the 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 $\mathbb{D}_S$, which means adding new feature f_i will increase the dependency of the selected feature set, then we add f_i into S with the "maximal-dependency" constraint.

If the dependency of $S \cup f_i$ is equal to $\mathbb{D}_S$, we will use the "maximal-significance" constraint for the analysis of 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. (15). We will discard features whose significance equals to 0. By the "maximal-relevance", "maximal-dependency" and "maximal-significance" constraints, we can select features with high correlation, high dependency, and low redundancy.

Let us use the data in Table 1 to illustrate our algorithm for online streaming feature selection. First of all, we initialize $S = \{\}$, $\mathbb{D}_S = 0$ and $Mean_{\mathbb{D}_S} = 0$.

At time stamp t_1, we get the sample data of f_1 as $X_{f_1} = [3, 5, 8, 13, 6, 5, 9, 15]^T$. Then, we calculate the dependency of X_{f_1} as $\gamma_{f_1} = 0.2625$. For $\gamma_{S\cup f_1} = \gamma_{f_1} > \mathbb{D}_S = 0$, we add f_1 into S, update $\mathbb{D}_S = 0.2625$ and $Mean_{\mathbb{D}_S} = 0.2625$.

At time stamp t_2, we get the sample data of f_2 as X_{f_2}. We calculate the dependency of f_2 as $\gamma_{f_2} = 0.33542$. For $\gamma_{f_2} = 0.33542 > Mean_{\mathbb{D}_S} = 0.2625$ and $\gamma_{S\cup f_2} = \gamma_{f_1\cup f_2} = 0.32292 > \mathbb{D}_S = 0.2625$, we add f_2 into S and update $\mathbb{D}_S = 0.32292$ and $Mean_{\mathbb{D}_S} = 0.29896$.

For feature f_3 at time stamp t_3, $\gamma_{f_3} = 0.29464 < Mean_{\mathbb{D}_S} = 0.29896$. Thus, we discard f_3 and continue to process the next arriving feature.

At time stamp t_4, we get the new arriving feature f_4. For $\gamma_{f_4} = 0.6 > Mean_{\mathbb{D}_S} = 0.29896$ and $\gamma_{S\cup f_4} = \gamma_{f_1\cup f_2\cup f_4} = 0.55357 > \mathbb{D}_S = 0.32292$, we add f_4 into S and update the value of $\mathbb{D}_S$ and $Mean_{\mathbb{D}_S}$.

After time stamp t_4, there are no more features available. Thus, the final selected feature subset is $\{f_1, f_2, f_4\}$.

4.5. Time complexity of OFS-A3M

The time complexity of OFS-A3M mainly depends on the dependency function *Dep-Adapted*.

Suppose the data set is *DS*, the number of instances in *DS* is N and the number of features in *DS* is F. According to Section 4.4, the time complexity of *Dep-Adapted* is $O(N^2 logN)$. At time stamp t_i, a new feature f_i is present to the OFS-A3M algorithm. Steps 6–8 calculate the dependency of f_i and compare it with $Mean_{\mathbb{D}_S}$ (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 $Mean_{\mathbb{D}_S}$, f_i will be discarded. Otherwise, we calculate the dependency of $S \cup f_i$ and compare it with $\mathbb{D}_S$ (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 $\mathbb{D}_S$, we add f_i into S and go on to the next feature. If the dependency of $S \cup f_i$ is smaller than $\mathbb{D}_S$, f_i will be discarded. Only if the dependency of $S \cup f_i$ is equal to $\mathbb{D}_S$, we will calculate each features' significance and remove the redundant features from S. The time complexity of this phase is $O(|S| \times N^2 logN)$.

Thus, the time complexity of OFS-A3M in the worst case is $O(F^2 \times N^2 logN)$. For real-world datasets, it is impossible to choose all the features. Therefore, the time complexity will be much smaller than $O(F^2 \times N^2 logN)$ for real-world applications.

5. Experiment results

5.1. Experiment setup

In this section, we apply the proposed online feature selection algorithm on fifteen data sets, including three UCI datasets (WDBC, HILL VALLEY with noise, HILL VALLEY without noise), eleven DNA microarray datasets (PROSTATE-std, COLON, LYMPHOMA-std, DLBCL, LUNG CANCER-std, GLIOMA, SRBCT-std, LUNG2, LEUKEMIA-std, MLL, CAR) [35,38] and one NIPS 2003 dataset (ARCENE) [33] as shown in Table 2.

In our experiments, we use two basic classifiers, KNN(K=1) and SVM in Matlab R2015b to evaluate a selected feature subset. We perform 10-fold cross-validation on each data set. We use 9/10 data samples for training and the rest 1/10 data for testing. 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 8 GB memory.

To validate whether OFS-A3M and its rivals have a significant difference in prediction accuracy, number of selected features and running time, we conduct the Friedman test at a 95% significance level, under the null-hypothesis. The performance of OFS-A3M and that of its rivals has no significant difference if the null-hypothesis is accepted. If the null-hypothesis at the Friedman test is rejected, we proceed with the Nemenyi test [3] as a post-hoc test. With the Nemenyi test, the perfor-

Table 2
Experimental data sets.

Data set	Instances	Features	Classes
WDBC	569	30	2
HILL	606	100	2
HILL(noise)	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
CAR	174	9182	11
ARCENE	200	10,000	2
LUNG	181	12,533	2

Table 3
Predictive accuracy of gap neighborhood vs. δ neighborhood (KNN).

Data set	Gap	r=0.05	r=0.1	r=0.15	r=0.2	r=0.25	r=0.3	r=0.35	r=0.4	r=0.45	r=0.5
WDBC	0.9332	0.9139	0.9402	**0.942**	**0.942**	**0.942**	**0.942**	**0.942**	**0.942**	**0.942**	**0.942**
HILL	0.614	0.6091	0.6248	0.6339	**0.638**	0.6281	0.6331	0.614	0.6107	0.5901	0.6025
HILL(noise)	**0.5826**	0.5479	0.5529	0.5612	0.5686	0.5298	0.5223	0.5314	0.5149	0.5116	0.5298
COLON	0.75	0.5667	0.6667	0.8	0.8	**0.8167**	0.7	0.7333	0.7333	0.6667	0.7833
SRBCT	**0.9667**	0.2667	0.4833	0.7667	0.8	0.7667	0.8333	0.85	0.8667	0.8833	0.8333
LUNG2	**0.94**	0.8	0.83	0.85	0.83	0.86	0.865	0.87	0.895	0.865	0.865
LYM	0.9333	0.7833	0.95	0.9333	0.9333	**0.9667**	**0.9667**	**0.9667**	**0.9667**	**0.9667**	**0.9667**
GLIOMA	**0.68**	0.48	0.44	0.42	0.56	0.52	0.48	0.62	0.5	0.56	0.52
MLL	**0.9714**	0.3571	0.3714	0.4	0.3857	0.3714	0.4429	0.4286	0.4143	0.4429	0.4429
PRO	0.88	0.83	0.9	0.89	0.9	0.84	0.88	0.87	**0.91**	0.89	0.86
DLBCL	**0.9**	0.6725	0.7675	0.705	0.735	0.725	0.755	0.83	0.8	0.83	0.8125
LEU	0.9143	0.9	0.9	0.9143	0.8857	**0.9571**	0.9429	**0.9571**	**0.9571**	0.9	0.9
CAR	**0.7588**	0.3235	0.3824	0.3588	0.3824	0.4059	0.4235	0.4	0.3941	0.4059	0.3824
ARCENE	**0.81**	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46
LUNG	0.9611	0.9556	0.9611	0.9722	0.9833	**0.9889**	0.9722	0.9889	0.9778	0.9722	0.9778
AVERAGE	**0.8396**	0.6310	0.6820	0.7071	0.7202	0.7185	0.7212	0.7374	0.7295	0.7257	0.7252
RANKS	**4.0**	9.8333	7.6333	6.6	5.7333	5.5667	5.5333	4.4333	5.0333	5.7	5.9333

mance of 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 [3] for more details).

5.2. Gap neighborhood vs. δ neighborhood

In this section, we compare our new non-parametric neighborhood relation with δ neighborhood relation. For the sake of fairness, both neighborhood relations use OFS-A3M as the algorithm framework.

For δ neighborhood, we use $\delta = r \times D_{\max}$ and run the experiments varying with values of r from 0.05 to 0.5 with step 0.05. Tables 3 and 4 show the experimental results of our new neighborhood relation competing with δ neighborhood on these data sets. The p-values of Friedman test on KNN, SVM are $1.0820e-04$ and $9.2512e-07$ respectively. Table 5 shows the running time of different algorithms on these data sets and the p-value is 7.1537e-36. Table 6 shows the mean number of selected features on these data sets and the p-value is 1.4835e-20. Thus, there is a significant difference among these eleven compared algorithms on predictive accuracy, running time and number of selected features respectively. According to the Nemenyi test, the value of CD(critical difference) is 3.8963.

From Tables 3–6, we have the following observations:

- According to the average ranks and the value of critical difference, there is no significant difference between Gap neighborhood and δ neighborhood with KNN classifier except for $r = 0.05$. Meanwhile, there is no significant difference with SVM classifier except for $r = 0.05$ and 0.1.
- On some data sets, such as WDBC, LYMPHOMA, and LEU, δ neighborhood gets the highest predictive accuracy with different r values. This indicates that δ neighborhood can perform very well with a proper r. However, on data sets GLIOMA, CAR and ARCENE, all r values of δ neighborhood get a very low predictive accuracy of around 0.4. This demonstrates that δ neighborhood cannot handle some types of data sets with imbalanced data distribution. Thus, Gap neighborhood is superior to δ neighborhood and can handle different types of data well.

Table 4
Predictive accuracy of gap neighborhood vs. δ neighborhood (SVM).

Data set	Gap	r=0.05	r=0.1	r=0.15	r=0.2	r=0.25	r=0.3	r=0.35	r=0.4	r=0.45	r=0.5
WDBC	0.949	0.9438	0.9508	**0.956**	**0.956**	0.9543	0.9543	0.9543	0.9542	0.9543	0.9525
HILL	0.5298	0.5289	0.5529	0.5628	0.5678	0.5669	**0.5744**	0.5702	0.557	0.5587	0.5603
HILL(noise)	**0.5504**	0.514	0.5256	0.5207	0.5421	0.5339	0.5281	0.5223	0.5215	0.5182	0.5397
COLON	0.8333	0.65	0.7333	0.7833	**0.8667**	0.8167	0.7833	0.8167	0.8167	0.8333	0.8167
SRBCT	**0.9667**	0.4333	0.5667	0.75	0.7333	0.8	0.7833	0.9	0.9	0.9167	0.8
LUNG2	**0.915**	0.875	0.875	0.875	0.885	0.88	0.87	0.89	0.88	0.89	0.89
LYMPHOMA	**0.9667**	0.7667	0.9	**0.9667**	0.95	**0.9667**	**0.9667**	0.9333	0.9333	0.95	0.9333
GLIOMA	**0.68**	0.32	0.32	0.32	0.34	0.44	0.38	0.54	0.38	0.4	0.4
MLL	**0.9571**	0.3857	0.3714	0.4	0.3714	0.3714	0.4	0.4143	0.4	0.4143	0.4143
PROSTATE	0.91	0.88	**0.92**	0.88	0.88	0.88	0.88	0.89	0.87	**0.92**	0.9
DLBCL	**0.855**	0.7175	0.7375	0.73	0.7675	0.7675	0.7925	0.78	0.7925	0.8175	0.78
LEU	0.9429	0.9429	0.9143	0.9286	0.9286	0.9429	**0.9571**	**0.9571**	**0.9571**	0.9286	0.9286
CAR	**0.8294**	0.3529	0.3588	0.3765	0.3824	0.3765	0.3882	0.4118	0.3941	0.3529	0.4059
ARCENE	**0.815**	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
LUNG	0.9833	0.9778	0.9833	0.9889	0.9778	**0.9944**	0.9833	0.9833	0.9778	0.9833	0.9778
AVERAGE	**0.84557**	0.6565	0.6846	0.7065	0.7139	0.7234	0.7200	0.7415	0.7262	0.7331	0.7239
RANKS	**3.0667**	9.5667	8.3333	6.8	6.0667	5.4	5.4667	4.3	6.3667	5.0667	5.5667

Table 5
Running time of gap neighborhood vs. δ neighborhood (second).

Data set	Gap	r=0.05	r=0.1	r=0.15	r=0.2	r=0.25	r=0.3	r=0.35	r=0.4	r=0.45	r=0.5
WDBC	0.6984	0.4796	0.375	**0.317**	0.3266	0.3369	0.3434	0.3506	0.3586	0.3638	0.37
HILL	10.5129	**4.2196**	4.7587	5.3206	5.5765	5.8174	5.7853	5.5508	5.3789	5.4996	5.6071
HILL(noise)	10.361	**3.9377**	4.1746	4.2302	4.5198	4.5029	4.4465	4.4252	4.5575	4.5194	4.97
COLON	0.9768	**0.9136**	0.9559	0.9805	1.0051	1.0264	1.036	1.0451	1.054	1.0697	1.0881
SRBCT	1.093	**1.0129**	1.0448	1.0641	1.0794	1.0975	1.11	1.1199	1.128	1.141	1.1558
LUNG2	9.004	**5.1424**	5.3318	5.4855	5.6489	5.7491	5.9228	6.0483	6.0518	6.2897	6.3158
LYMPHOMA	2.1402	**1.7966**	1.8678	1.9056	1.9256	1.9501	1.9758	1.9939	2.0037	2.0278	2.0707
GLIOMA	1.5983	**1.596**	1.6674	1.6983	1.714	1.7277	1.7435	1.7546	1.768	1.7753	1.7871
MLL	3.1706	**2.9509**	3.0417	3.0786	3.1082	3.1421	3.1596	3.1856	3.2039	3.2149	3.2412
PROSTATE	5.1647	**4.2495**	4.4261	4.5053	4.6069	4.6773	4.71	4.7839	4.8182	4.8525	4.8881
DLBCL	4.4067	**3.7912**	3.9089	3.987	4.0353	4.0775	4.1226	4.1676	4.2121	4.2445	4.2663
LEU	4.2779	**3.5843**	3.7307	3.817	3.8459	3.8888	3.9255	3.9811	4.0249	4.0516	4.0881
CAR	17.8033	**12.4166**	12.7365	12.9179	13.0807	13.2331	13.3824	13.5159	13.6885	13.7538	13.8967
ARCENE	23.0345	**15.4814**	15.8433	16.0419	16.226	16.3903	16.5529	16.683	16.7851	16.8921	16.9897
LUNG	29.3758	**17.4496**	18.0579	18.4618	18.8473	19.1299	19.3466	19.6109	19.7916	19.9248	20.0877
AVERAGE	8.2412	**5.2681**	5.4614	5.5874	5.7030	5.7831	5.8375	5.8810	5.9216	5.9747	6.0548
RANKS	9.2	**1.6**	2.5333	3.0	4.4667	5.4667	6.2	6.8667	7.9333	8.7333	10.0

- On running time, there is a significant difference between Gap neighborhood and δ neighborhood when $r = 0.05, 0.1, 0.15, 0.2$ and there is no significant difference when $r = 0.25, 0.3, 0.35, 0.4, 0.45, 0.5$. δ neighborhood with $r = 0.05$ is the fastest. $r = 0.05$ is the smallest ratio of the radius and it considers fewer neighbors than other r values which makes it run faster. For Gap neighborhood, it sorts all the neighbors from the nearest to farthest. Meanwhile, δ neighborhood just considers the neighbors whose distance are less than a fixed radius. Thus, Gap neighborhood is a little slower than δ neighborhood.
- On the number of selected features, there is a significant difference between Gap neighborhood and δ neighborhood when $r = 0.05, 0.1, 0.15$. δ neighborhood with $r = 0.05$ selects the smallest mean number of features. On the whole, δ neighborhood selects more features with the increase of the r values. On data set ARCENE, all the different r values of δ neighborhood just select one feature. This indicates that for some data sets, δ neighborhood cannot specify any good parameters.

In sum, Gap neighborhood is superior to δ neighborhood on predictive accuracy and inferior to it on running time. Meanwhile, for δ neighborhood, it is difficult to select a uniform parameter for all different types of data sets.

5.3. Gap neighborhood vs. k-nearest neighborhood

In this section, we compare our new non-parametric neighborhood relation with k-nearest neighborhood relation. For the sake of fairness, both neighborhood relations use OFS−A3M as the algorithm framework.

For k-nearest neighborhood, we run the experiments varying with values of k from 1 to 10. Tables 7 and 8 show the experimental results of our new neighborhood relation competing with the k-nearest neighborhood on these data sets. The p-values of Friedman test on KNN and SVM are 0.0223 and 0.2319 respectively. Tables 9 and 10 show the running time and mean number of selected features on these data sets. The p-values are 9.9735e-07 and 1.5644e-19 respectively. Thus, there

Table 6
Gap neighborhood vs. δ neighborhood (the mean number of selected features).

Data set	Gap	r=0.05	r=0.1	r=0.15	r=0.2	r=0.25	r=0.3	r=0.35	r=0.4	r=0.45	r=0.5
WDBC	5.5	**3.4**	5.6	7.4	7.4	7.3	7.3	7.3	7.5	7.3	7.4
HILL	**7.2**	9.6	18.9	29.2	34	34.5	34.1	31.5	26.5	27.9	30.5
HILL(noise)	5.4	**4.8**	7.3	7.1	13.3	11.3	9.2	8.1	10.8	9.3	19.1
COLON	17.2	**2**	2.5	3.3	3.5	3.3	4.1	4.3	4.1	4.4	4.7
SRBCT	8.2	2	**1.6**	2.8	3.4	3.4	3.9	3.9	4	4.4	4.4
LUNG2	42.6	3	**2.9**	3.5	3.8	4.1	4.2	4.6	4.8	4.8	5.4
LYMPHOMA	4.4	**1.7**	2.1	2	2.3	2.1	2.8	2.5	2.7	2.2	2.5
GLIOMA	21.4	**1.7**	**1.7**	2	2.2	2.2	2.2	2.5	2.6	2.7	2.9
MLL	7.1	**1.2**	1.4	1.4	1.4	1.5	1.4	1.5	1.5	1.6	1.7
PROSTATE	24.6	**2.2**	2.8	3	3.1	3.8	4	4	4	4.7	4.5
DLBCL	22.2	**2**	2.1	2.5	2.8	2.9	3.1	3.7	3.9	4	3.4
LEU	24.5	**2**	2.2	2.6	2.6	2.8	2.7	2.8	2.5	3.2	3.2
CAR	41.6	**2.2**	2.3	2.8	2.8	3.3	3.4	3.6	3.9	4.1	4.3
ARCENE	33.2	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
LUNG	10.2	**2**	2.3	2.5	2.6	2.8	2.5	2.9	2.9	2.9	3.3
AVERAGE	18.3	**2.7**	3.7	4.8	5.7	5.7	5.7	5.6	5.5	5.6	6.5
RANKS	9.1333	1.5333	2.5333	3.9333	5.5	5.8667	6.2	6.9667	7.3667	7.9667	9.0

Table 7
Predictive accuracy of gap neighborhood vs. k-nearest neighborhood (KNN).

Data set	Gap	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
WDBC	**0.9474**	0.9473	0.9386	**0.9474**	0.9368	0.9368	0.9351	0.9403	0.9421	0.9421	0.9403
HILL	0.6107	0.5876	0.6339	0.6372	0.6579	0.6562	**0.6694**	0.6496	0.6455	0.6438	0.624
HILL(noise)	0.5496	0.576	**0.5769**	0.5694	0.5537	0.5636	0.5496	0.5182	0.5322	0.5446	0.557
COLON	0.75	0.7167	0.7667	0.7667	**0.8167**	**0.8167**	0.8	0.7833	0.8	0.8	**0.8167**
SRBCT	**1**	0.85	0.85	0.9	0.8833	0.8833	0.8833	0.8833	0.9167	**1**	**1**
LUNG2	0.9	0.875	0.9	0.905	0.925	0.92	**0.935**	0.925	0.925	0.92	0.925
LYMPHOMA	0.9333	0.95	0.85	**0.95**	0.9833	0.8833	0.9333	0.9333	0.9167	0.9333	0.9333
GLIOMA	0.74	0.6	0.68	0.7	0.6	0.68	0.72	0.66	**0.84**	0.68	0.78
MLL	0.9429	0.8571	0.9286	0.9429	0.9286	0.9286	0.9571	0.9714	0.9429	0.9429	**0.9714**
PROSTATE	0.84	0.83	0.87	0.86	0.89	0.88	0.88	0.91	**0.92**	0.9	0.9
DLBCL	0.8875	0.7625	0.875	0.85	0.8375	0.9	0.8625	0.875	0.8875	**0.9125**	0.8875
LEU	0.8857	0.8714	0.8571	0.9143	0.9286	**0.9429**	**0.9429**	0.9286	0.9286	**0.9429**	0.8857
CAR	0.7471	0.7588	0.8176	0.8412	**0.8529**	0.8412	0.8176	0.8353	0.8294	0.8176	0.8118
ARCENE	0.75	0.805	0.835	0.83	0.83	0.785	0.825	0.83	0.835	0.815	**0.85**
LUNG	0.9889	0.9833	0.9833	0.9722	0.9833	0.9889	0.9833	0.9722	0.9889	**0.9944**	0.9889
AVERAGE	0.8315	0.7980	0.8241	0.8390	0.8405	0.8404	0.8462	0.8410	0.8567	0.8526	**0.8581**
RANKS	6.8667	8.7333	7.6667	5.9333	5.6333	5.6667	5.6000	5.9333	4.5667	4.9333	**4.4667**

is a significant difference among these eleven compared algorithms on predictive accuracy(KNN), running time and number of selected features respectively. Meanwhile, there is no significant difference on predictive accuracy with SVM. According to the Nemenyi test, the value of CD(critical difference) is 3.8963.

From Tables 7–10, we have the following observations:

- According to the average ranks and the value of critical difference, there is no significant difference between Gap neighborhood and k-nearest neighborhood with both KNN and SVM classifiers for all different k values.
- On some data sets, such as GLIOMA, SRBCT, and ARCENE, the k-nearest neighborhood can get the highest predictive accuracy with a proper k. This indicates that k-nearest neighborhood can perform very well with a proper parameter k. However, the k value which can get the highest performance for one data set may be not good for another data set. This demonstrates that it is not easy to select a uniform parameter k for all types of data sets.
- On running time, there is a significant difference between Gap neighborhood and the k-nearest neighborhood except for $k = 8$. In general, the k-nearest neighborhood is faster than Gap neighborhood. The main reason for this is Gap neighborhood sort all the neighbors from nearest to farthest and check an uncertain number of neighbors while k-nearest neighborhood checks a fixed number of neighbors.
- On the number of selected features, there is no significant difference between Gap neighborhood and k-nearest neighborhood, except for $k = 1$. K-nearest neighborhood with $k = 1$ selects the smallest mean number of features. On the whole, k-nearest neighborhood selects more features with the increase of the k values.

In sum, Gap neighborhood is comparable to the k-nearest neighborhood with different k values on predictive accuracy and a little slower on running time. However, it is still a challenge for k-nearest to specify a uniform parameter k for all datasets. Thus, it is an outstanding advantage for Gap neighborhood which does not need to specify any parameters in advance.

Table 8
Predictive accuracy of gap neighborhood vs. *k*-nearest neighborhood (SVM).

Data set	Gap	*k*=1	*k*=2	*k*=3	*k*=4	*k*=5	*k*=6	*k*=7	*k*=8	*k*=9	*k*=10
WDBC	0.9596	0.9596	0.9543	**0.9613**	0.9579	0.9578	0.9596	0.9561	0.9561	**0.9613**	0.9578
HILL	0.5314	0.5281	0.5421	0.5554	0.5537	**0.5603**	0.557	0.5579	0.5521	0.5504	0.5438
HILL(noise)	0.514	**0.5298**	0.5223	0.5248	0.5099	0.5091	0.5091	0.5074	0.5058	0.4942	0.486
COLON	0.8	0.7667	0.8	0.7667	0.7833	0.7167	0.8	0.7667	**0.8667**	0.75	0.7333
SRBCT	0.9833	0.85	0.8667	0.8833	0.8667	0.8667	0.8667	0.85	0.9	0.9667	**1**
LUNG2	0.92	0.91	0.89	0.91	**0.94**	0.935	0.92	0.89	0.9	0.91	0.935
LYMPHOMA	0.9333	0.9	0.8333	0.9667	**0.9833**	0.9333	0.9167	0.9333	0.9333	0.9167	0.9333
GLIOMA	0.72	0.58	0.74	0.7	0.72	0.68	**0.78**	0.6	0.72	0.66	0.7
MLL	0.9286	0.8857	0.9143	0.9286	0.9429	0.9286	0.9143	0.9286	0.9143	0.9429	**0.9714**
PROSTATE	0.82	0.86	0.86	0.88	0.88	0.87	0.85	0.87	0.89	**0.91**	0.87
DLBCL	0.875	0.8375	0.9125	0.8875	0.8625	0.8	0.9	0.8875	0.9	**0.925**	0.9125
LEU	0.8714	0.9286	0.8857	0.9286	0.9143	0.9286	**0.9714**	0.9429	0.9286	**0.9714**	0.9
CAR	0.7706	0.7647	0.8294	0.8176	0.8118	**0.8588**	0.8412	0.8412	0.8471	0.8294	0.8529
ARCENE	0.75	0.75	0.78	0.76	0.795	0.78	0.785	0.83	0.79	0.8	**0.815**
LUNG	0.9889	0.9889	0.9778	0.9722	0.9889	0.9833	0.9889	0.9778	**0.9944**	**0.9944**	**0.9944**
AVERAGE	0.8244	0.8026	0.8205	0.8295	0.8340	0.8205	0.8373	0.8226	0.8398	0.8388	**0.8403**
RANKS	6.5333	8.2333	7.3667	5.6000	5.0667	6.2333	5.2667	6.5667	5.2333	**4.8333**	5.0667

Table 9
Running time of gap neighborhood vs. *k*-nearest neighborhood (second).

Data set	Gap	*k*=1	*k*=2	*k*=3	*k*=4	*k*=5	*k*=6	*k*=7	*k*=8	*k*=9	*k*=10
WDBC	0.6899	0.6063	0.6073	0.5955	0.6006	0.5734	0.5731	**0.5668**x	0.5879	0.5729	0.5729
HILL	10.1766	**7.2154**	7.9743	8.1567	8.7216	8.5508	9.4345	9.8006	10.503	9.5903	9.2195
HILL(noise)	10.1322	9.3604	8.8078	8.4224	7.912	7.8314	7.6347	7.6439	7.7622	7.4514	**7.3396**
COLON	0.9872	0.9354	0.9546	0.9422	0.9364	0.9069	0.8772	0.8777	0.8806	**0.857**	0.8727
SRBCT	1.0727	**0.8914**	0.893	0.8951	0.8957	0.9021	0.9042	0.9078	0.9156	0.9264	0.9261
LUNG2	9.5273	9.0099	8.7382	9.3881	8.3438	8.9094	8.5721	8.576	8.535	8.0184	**7.9948**
LYMPHOMA	2.1232	**1.6433**	1.7905	1.925	1.9071	2.0633	2.2135	2.2145	2.1162	2.3258	2.2252
GLIOMA	1.6239	**1.4353**	1.4432	1.4918	1.5052	1.5356	1.5045	1.5549	1.5813	1.5877	1.542
MLL	3.1765	**2.6611**	2.7075	2.7643	2.7597	2.8109	2.8704	2.8721	2.8951	2.8962	2.9242
PROSTATE	5.1129	**4.3171**	4.3514	4.3227	4.4209	4.4356	4.4167	4.4214	4.5207	4.4788	4.5703
DLBCL	4.4048	3.6622	**3.6575**	3.7036	3.7781	3.8455	3.7975	3.7746	3.7346	3.7627	3.8617
LEU	4.1749	**3.5339**	3.5496	3.7053	3.5895	3.6599	3.73	3.754	3.6499	3.6744	3.6665
CAR	17.6736	16.2936	15.3579	**14.9634**	15.1571	15.1638	15.055	15.1574	15.2149	14.9653	15.1031
ARCENE	23.1423	**18.4634**	18.8765	19.0717	19.329	19.2928	19.1178	19.0015	19.2361	18.6928	19.0262
LUNG	29.2766	**22.2608**	22.4768	22.6138	22.5449	22.7162	22.833	22.7999	22.8687	22.9523	23.084
AVERAGE	8.2196	6.8193	**6.8124**	6.8641	6.8267	6.8798	6.9022	6.9282	7.0001	6.8501	6.8619
RANKS	10.6667	**3.7333**	4.4	5.0	5.2667	6.1333	5.6	6.2	6.8667	5.9	6.2333

5.4. The influence of feature stream order

In this section, we 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 with our new algorithm OFS−A3M.

Figs. 4 and 5 show the experimental results of our new algorithm with three different feature stream orders on these data sets (the data sets from 1 to 15 are WDBC, HILL, HILL (noise), COLON, SRBCT, LUNG2, LYMPHOMA, GLIOMA, MLL, PROSTATE, DLBCL, LEU, CAR, ARCENE, and LUNG). Figs. 6 and 7 show the number of selected features and running time on these data sets.

In order to validate whether these three types of feature stream orders have a significant difference in predictive accuracy, running time and number of selected features, we conduct the Friedman test at a 95% significance level under the null hypothesis. The *p*-values of the original vs. inverse and random orders are given in Table 11.

From Table 11, we can observe that there is no significant difference between these three types of feature stream orders. From Figs. 4–7, we can see that there are minor fluctuations on some data sets, and on most of these data sets, these three orders basically overlap each other. This indicates that the feature stream order has little influence on our new neighborhood relation and new online streaming feature selection algorithm.

5.5. OFS-A3M vs. traditional feature selection methods

In this section, we compare OFS-A3M with seven representative traditional feature selection methods, including Fisher [6], ReliefF [22], PCC (Pearson Correlation Coefficient) [32], Laplacian Score [8], L0 [7], MI (mutual information) [27] and FSV [2].

Table 10
Gap neighborhood vs. *k*-nearest neighborhood (the mean number of selected features).

Data set	Gap	***k*=1**	***k*=2**	***k*=3**	***k*=4**	***k*=5**	***k*=6**	***k*=7**	***k*=8**	***k*=9**	***k*=10**
WDBC	5.8	**4.9**	6	7.1	6.5	6.9	7	6.6	6.7	7	6.9
HILL	6.1	**4.9**	7.6	13.3	12.6	14.3	17.2	17.1	13.2	12.2	10.4
HILL(noise)	5.7	11.9	9.4	7.7	4.8	5.5	4.5	4.2	4	3.1	**2.7**
COLON	15.8	**9.2**	11.7	17.5	14.8	18.6	21.8	22.1	15.5	20.6	23.9
SRBCT	8.9	4.9	3.6	4.4	**3.3**	4.5	7.3	8	9.7	13.6	16.1
LUNG2	38.2	**13.1**	17.9	26	22.2	35.7	42.5	40.2	29.4	41.6	47.2
LYMPHOMA	5.2	**3.2**	3.5	4.6	3.9	4.9	6.1	5.8	6.4	6.5	6.5
GLIOMA	19.6	**8.9**	11.8	18.3	16.4	25.7	25.8	24.8	22	26.6	31.7
MLL	9.6	**6.1**	6.9	8.2	9.5	10.8	11	13.3	10.6	13.9	16.5
PROSTATE	29.3	**9.5**	10.9	16.2	15.6	23.6	27.3	29.5	24.5	33.2	37.1
DLBCL	18.8	**7.7**	10	14.5	12.1	17.9	19.7	20.2	14.3	21.2	23.7
LEU	28.9	**5.5**	6.3	9.9	9.6	13.5	14.2	18.8	17	25.3	24.7
CAR	42.4	**25.1**	26.2	32.5	31.9	37.6	37.5	39.5	36.1	40.3	38.4
ARCENE	27.7	**15.2**	19.2	28.1	24.5	33.9	34.6	36.2	30.5	35.1	36.4
LUNG	16.6	**3.4**	4.5	5.8	6.8	7.2	9.2	9.4	11.7	13	14.8
AVERAGE	18.5	**8.9**	10.3	14.2	12.9	17.3	19.0	19.7	16.7	20.8	22.4
RANKS	6.6667	1.9333	2.6667	5.1333	3.4667	6.3667	7.7667	7.9333	6.0667	8.8	9.2

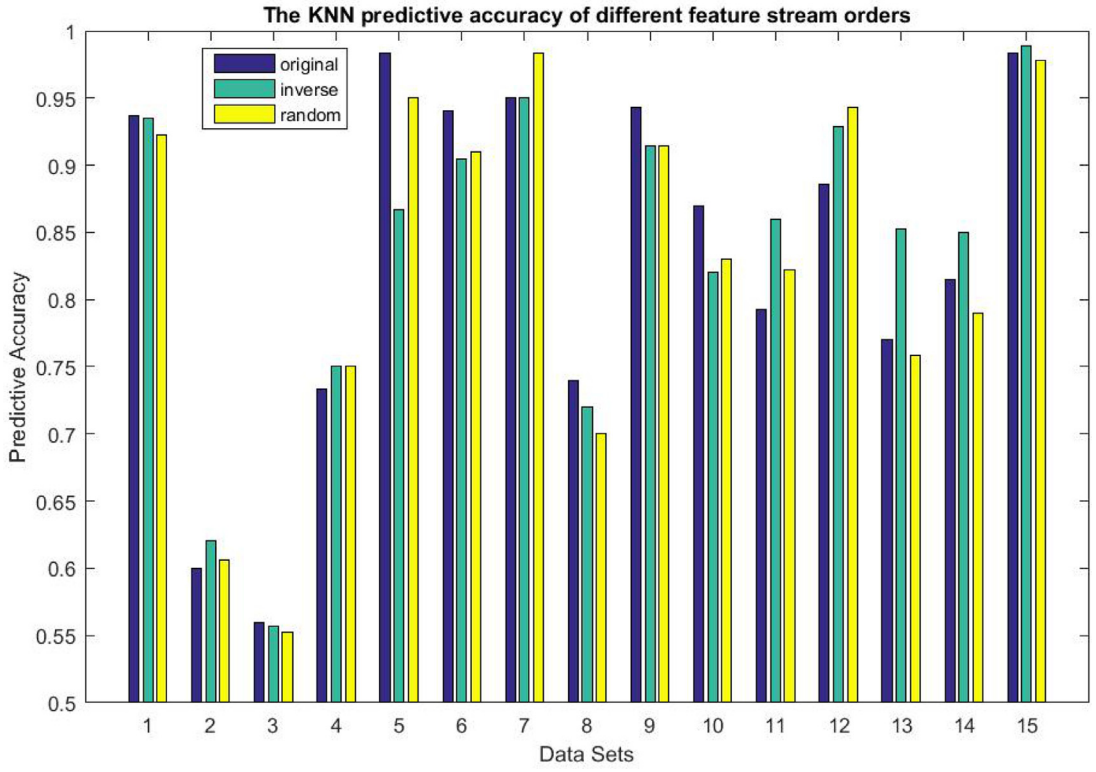

Fig. 4. Predictive accuracy of three different feature stream orders (KNN).

Table 11
The *p*-values of original vs. inverse and random.

	Original	Inverse	Random
KNN classifer	–	1.0000	0.1967
SVM classifer	–	0.5930	0.7815
Running time	–	0.1967	0.1967
Number of selected features	–	0.1967	1.0000

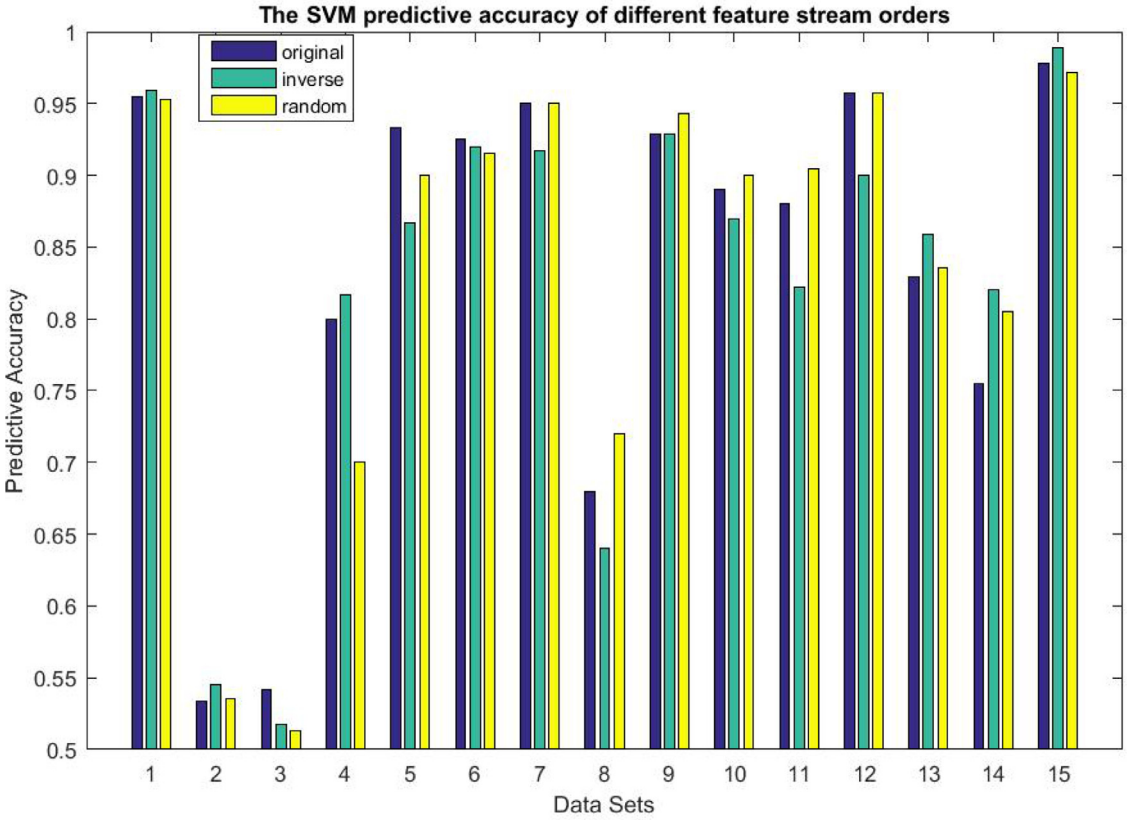

Fig. 5. Predictive accuracy of three different feature stream orders (SVM).

Table 12
Predictive accuracy using the KNN classifier.

Data set	OFS-A3M	Fisher	PCC	ReliefF	MI	Laplacian	L0	FSV
WDBC	**0.9366**	0.9332	0.9332	0.8929	0.9314	0.9297	0.9322	0.8981
HILL	**0.5992**	0.5372	0.5372	0.514	0.5545	0.5058	0.5603	0.5512
HILL(noise)	**0.5529**	0.4893	0.4893	0.5322	0.5041	0.4983	0.505	0.4967
COLON	**0.8**	**0.8**	0.7833	0.7667	0.7667	0.5	0.7333	0.6167
SRBCT	0.95	**0.9667**	0.85	0.8833	0.95	0.3833	0.6833	0.6833
LUNG2	**0.915**	0.875	0.9	0.895	0.89	0.86	0.775	0.885
LYMPHOMA	**0.95**	0.85	0.85	0.7833	**0.95**	0.8167	0.5833	0.5833
GLIOMA	**0.74**	0.6	0.72	0.44	0.56	0.5	0.52	0.6
MLL	**0.9286**	0.8286	0.7714	0.8857	0.8571	0.9	0.6143	0.8714
PROSTATE	**0.91**	0.87	0.87	0.9	0.9	0.58	0.89	0.63
DLBCL	**0.95**	0.885	0.885	0.905	0.7375	0.855	0.7475	0.71
LEU	0.9	0.9143	0.9143	0.9429	**0.9571**	0.6714	0.9	0.6571
CAR	0.8059	0.7	0.7412	**0.8882**	0.8471	0.7824	0.5118	0.6824
ARCENE	0.8	0.66	0.465	0.56	0.73	0.69	**0.83**	0.71
LUNG	**0.9944**	0.9889	0.9889	0.9889	0.9889	0.85	0.9722	0.8444
AVERAGE	**0.8869**	0.8282	0.8115	0.8199	0.8445	0.6990	0.7300	0.7061
RANKS	**1.6667**	4.3333	4.4667	4.2333	3.5667	6.1333	5.4333	6.1667

All these algorithms are implemented in MATLAB. The K value of ReliefF is set to 5 for the best performance. None of these seven traditional feature selection methods can handle the scenario of feature streaming in an online manner. Thus, we rank all features from high to low and select the same number of features as OFS-A3M. We evaluate OFS-A3M and all competing ones on the predictive accuracy with 10-fold cross-validation.

Tables 12 and 13 summarize the predictive accuracy of OFS-A3M against the other seven competing algorithms using the basic classifiers of KNN and SVM. The p-values on predictive accuracy with KNN and SVM are 2.1935e – 07 and 1.0765e – 06 respectively. Thus, there is a significant difference among these eight compared algorithms on predictive accuracy with both KNN and SVM. According to the Nemenyi test, the value of CD(critical difference) is 2.7132.

From Tables 12 and 13, we have the following observations.

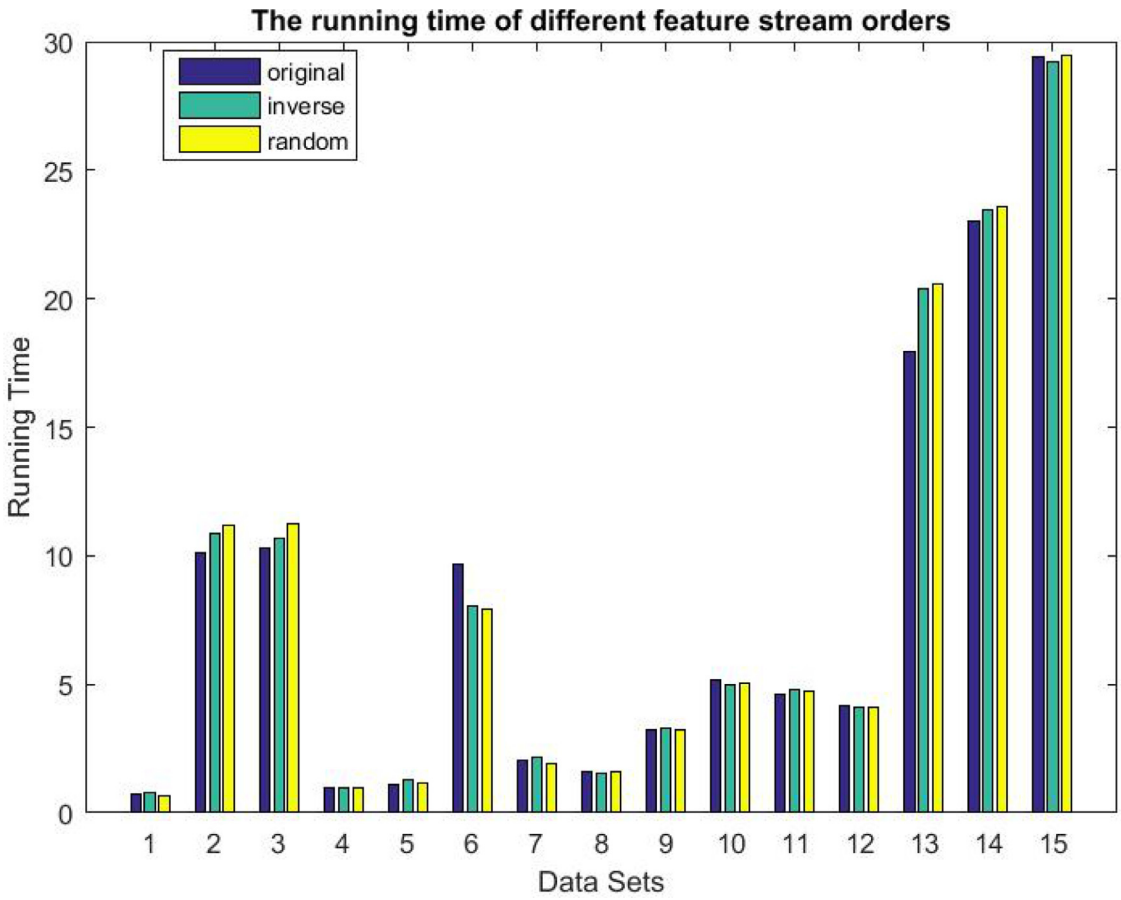

Fig. 6. Running time of three different feature stream orders (second).

Table 13
Predictive accuracy using SVM as the base classifier.

Data set	OFS-A3M	Fisher	PCC	ReliefF	MI	Laplacian	L0	FSV
WDBC	0.9525	**0.9578**	**0.9578**	0.942	0.9279	0.9331	0.9344	0.9279
HILL	**0.5281**	0.5099	0.5099	0.5116	0.5116	0.5083	0.514	0.5116
HILL(noise)	**0.5157**	0.4843	0.4843	0.495	0.4909	0.4835	0.4917	0.4909
COLON	**0.8167**	0.7667	0.7667	0.75	0.7833	0.6333	0.6667	0.7
SRBCT	**0.9667**	0.9333	0.85	0.9167	0.95	0.4833	0.7333	0.7333
LUNG2	0.92	0.85	0.895	**0.925**	0.89	0.845	0.845	0.885
LYMPHOMA	0.9	0.8833	0.8667	0.7333	**0.9333**	0.8667	0.6333	0.6333
GLIOMA	**0.8**	0.52	0.6	0.42	0.58	0.44	0.68	0.54
MLL	**0.9286**	0.8429	0.8571	0.9	0.9143	0.9143	0.6571	0.8857
PROSTATE	0.88	0.87	0.87	0.9	**0.92**	0.67	0.9	0.72
DLBCL	**0.95**	0.88	0.8675	0.8675	0.8375	0.9	0.76	0.735
LEU	0.9286	0.9286	0.9286	**0.9429**	0.9286	0.7143	0.9	0.6429
CAR	0.8118	0.7059	0.7588	**0.8824**	0.8647	0.7765	0.7	0.7059
ARCENE	0.745	0.745	0.565	0.66	0.73	0.59	**0.83**	0.775
LUNG	**0.9889**	0.9833	0.9833	**0.9889**	0.9833	0.8722	0.9778	0.8889
AVERAGE	**0.8863**	0.8257	0.8173	0.8238	0.8595	0.7254	0.7736	0.7370
RANKS	**1.9667**	4.6	4.6667	3.6333	3.5333	6.3667	5.2667	5.9667

- OFS – A3M vs. Fisher. According to the average ranks and the value of critical difference, there is no significant difference between OFS – A3M and Fisher on predictive accuracy at a 95% significance level under the null hypothesis. OFS – A3M outperforms Fisher on twelve of the fifteen datasets in both cases with KNN and SVM. Fisher computes a score for each feature as the ratio of inter-class separation and intra-class variance. It measures the features independently, and it cannot consider the information of the selected feature set as a whole.

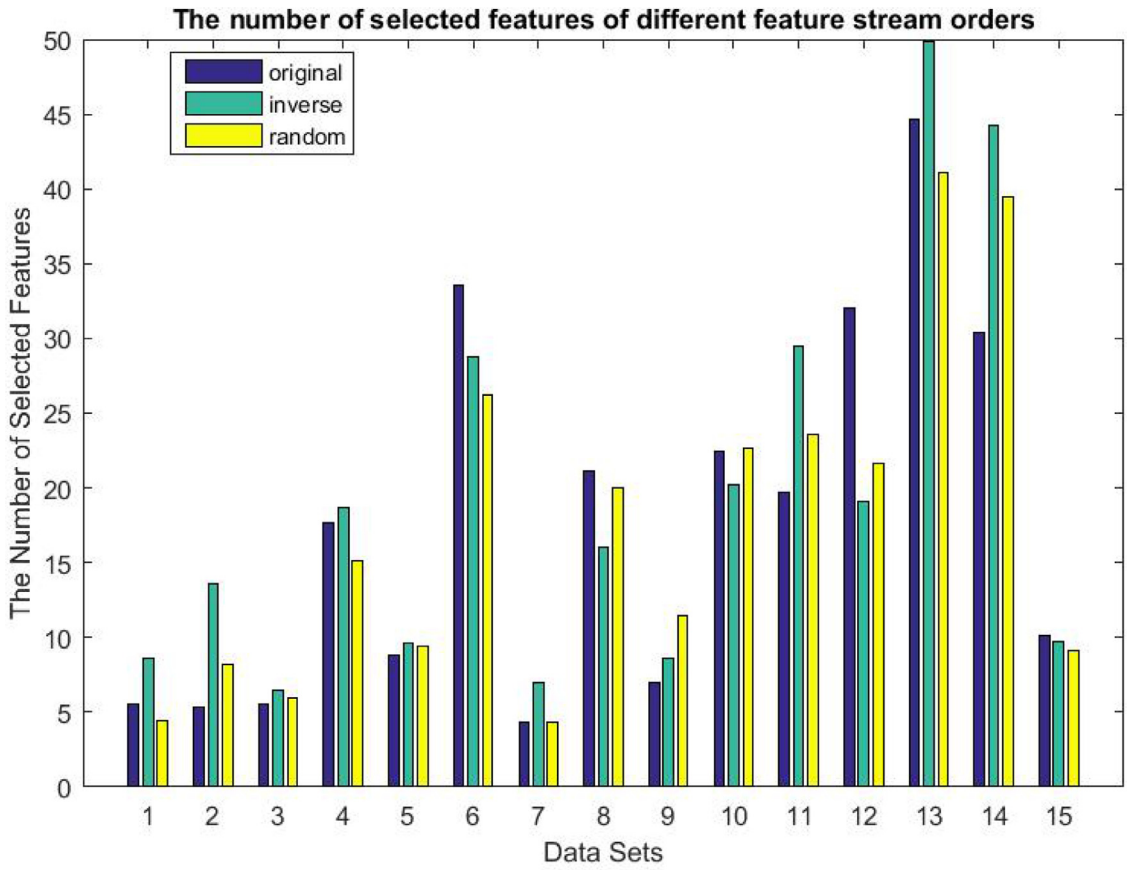

Fig. 7. The mean number of selected features in three different feature stream orders.

- OFS-A3M vs. PCC. There is no significant difference between OFS-A3M and PCC on predictive accuracy. OFS-A3M outperforms PCC on fourteen of the fifteen datasets in both cases. PCC gets the predictive accuracy about 46% and 56% on dataset ARCENE with KNN and SVM, while OFS-A3M gets the predictive accuracy of 80% and 74% respectively. OFS-A3M is higher PCC over 20% on the predictive accuracy of this dataset. In general, PCC cannot handle some datasets well.
- OFS-A3M vs. ReliefF. ReliefF is similar to OFS-A3M because they both use the neighbors' information for feature selection. There is no significant difference between OFS-A3M and ReliefF on predictive accuracy with KNN and SVM. OFS-A3M gets a higher predictive accuracy than ReliefF on eleven of the fifteen datasets. ReliefF estimates the quality of the features according to how well their values differentiate data samples that are near to each other. ReliefF does not discriminate among redundant features, and performance decreases with few data.
- OFS-A3M vs. MI. There is no significant difference between OFS-A3M and MI with both KNN and SVM. OFS-A3M outperforms MI on ten of the fifteen datasets at least. Especially on data set GLIOMA, OFS-A3M over MI 20% on predictive accuracy with both KNN and SVM. MI considers the mutual information between the distribution of the values of a given feature and the membership to a particular class, and the features are evaluated independently.
- OFS-A3M vs. Laplacian Score. Laplacian Score gets the lowest mean predictive accuracy among these competing algorithms. There is a significant difference between OFS-A3M and Laplacian Score on predictive accuracy with both KNN and SVM. OFS-A3M outperforms Laplacian Score on all of these datasets. Laplacian Score is an unsupervised feature selection algorithm which does not use the class information of each instance, and the importance of a feature is evaluated by its power of locality preserving.
- OFS-A3M vs. L0. L0 is an embedded feature selection method. There is a significant difference between OFS-A3M and L0 on predictive accuracy with both KNN and SVM. OFS-A3M outperforms L0 on thirteen of the fifteen data sets at least. On data sets SRBCT, LYMPHOMA and MLL, OFS-A3M over L0 almost 30% on predictive accuracy with both KNN and SVM. This indicates that L0 cannot handle some types of datasets well.
- OFS-A3M vs. FSV. FSV is a wrapper method, where the feature selection process is injected into the training of an SVM by a linear programming technique. There is a significant difference between OFS-A3M and FSV on predictive accuracy. Meanwhile, FSV cannot handle some types of datasets well.

Table 14
Predictive accuracy using KNN as the base classifier.

Data set	OFS-A3M	Alpha-investing	OSFS	Fast-OSFS	SAOLA
WDBC	0.9367	0.9561	**0.9614**	**0.9614**	0.9122
HILL	0.6140	**0.6455**	0	0	0
HILL(noise)	0.5653	**0.5686**	0	0	0
COLON	**0.7833**	0.4667	0.7333	0.7667	0.7167
SRBCT	**0.9667**	0.5667	0.85	0.75	0.7167
LUNG2	**0.92**	0.875	0.79	0.865	0.89
LYMPHOMA	**0.95**	0.7167	0.9167	0.9167	**0.95**
GLIOMA	**0.74**	0.46	0.58	0.6	0.5
MLL	**0.9429**	0.9286	0.7714	0.8714	**0.9429**
PROSTATE	**0.88**	0.85	0.84	0.86	0.86
DLBCL	**0.9125**	0.805	0.8875	0.8425	0.8875
LEU	**0.9429**	0.6143	0.8857	0.9143	0.8857
CAR	**0.7588**	0.7176	0.4588	0.6471	0.7529
ARCENE	**0.785**	0.71	0.62	0.69	0.63
LUNG	0.9778	0.8222	0.9778	**0.9833**	**0.9833**
AVERAGE	**0.845**	0.7135	0.6848	0.7112	0.7085
RANKS	**1.5667**	3.6667	3.7	3.0	3.0667

Table 15
Predictive accuracy using SVM as the base classifier.

Data set	OFS-A3M	Alpha-investing	OSFS	Fast-OSFS	SAOLA
WDBC	0.9543	**0.9737**	0.9649	0.9614	0.9192
HILL	0.5339	**0.5628**	0	0	0
HILL(noise)	**0.5355**	0.5273	0	0	0
COLON	**0.8333**	0.6333	0.7333	0.7833	0.8167
SRBCT	**0.95**	0.3667	0.8	0.7667	0.8
LUNG2	**0.935**	**0.935**	0.875	0.87	0.885
LYMPHOMA	0.9333	0.7333	0.9	0.8833	**0.95**
GLIOMA	**0.78**	0.44	0.54	0.58	0.66
MLL	0.9143	**0.9571**	0.8429	0.8714	0.8857
PROSTATE	0.85	0.88	**0.89**	**0.89**	**0.89**
DLBCL	**0.9175**	0.8425	0.85	0.88	0.8925
LEU	**0.9429**	0.7	0.9143	0.9286	0.8857
CAR	**0.8294**	0.6353	0.5059	0.6824	0.7882
ARCENE	**0.805**	0.755	0.625	0.64	0.655
LUNG	0.9833	0.8833	0.9778	0.9833	**0.9889**
AVERAGE	**0.8465**	0.7216	0.6946	0.7146	0.7344
RANKS	**1.8**	3.4333	3.7	3.3667	2.7

In sum, OFS-A3M provides best overall performance on these datasets and gets the highest mean predictive accuracy with both KNN and SVM.

5.6. OFS-A3M vs. online streaming feature selection methods

In this section, we compare our algorithm with four state-of-the-art online feature selection methods: Alpha-investing [40], OSFS [33], Fast-OSFS [33], SAOLA [37].

All aforementioned algorithms are implemented in MATLAB [36]. The significance level α is set to 0.01 for OSFS, Fast-OSFS, and SAOLA. For Alpha-investing, the parameters are set to the values used in [40].

Tables 14 and 15 summarize the predictive accuracy of OFS-A3M against the other four algorithms using the KNN and SVM classifiers. The p-values of Friedman test on KNN and SVM are 5.7467e-04 and 0.0059 respectively. Tables 16 and 17 show the running time and the mean number of selected features of OFS-A3M against other four algorithms. The p-values are 2.5014e-21 and 1.1009e-07 respectively. Thus, there is a significant difference among these five compared algorithms on predictive accuracy, running time and number of selected features. According to the Nemenyi test, the value of CD(critical difference) is 1.5758.

From Tables 14–17, we have the following observations.

- OFS-A3M vs. Alpha-investing. Alpha-investing is the fastest algorithm among all these five compared methods. According to the average ranks and the value of critical difference, there is a significant difference between OFS-A3M and Alpha-investing on predictive accuracy with KNN and SVM. From Tables 14 and 15, we can see that OFS-A3M outperforms Alpha-investing on eleven of the fifteen data sets at least with both KNN and SVM. Meanwhile, we can see that the features selected by Alpha-investing cannot fit for some data sets. For example, Alpha-investing only gets the predictive

Table 16
Running time (seconds).

Data set	OFS-A3M	Alpha-investing	OSFS	Fast-OSFS	SAOLA
WDBC	0.7157	**0.0034**	0.1193	0.0546	0.0137
HILL	10.6571	**0.0097**	0.0154	0.0151	0.0154
HILL(noise)	10.8333	**0.0098**	0.0153	0.0151	0.0155
COLON	1.0537	**0.0697**	0.4393	0.3098	0.3286
SRBCT	1.0922	**0.081**	1.834	0.5049	0.8169
LUNG2	9.1273	**0.6148**	73.2368	3.1801	2.1788
LYMPHOMA	2.2202	**0.1958**	10.2772	2.0675	4.4469
GLIOMA	1.6099	**0.2234**	5.1661	1.4063	2.2471
MLL	3.4648	**0.4317**	11.7961	2	4.8164
PROSTATE	5.1458	**0.3403**	2.4835	1.121	1.4128
DLBCL	4.412	**0.4196**	3.3888	1.3873	1.7952
LEU	4.2766	**0.4177**	4.1138	1.3986	1.9001
CAR	18.31	**1.557**	77.7738	4.1396	3.7589
ARCENE	23.0144	**0.9647**	9.2767	2.1529	3.334
LUNG	29.3495	**1.0958**	80.7153	5.247	8.197
AVERAGE	8.3521	**0.4289**	18.71	1.6666	2.3518
RANKS	4.3333	1.0	4.3667	2.2000	3.1

Table 17
The mean number of selected features.

Data set	OFS-A3M	Alpha-investing	OSFS	Fast-OSFS	SAOLA
WDBC	5.7	19.3	3.1	4	**2**
HILL	6.2	9	**0**	**0**	**0**
HILL(noise)	5.4	7	**0**	**0**	**0**
COLON	14.6	**1**	2	2.5	3.9
SRBCT	9.7	**1**	2.8	4.7	19.5
LUNG2	43.7	39.4	**6.5**	9.5	28.2
LYMPHOMA	5.2	3.2	**3.1**	6.1	37.6
GLIOMA	21.5	2.7	**1.7**	4.2	15.2
MLL	7.1	9.9	**2.6**	5.2	31
PROSTATE	27.9	2.1	**1.5**	3.6	12.2
DLBCL	19.8	**2.2**	2.3	4.2	12.9
LEU	22.1	**2.1**	2.5	4.8	20.9
CAR	40.7	27.8	**5.6**	9.4	39.7
ARCENE	25.8	9.1	**3.1**	5.9	19.6
LUNG	11.2	**2.6**	4.3	7.7	52.9
AVERAGE	17.7	9.2	**2.7**	4.7	19.7
RANKS	4.4000	2.6667	1.5333	2.6667	3.7333

accuracy of around 0.4 and 0.5 on data sets GLIOMA and SRBCT in cases of KNN and SVM respectively. The reason is that these data sets are very sparse and Alpha-investing can only select the first few features of these data sets.

- OFS-A3M vs. OSFS. There is a significant difference between OFS-A3M and OSFS on predictive accuracy in cases of KNN and SVM respectively. OFS-A3M outperforms OSFS on twelve of the fifteen datasets with both KNN and SVM. On data sets HILL and HILL with noise, OSFS cannot select any features and gets the prediction accuracy 0. On dataset GLIOMA, OSFS only gets the predictive accuracy around 0.5 in cases of KNN and SVM respectively, while OFS-A3M gets the predictive accuracy above 0.7. In addition, OFS-A3M is faster than OSFS on running time. OSFS selects the fewest number of features, among all these five compared methods. Thus, some important information is missing which causes the lower predictive accuracy.
- OFS-A3M vs. Fast-OSFS. There is no significant difference between OFS-A3M and Fast-OSFS on predictive with KNN and SVM respectively. OFS-A3M performs better than Fast-OSFS on thirteen of the fifteen datasets. Fast-OSFS is faster than OFS-A3M. Nevertheless, similar to OSFS, Fast-OSFS selects very few features on data sets, which leads to the missing of some important information.
- OFS-A3M vs. SAOLA. There is no significant difference between OFS-A3M and SAOLA on predictive accuracy. SAOLA is faster than OFS-A3M. However, OFS-A3M outperforms SAOLA on twelve of the fifteen datasets with both KNN and SVM. On the dataset HILL and HILL with noise, SAOLA cannot select any features and get the predictive accuracy 0. This demonstrates that SAOLA cannot handle some types of data well and can not select any features on these datasets.

In sum, our algorithm OFS-A3M is not faster than some competing algorithms of Alpha-investing, Fast-OSFS, and SAOLA, but it outperforms all competing algorithms in predictive accuracy on these datasets.

6. Conclusion

Most of existing online streaming feature selection methods need domain information before learning and specifying some parameters in advance. It is hence a challenge to specify unified and optimal values of parameters for all different types of data sets.

In this paper, we defined a new Gap neighborhood relation with adapted neighbors and proposed a new online streaming feature selection method, called OFS-A3M. The Gap neighborhood relation uses the gap information between neighbors of the target object to decide the set of final considered instances. In terms of Neighborhood Rough Set Theory, OFS-A3M does not need the domain knowledge before learning. Based on the Gap neighborhood relation, OFS-A3M needn't specify any parameters in advance. With three evaluation criteria of "maximal-dependency, maximal-relevance, and maximal-significance", our new approach can select features with high correlation, high dependency, and low redundancy. In order to validate the effectiveness of our new Gap neighborhood relation, we conducted the experiments to compare with δ neighborhood relation and k-nearest neighborhood relation. The results with the Friedman test showed that there is no significant difference between Gap and the other two neighborhood relations with optimal parameter values. When comparing to seven traditional feature selection methods and four state-of-the-art online feature selection algorithms, OFS-A3M performed better than traditional feature selection methods with the same number of features and it was superior to online streaming feature selection algorithms in an online manner.

However, the time complexity of OFS-A3M is relatively high in comparison with most of the state-of-the-art online streaming feature selection methods. Thus, our future work will focus on how to reduce the running time of our algorithm.

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] V. Bolón-Canedo, D. Rego-Fernández, D. Peteiro-Barral, A. Alonso-Betanzos, B. Guijarro-Berdiñas, N. Sánchez-Maroño, On the scalability of feature selection methods on high-dimensional data, Knowl. Inf. Syst. 56 (2) (2018) 395–442.
[2] 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.
[3] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (1) (2006) 1–30.
[4] 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.
[5] S. Eskandari, M. Javidi, Online streaming feature selection using rough sets, Int. J. Approx. Reason. 69 (C) (2016) 35–57.
[6] Q. Gu, Z. Li, J. Han, Generalized fisher score for feature selection, Proceedings of the Conference on Uai, 2011, pp. 266–273.
[7] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (1–3) (2002) 389–422.
[8] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, Adv Neural Inf. Process Syst. 17 (2005) 507–514.
[9] Q. Hu, J. Liu, D. Yu, Mixed feature selection based on granulation and approximation, Knowl. Based Syst. 21 (4) (2008) 294–304.
[10] Q. Hu, D. Yu, J. Liu, C. Wu, Neighborhood rough set based heterogeneous feature subset selection, Inf. Sci. (Ny) 178 (18) (2008) 3577–3594.
[11] Q. Hu, D. Yu, Z. Xie, Numerical attribute reduction based on neighborhood granulation and rough approximation, J. Softw. 19 (3) (2008) 640–649.
[12] M.M. Javidi, S. Eskandari, Streamwise feature selection: a rough set method, Int. J. Mach. Learn. Cybern. 9 (4) (2018) 667–676.
[13] S.U. Kumar, H.H. Inbarani, Pso-based feature selection and neighborhood rough set-based classification for BCI multiclass motor imagery task, Neural Comput. Appl. 28 (11) (2017) 3239–3258.
[14] H. Liu, H. Motoda, Computational methods of Feature Selection, Chapman and Hall/CRC Press, 2007.
[15] P. Maji, S. Paul, Rough set based maximum relevance-maximum significance criterion and gene selection from microarray data, Int. J. Approx. Reason. 52 (2011) 408–426.
[16] R.-J. Palma-Mendoza, D. Rodriguez, L. de Marcos, Distributed Relieff-based feature selection in spark, Knowl. Inf. Syst. 57 (1) (2018) 1–20.
[17] Z. Pawlak, Rough Sets - Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, Boston, 1991.
[18] 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.
[19] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.
[20] S. Perkins, J. Theiler, Online feature selection using grafting, in: Proceedings of the 20th International Conference on Machine Learning, 2003, pp. 592–599.
[21] M. Rahmaninia, P. Moradi, Osfsmi: online stream feature selection method based on mutual information, Appl. Soft. Comput. 68 (2018) 733–746.
[22] M. Robnik-Sikonja, I. Kononenko, Theoretical and empirical analysis of Relieff and Rrelieff, Mach. Learn. 53 (1–2) (2003) 23–69.
[23] S. Stawicki, D. Slezak, Recent advances in decision bireducts: complexity, heuristics and streams, Rough Sets Knowl. Technol. 8171 (2013) 200–212.
[24] R.W. Swiniarski, A. Skowron, Rough set methods in feature selection and recognition, Pattern Recognit. Lett. 24 (6) (2003) 833–849.
[25] T.Y. Lin, Granular computing on binary relations i: data mining and neighborhood systems, in: Proceedings of the Rough Sets in Knowledge Discovery, 1998, pp. 107–121.
[26] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B(Methodol.) (1996) 267–288.
[27] J.R. Vergara, P.A. Estévez, A review of feature selection methods based on mutual information, Neural Comput. Appl. 24 (1) (2014) 175–186.
[28] D. Wang, D. Irani, C. Pu, Evolutionary study of web spam: Webb spam corpus 2011 versus Webb spam corpus 2006, in: Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, in: CollaborateCom-2012, 2012, pp. 40–49.
[29] H. Wang, G. Wang, X. Zeng, S. Peng, Online streaming feature selection based on conditional information entropy, in: Proceedings of the 8th IEEE International Conference on Big Knowledge, 2017, pp. 230–235.
[30] 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.
[31] 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.

[32] M. Wasikowski, X. Chen, Combating the small sample class imbalance problem using feature selection, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1388–1400.
[33] 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.
[34] X. Xue, M. Yao, Z. Wu, A novel ensemble-based wrapper method for feature selection using extreme learning machine and genetic algorithm, Knowl. Inf. Syst. 57 (2) (2018) 389–412.
[35] K. Yang, Z. Cai, J. Li, G. Lin, A stable gene selection in microarray data analysis, BMC Bioinf. 7 (2006) 228.
[36] K. Yu, W. Ding, X. Wu, Lofs: library of online streaming feature selection, Knowl. Based Syst. 113 (1–3) (2016).
[37] 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).
[38] 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.
[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] J. Zhou, D.P. Foster, R.A. Stine, L.H. Ungar, Streamwise feature selection, J. Mach. Learn. Res. 3 (2) (2006) 1532–4435.
[41] 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, pp. 135–142.
[42] P. Zhou, X. Hu, P. Li, X. Wu, Online feature selection for high-dimensional class-imbalanced data, Knowl. Based Syst. 136 (2017) 187–199.
[43] D. Slezak, M. Grzegorowski, A. Janusz, M. Kozielski, S.H. Nguyen, M. Sikora, S. Stawicki, Ł. Wróbel, A framework for learning and embedding multi-sensor forecasting models into a decision support system: a case study of methane concentration in coal mines, Inf. Sci. (Ny) 451–452 (2018) 112–133.