A New Online Feature Selection Method Using Neighborhood Rough Set

Peng Zhou
Hefei University of Technology,
Hefei 230009, China
Email:doodzhou@hotmail.com

Xuegang Hu
Hefei University of Technology,
Hefei 230009, China
Email: jsjxhuxg@hfut.edu.cn

Peipei Li
Hefei University of Technology,
Hefei 230009, China
Email:peipeili@hfut.edu.cn

***Abstract*—Online feature selection, as a new method 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. In this paper, we define a new Neighborhood Rough Set relation with adapted neighbors and propose a new online streaming feature selection method based on this relation. Our approach 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, our new approach can select features with high correlation, high dependency and low redundancy. Experimental studies on ten different types of data sets show that our approach 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.**

I. Introduction

Feature selection is an important technique in data mining. Given a set of samples $U = (C, D)$, with some condition features C and decision classes D. Feature selection 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 [1].

With the increasing of the scale of data, traditional batch feature selection method can not meet the demand in efficiency any more. For example, Web Spam Corpus 2011, a collection of approximately 330,000 spam web pages and 16,000,000 features (attributes) [2]. Meanwhile, in many real-world applications, features are generated dynamically, and arrive one by one over time. A real-world application is the Mars crater detection from high resolution planetary images [3]. It is infeasible to acquire the entire feature set which means to have a near global coverage of the Martian surface. 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 [4], [5], [6], [7], [8], [9]. For instance, OSFS (Online Streaming Feature Selection) [5], a method of selecting strongly relevant and non-redundant features on the fly and SAOLA (a Scalable and Accurate OnLine Approach) [8], which employs novel online pairwise comparison techniques to address the extremely high dimensionality and highly scalable challenges in an online manner. However, all aforementioned algorithms need domain information or specify some parameters in advance. It is hence a challenge to select unified and optimal parameters before learning from different data sets.

Rough Set theory, proposed by Pawlak, has been proven to be an effective tool for feature selection, rule extraction and knowledge discovery [10]. Pawlak's rough sets are originally proposed to deal with categorical data. However, in real-world applications, there are many numerical features in data sets. Thus, Neighborhood Rough Set which supports both continuous and discrete data was proposed to deal with this challenge [11]. One of the most important advantages for Rough Set based data mining is that they do not require any other domain knowledge [9]. There are some works using neighborhood rough set for feature selection [12], [13], [14], [15], [16]. Nevertheless, all these methods mentioned above are proposed for traditional feature selection and they need to specify parameter values in advance. 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 which automatically select the number of neighbors for each target object by its surrounding instances distribution. In terms of this relation, a new online streaming feature selection algorithm is proposed to handle feature selection in an online manner. This new approach does not need to specify any parameters before learning and it can handle different types of data well. In addition, we use three maximal evaluation criteria ("maximal-dependency, maximal-relevance and maximal-significance") during feature selection which makes our new approach can select features with high correlation, high dependency and low redundancy.

The rest of 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.

II. Related Work

Feature selection is one of the most important techniques in machine learning. There are many representative algorithms for traditional feature selection, such as ReliefF [17], Fisher

978-1-5386-3120-1/17 $31.00 © 2017 IEEE
DOI 10.1109/ICBK.2017.41

Score [18], MI(Mutual Information)[19], mRMR (minimal Redundancy and Maximal Relevance) [20] and LASSO (Least Absolute Shrinkage and Selection Operator) [21]. All aforementioned approaches assume that all candidate features are available before learning. However, in many real-world applications, features are generated dynamically, and arrive one by one over time [22]. To deal with feature streams in an online manner, many online streaming feature selection methods have been proposed recently.

More specifically, Zhou et al. [4] 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. Wu et al. [5] 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 and online redundancy analysis. Yu et al. [8] 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. Eskandari et al. [9] proposed a Rough Set based method (OS-NRRSAR-SA) for online streaming feature selection. The proposed algorithm uses classical significance analysis concepts in Rough Set theory to control an unknown feature space in online streaming feature selection. However, Alpha investing, OSFS and SAOLA require the domain information before learning and specifying parameters in advance. OS-NRRSAR-SA is a classical Rough Set based method which can not deal with numerical features directly.

Meanwhile, we can find that there are some Neighborhood Rough Set based methods for feature selection, such as [12], [13], [14], [15], [16]. Nevertheless, all these methods are proposed for traditional batch feature selection, and as we know, there is no existing online streaming feature selection algorithm using Neighborhood Rough Set theory yet. In the following sections, we will give a brief introduction to Neighborhood Rough Set theory and present our new online streaming feature selection method based on the new Neighborhood Rough Set relation and three maximal evaluation criteria.

III. Neighborhood rough set

Neighborhood Rough Set is used to replace the approximation based on equivalence relation of traditional rough set model [10] with neighborhood relation, which supports both continuous and discrete data sets. In this section, we briefly review some basic concepts and notations of neighborhood rough set as follows [14], [15].

An information system $S = (U, A)$, where $U = \{x_1, x_2, ..., x_n\}$ is a nonempty finite set of objects, called a universe. $A = \{a_1, a_2, ..., a_m\}$ 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.

Definition 1: 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:

$$\delta_B(x) = \{y \mid \Delta(x, y) \leq \delta, y \in U\}, \tag{1}$$

where Δ is a distance function, 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 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 relation R, are defined as

$$\underline{R}(X) = \{x \in U \mid \delta(x) \subseteq X\} \tag{2}$$

$$\overline{R}(X) = \{x \in U \mid \delta(x) \cap X \neq \emptyset\} \tag{3}$$

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

$$BR(X) = \overline{R}(X) - \underline{R}(X) \tag{4}$$

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

$$\gamma_B(D) = \frac{CARD(POS_B(D))}{CARD(U)} \tag{5}$$

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.

IV. Our Online Feature Selection Method

In this section, we will introduce our new online streaming feature selection approach in detail. We first give a formal definition on online streaming feature selection. Then we introduce three evaluation criteria of "maximal-dependency, maximal-relevance and maximal-significance" based on the dependency between condition features and decision classes. In terms of the new neighborhood relation defined in our paper, we will present a new online streaming feature selection algorithm.

A. Definition of Online Streaming Feature Selection

Let $OFS = (U, C \cup D, f, t)$ denote an online streaming feature selection framework, where U is a non-empty finite set of objects, C is the condition attribute set, and D is the decision attribute set. Let $C = [x_1, x_2, ..., x_n]^T \in R^{n \times d}$ consist of n samples over a d-dimensional feature space $F = [f_1, f_2, ..., f_d]^T \in R^d$. Let $D = [y_1, y_2, ..., y_n]^T \in R^{n \times 1}$ consist of n samples over the class label (decision feature space) $L = [l_1, l_2, ..., l_m]^T \in R$. Given U, C and D, at each time stamp t, we get a feature f_t of $C \cup D$ without knowing the exact number of d in advance. The problem is to derive a mapping $f : C' \to L$ 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, 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 will introduce three evaluation criteria as follows.

B. 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 [23].

1) Maximal-dependency: Let $C = C_1, C_2, ..., C_m$ denote the set of m condition features of a given data set. The task of feature selection aims to find a feature subset $S \subseteq C$ with d features ($d < m$) which have the largest dependency $\mathbb{D}$ on the decision attributes set D.

$$\mathbf{Max}\mathbb{D}(S, D), \mathbb{D} = \gamma_{\{C_i, i=1,2,...d\}}(D), \tag{6}$$

where $\mathbb{D} = \gamma_{\{C_i, i=1,2,...d\}}(D)$ represents the dependency between the feature subset S and the target class label D as shown in Eq. 5.

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 [20]. 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.

2) Maximal-relevance: Maximal-relevance is to search feature with approximates $\mathbb{D}(S, D)$ using Eq.6 with the mean value of all dependency values between individual feature C_i and target class label D:

$$\mathbf{Max}\mathbb{R}(S, D), \mathbb{R} = \frac{1}{|S|} \sum_{C_i \in S} \gamma_{C_i}(D). \tag{7}$$

The dependency among features which 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.

3) Maximal-significance: The significance of a feature F to feature set S ($F \in S$) is defined as follows:

Definition 4: Given a condition attribute set S and a decision attribute set D, a feature $F \in S$, the significance of the feature F to S is defined as:

$$\sigma_S(D, F) = \gamma_S(D) - \gamma_{S-F}(D) \tag{8}$$

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:

$$\mathbf{Max}\mathbb{S}(S, D), \mathbb{S} = \frac{1}{|S|} \sum_{C_i \in S} \{\sigma_S(D, C_i)\}. \tag{9}$$

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. 6. Thus, we use the maximal-relevance condition to select relevant features and discard irrelevant features at first. Then we use the maximal-significance criterion to remove nonsignificant features in selected feature set. The maximal-dependency criterion will be used as the final goal of selecting the feature set with maximal dependency. More details refer to Section D.

C. Neighborhood Relation

Definition 1 defines the neighborhood relation with a fixed distance δ of the nearest neighbors to the target object. However, for different data sets, the distribution of samples is asymmetrical. It is difficult to select a uniform δ for all types of data. 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 follows.

Let $\mathbb{S}_C(x_i) = \{x_{(i,1)}, x_{(i,2)}, ..., x_{(i,n-1)} \mid x_i \cup x_{(i,1)} \cup x_{(i,2)} \cup ... \cup x_{(i,n-1)} = U, \Delta(x_i, x_{(i,1)}) \leq \Delta(x_i, x_{(i,2)}) \leq ... \leq \Delta(x_i, x_{(i,n-1)})\}$ denote all of the neighbors of x_i sorted by the distance from the nearest to the farest. From $x_{(i,1)}$ to $x_{(i,n-1)}$, assuming it is evenly distributed, we divided $\Delta(x_i, x_{(i,n-1)})$ into $n-1$ parts $\{P_1, P_2, ...P_{n-1} \mid Width_{P_1} = Width_{P_2} = ... = Width_{P_{n-1}} = \mathbb{P}\}$, and each part contains one sample. Certainly, it is always 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 $\mathbb{P}$, it is called a **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.

Based on this, we proposed a new neighborhood relation with adapted neighbors, denoted as $A_C(x)$ as shown in Eq. 10.

Definition 5: Given a set of finite and nonempty objects $U = \{x_1, x_2, ..., x_n\}$, the condition feature set C and a feature subset B ($B \subseteq C$). For target object x_i, let $N_{x_i} = \{x_{(i,1)}, x_{(i,2)}, ..., x_{(i,n-1)}\}$ denote all the neighbors of x_i form the nearest to the farest on B. The adapted neighborhood of arbitrary object $x_i \subseteq U$ on B is defined as:

$$A_B(x_i) = \{x \mid x \in \{x_{(i,1)}, x_{(i,2)}..., x_{(i,k)}\}, k \leq n-1\}, \quad (10)$$

where $Gap(x_{(i,k)}, x_{(i,k+1)})$ is the first **Gap** from $x_{(i,1)}$ to $x_{(i,n-1)}$.

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_{x_i}. Thus, the average distance between elements in N_{x_i} is $D_{mean} = \frac{Dist_{max} - Distmin}{n-1}$. We can define the width of **Gap** as $W_{Gap} = 1.5 * D_{mean}$. From $x_{(i,1)}$ to $x_{(i,n-1)}$, if $Gap(x_{(i,k)}, x_{(i,k+1)})$ is the first Gap, which means $\Delta(x_i, x_{(i,k+1)}) - \Delta(x_i, x_{(i,k)}) \geq W_{Gap}$ and all the neighbors $\{x_{(i,j)} \mid 2 \leq j \leq k\}$ have $\Delta(x_i, x_{(i,j)}) - \Delta(x_i, x_{(i,j-1)}) < W_{Gap}$, we will consider $\{x \mid x \in \{x_{(i,1)}, x_{(i,2)}, ..., x_{(i,k)}\}$ as the neighbors of x_i.

$A_B(x)$ uses different numbers of neighbors for dependency calculation which makes it competent to handle different kind of data. The dependency calculating method using this new neighborhood relation, denoted as **Dep-A** is given as follows.

Algorithm 1 Dep-A

Require:
X_S: sample values on feature set S;
R: neighborhood relation $A_B(x)$,
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;

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 the neighbors of x_i by distance. The time complexity of quick sort function is $O(n * logn)$. Thus, the time complexity of **Dep-A** is $O(|X_S|^2 * log|X_S|)$. In the next section, we use **Dep-A** for neighborhood dependency calculation in our new online streaming feature selection algorithm.

D. Our New Online Feature Selection Algorithm

In this section, we introduce our new online feature selection algorithm using the new neighborhood relation $A_B(x)$ and the "maximal-dependency, maximal-relevance and maximal-significance" evaluation criteria mentioned above, called "OFS-A3M" as shown in Alg. 2. The main goal of this online feature selection algorithm is to maximize Dep_S with the feature streams.

Algorithm 2 OFS-A3M

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: Dep_S:the dependency of S to Y, initialized to 0;
3: $Mean_{Dep_S}$: the mean dependency of features in S, $Mean_{Dep_S} = \frac{1}{|S|} \sum_{C_i \in S} \gamma_{C_i}(D)$, initialized to 0;
4: **Repeat**
5: Get a new feature f_i of X at time stamp t_i as X_{f_i};
6: Calculate the dependency of X_{f_i} as γ_{f_i} using **Dep-A**;
7: IF $\gamma_{f_i} < Mean_{Dep_S}$
8: discard feature f_i ; and go to Step 24;
9: ELSE
10: IF $\gamma_{S \cup f_i} > Dep_S$
11: $S = S \cup f_i$
12: ELSE IF $\gamma_{S \cup f_i} == Dep_S$
13: $S = S \cup f_i$
14: FOR each feature in S
15: randomly select a feature f' in S
16: calculate f' 's significance as $\sigma_S(f')$
17: IF $\sigma_S(f') = 0$
18: remove feature f' from S
19: END IF
20: END FOR
21: ELSE
22: discard feature f_i
23: END IF
24: **Until** no more features are available;
25: **return** S;

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-A**. All the following dependency computation calls from step 7 to step 17 use Algorithm 1 (**Dep-A**) 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 $Mean_{Dep_S}$ and we add f_i into S, the $Mean_{Dep_S}$ will decrease. Thus, with the "maximal-relevance" constraint, f_i is discarded when it is smaller than $Mean_{Dep_S}$.

If f_i satisfies the "maximal-relevance" constraint, step 10 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 with the "maximal-dependency" constraint.

If the dependency of $S \cup f_i$ is equal to Dep_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. 9. We will discard features whose significance equal to 0. By the "maximal-relevance", "maximal-dependency" and "maximal-significance" constraints, we can select features with high correlation, high dependency and low redundancy.

E. Time Complexity of OFS-A3M

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

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 C, the time complexity of **Dep-A** 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 $Mean_{Dep_S}$ (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 current 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 smaller than Dep_S, f_i will be discarded. Only if the dependency of $S \cup f_i$ is equal to 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-case time complexity of OFS-A3M is $O(F * |S| * N^2 logN)$.

V. Experimental Results

A. Experiment Setup

In this section, we apply the proposed online feature selection algorithm on ten data sets, including two UCI data sets (WDBC, HILL VALLEY with noise), seven DNA microarray data sets (PROSTATE, DLBCL, GLIOMA, SRBCT, LUNG2, MLL, CAR) [24], [25] and one NIPS 2003 data set (ARCENE) [5] as shown in Table I.

TABLE I
Experimental Data sets

Data Set	Instances	Features	Classes
WDBC	569	30	2
HILL	606	100	2
SRBCT	83	2308	4
LUNG2	203	3312	5
GLIOMA	50	4433	4
MLL	72	5848	3
PROSTATE	102	5966	2
DLBCL	77	6285	2
CAR	174	9182	11
ARCENE	200	10000	2

In our experiments, we use two basic classifiers, KNN and SVM in Matalab R2015b to evaluate a selected feature subset. We perform 10-fold cross-validation on each data set. All experimental results are conducted on a PC with Intel(R) i5-3470S, 2.9 GHz CPU, and 8 GB memory.

B. OFS-A3M vs. Traditional Feature Selection Methods

In this section, we compare OFS-A3M with three representative traditional feature selection methods, including ReliefF [17], PCC (Pearson Correlation Coefficient) [26] and MI (mutual information) [19].

All these algorithms are implemented in MATLAB. The K value of ReliefF is set to 5 for the best performance. None of these three 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 for OFS-A3M. We evaluate OFS-A3M and all competing ones on the predictive accuracy with 10-fold cross-validation.

Table II and Table III summarize the prediction accuracy of OFS-A3M against the other three competing algorithms using the basic classifiers of KNN (k=1) and SVM .

TABLE II
Predictive accuracy using the KNN classifier (%)

Data Set	OFS-A3M	MI	PCC	ReliefF
WDBC	95.43	95.25	**95.78**	**95.78**
HILL	**58.91**	52.97	50.17	54.62
SRBCT	**100**	98.8	96.39	87.95
LUNG2	**99.01**	90.15	84.24	86.21
GLIOMA	**100**	64	74	44
MLL	**100**	86.67	53.33	**100**
PROSTATE	**96.08**	92.16	91.18	90.2
DLBCL	**100**	89.61	84.42	96.1
CAR	**97.7**	89.66	83.91	90.8
ARCENE	**77**	72	52	55
AVERAGE	**92.41**	83.13	76.54	80.07

TABLE III
Predictive accuracy using SVM as the base calssifier (%)

Data Set	OFS-A3M	MI	PCC	ReliefF
WDBC	**97.36**	96.31	97.01	96.84
HILL	**53.63**	51.16	49.83	54.13
SRBCT	91.57	95.18	**96.39**	90.36
LUNG2	**95.57**	90.15	84.73	89.66
GLIOMA	**86**	58	74	48
MLL	**100**	**100**	26.67	**100**
PROSTATE	92.16	**95.1**	92.16	91.18
DLBCL	**97.4**	92.21	88.31	88.31
CAR	89.08	**89.66**	85.06	**89.66**
ARCENE	**71**	67	52	64
AVERAGE	**87.67**	83.67	74.62	80.92

From Table II and Table III, we have the following observations.

- OFS-A3M vs. PCC. OFS-A3M outperforms PCC on nine of the ten data sets at least in both cases. PCC gets the predictive accuracy about 50% and 70% on data sets ARCENE and GLIOMA with both KNN and SVM, while OFS-A3M gets the predictive accuracy by higher than

70% and 90% respectively. OFS-A3M is higher PCC 20% on predictive accuracy on these two data sets. Thus, OFS-A3M can handle different types of data better than PCC.

- OFS-A3M vs. ReliefF. OFS-A3M gets the higher predictive accuracy than ReliefF on eight of the ten data sets. ReliefF is similar to OFS-A3M, because they both use the neighbors's information for feature selection. However, OFS-A3M is superior to ReliefF on predictive accuracy with adapted neighbors.
- OFS-A3M vs. MI. OFS-A3M outperforms MI on seven of the ten data sets at least. Especially on data set GLIOMA, OFS-A3M over MI almost 30% on predictive accuracy with both KNN and SVM.

In sum, OFS-A3M provides best overall performance on seven of the ten data sets, while it is also comparable to the best competing approaches on the rest three data sets. Meanwhile, OFS-A3M gets the highest mean predictive accuracy with both KNN and SVM.

C. OFS-A3M vs. Online Feature Selection Methods

In this section, we compare our algorithm with four state-of-the-art online feature selection methods: Alpha-investing [4], OSFS [5], Fast-OSFS [5], SAOLA [8].

All aforementioned algorithms are implemented in MATLAB [27]. 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 [4].

Tables IV and Table V summarize the predictive accuracy of OFS-A3M against the other four algorithms using the KNN (k=1) and SVM classifiers. Table VI and Table VII show the running time and number of selected features of OFS-A3M against other four algorithms.

From Tables IV - VII , we have the following observations.

- OFS-A3M vs. Alpha-investing. Alpah-investing is the fastest algorithm among all these five compared methods. However, in Table IV and Table V, we can see that OFS-A3M outperforms Alpha-investing on eight of the ten data sets at least with both KNN and SVM. Meanwhile, we can see that the features selected by Alpha-investing can not fit for some data sets. For example, Alpha-investing only gets the predictive accuracy of 38% and 56% on GLIOMA 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. OFS-A3M outperforms OSFS on eight of the ten data sets with both KNN and SVM. On data sets HILL, OSFS can not select any features and gets the prediction accuracy 0. In addition, on GLIOMA, OSFS only gets the predictive accuracy 66% and 74% in cases of KNN and SVM respectively, while OFS-A3M gets the predictive accuracy 100% and 90%. 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. OFS-A3M performs better than Fast-OSFS on eight of the ten data sets. 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. SAOLA is faster than OFS-A3M. However, OFS-A3M outperforms SAOLA on nine of the ten data sets with both KNN and SVM. On the data set HILL, SAOLA can not select any features and get the predictive accuracy 0. This demonstrates that SAOLA can not handle some types of data well and can not select any features on these data sets. Thus, OFS-A3M is superior to SAOLA.

In sum, our algorithm OFS-A3M is not faster than some competing algorithms of Alpah-investing, Fast-OSFS and SAOLA, but it outperforms all competing algorithms on all data sets.

VI. Conclusion

In this paper, we gave a brief introduction of Neighborhood Rough Set theory and defined a new neighborhood relation with adapted neighbors. In order to select features which can get high separability, we used the information of adapted number of neighboring instances near by the target object. Based on this new neighborhood relation, we proposed a new online streaming feature selection approach. With three evaluation criteria "maximal-dependency, maximal-relevance and maximal-significance", our new approach can select features with high correlation, high dependency and low redundancy. As compared to three traditional feature selection methods and four state-of-the-art online feature selection algorithms, the proposed algorithm performs better on feature selection with feature streams in an online manner.

Acknowledgment

This work is supported in part by the National Key Research and Development Program of China under grant 2016YFB1000901, the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education, China, under grant IRT13059, the Specialized Research Fund for the Doctoral Program of Higher Education under grant 20130111110011, the Natural Science Foundation of China under grants (61273292, 61229301, 61503112, 61673152).

References

[1] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. Chapman and Hall/CRC Press, 2007.

[2] D. Wang, D. Irani, and 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*, ser. CollaborateCom-2012, 2012, pp. 40–49.

[3] W. Ding, T. F. Stepinski, Y. Mu, L. Bandeira, R. Ricardo, Y. Wu, Z. Lu, T. Cao, and X. Wu, "Subkilometer crater discovery with boosting and transfer learning," *Acm Transactions on Intelligent Systems & Technology*, vol. 2, no. 4, pp. 1–22, 2011.

[4] J. Zhou, D. P. Foster, R. A. Stine, and L. H. Ungar, "Streamwise feature selection," *Journal of Machine Learning Research*, vol. 3, no. 2, pp. 1532–4435, 2006.

TABLE IV
PREDICTIVE ACCURACY USING KNN AS THE BASE CLASSIFIER (%)

Data Set	OFS-A3M	Alpha-investing	OSFS	Fast-OSFS	SAOLA
WDBC	95.25	95.61	**96.13**	95.78	91.56
HILL	**57.59**	51.49	0	0	0
SRBCT	**100**	90.36	87.95	87.95	93.98
LUNG2	**99.01**	92.61	84.24	86.21	89.66
GLIOMA	**100**	38	66	78	76
MLL	86.67	**93.33**	**93.33**	73.33	80
PROSTATE	**97.06**	90.2	93.14	90.2	96.08
DLBCL	**100**	92.21	98.7	96.1	97.4
CAR	**98.28**	75.29	63.79	76.44	81.61
ARCENE	**73**	55	51	66	62
AVERAGE	**90.69**	77.41	73.43	75.00	76.83

TABLE V
PREDICTIVE ACCURACY USING SVM AS THE BASE CLASSIFIER (%)

Data Set	OFS-A3M	Alpha-investing	OSFS	Fast-OSFS	SAOLA
WDBC	**97.89**	97.01	96.31	96.31	91.92
HILL	**51.16**	50.33	0	0	0
SRBCT	**95.18**	92.77	91.57	90.36	92.77
LUNG2	92.12	**95.07**	90.15	90.15	89.66
GLIOMA	**90**	56	74	84	82
MLL	**100**	80	93.33	80	73.33
PROSTATE	90.2	**97.06**	94.12	95.1	97.06
DLBCL	**98.7**	96.1	96.1	97.4	**98.7**
CAR	**89.08**	72.41	64.37	78.74	86.21
ARCENE	**74**	70	64	69	65
AVERAGE	**87.83**	80.67	76.40	78.10	77.66

TABLE VI
RUNNING TIME (SECONDS)

Data Set	OFS-A3M	Alpha-investing	OSFS	Fast-OSFS	SAOLA
WDBC	2.6282	**0.0037**	0.2204	0.1026	0.0149
HILL	5.6747	**0.0055**	0.0151	0.0149	0.0154
SRBCT	5.3277	**0.2348**	10.991	1.313	0.8906
LUNG2	40.4546	**0.7753**	523.7902	12.9141	2.5247
GLIOMA	6.0151	**0.2372**	21.6944	2.1758	2.5885
MLL	10.3455	**0.3735**	36.1819	3.1331	5.0869
PROSTATE	11.1533	**0.4904**	13.3175	1.8618	1.7808
DLBCL	12.9231	**0.458**	20.5061	1.9523	2.1477
CAR	82.852	**1.4673**	868.9788	14.5953	4.5173
ARCENE	32.718	**0.8319**	20.4237	2.1466	4.9024
AVERAGE	21	**0.4877**	151.61	4.02	2.45

TABLE VII
THE NUMBER OF SELECTED FEATURES

Data Set	OFS-A3M	Alpha-investing	OSFS	Fast-OSFS	SAOLA
WDBC	18	20	3	4	**2**
HILL	9	**4**	0	0	0
SRBCT	12	26	**5**	8	17
LUNG2	26	45	**11**	16	30
GLIOMA	17	4	**3**	7	17
MLL	9	7	**3**	8	28
PROSTATE	30	12	**3**	5	12
DLBCL	12	8	**5**	8	22
CAR	44	25	**8**	14	40
ARCENE	27	**4**	5	7	37
AVERAGE	20.4	15.5	**4.6**	7.7	20.5

[5] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1178–1192, 2013.

[6] J. Wang, M. Wang, P. Li, L. Liu, Z. Zhao, X. Hu, and X. Wu, "Online feature selection with group structure analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, pp. 3029–3041, 2015.

[7] J. Wang, P. Zhao, S. C. Hoi, and R. Jing, "Online feature selection and its applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 698–710, 2013.

[8] K. Yu, X. Wu, W. Ding, and J. Pei, "Scalable and accurate online feature

selection for big data," *ACM Transactions on Knowledge Discovery from Data*, vol. 11, no. 2, 2016.
[9] S. Eskandari and M. Javidi, "Online streaming feature selection using rough sets," *International Journal of Approximate Reasoning*, vol. 69, no. C, pp. 35–57, 2016.
[10] Z. Pawlak, *Rough Sets - Theoretical Aspects of Reasoning about Data*. Dordrecht , Boston: Kluwer Academic Publishers, 1991.
[11] L. T and 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.
[12] Q. Hu, D. Yu, J. Liu, and C. Wu, "Neighborhood rough set based heterogeneous feature subset selection," *Information Sciences*, vol. 178, no. 18, pp. 3577–3594, 2008.
[13] Q. Hu, J. Liu, and D. Yu, "Mixed feature selection based on granulation and approximation," *Knowledge-Based Systems*, vol. 21, no. 4, pp. 294–304, 2008.
[14] J. Zhang, T. Li, D. Ruan, and D. Liu, "Neighborhood rough sets for dynamic data mining," *International Journal of Intelligent Systems*, vol. 27, no. 4, pp. 317–342, 2012.
[15] Q. Hu, D. Yu, and Z. Xie, "Numerical attribute reduction based on neighborhood granulation and rough approximation," *Journal of Software*, vol. 19, no. 3, pp. 640–649, 2008.
[16] S. U. Kumar and H. H. Inbarani, "Pso-based feature selection and neighborhood rough set-based classification for bci multiclass motor imagery task," *Neural Computing and Applications*, pp. 1–20, 2016.
[17] M. Robnik-Sikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Machine Learning*, vol. 53, no. 1-2, pp. 23–69, 2003.
[18] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," in *Conference on Uai*, 2011.
[19] J. R. Vergara and P. A. Estvez, "A review of feature selection methods based on mutual information," *Neural Computing and Applications*, vol. 24, no. 1, pp. 175–186, 2014.
[20] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
[21] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B(Methodological)*, pp. 267–288, 1996.
[22] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu, "Multimodal graph-based reranking for web image search," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4649–4661, 2012.
[23] P. Maji and S. Paul, "Rough set based maximum relevance-maximum significance criterion and gene selection from microarray data," *International Journal of Approximate Reasoning*, vol. 52, pp. 408–426, 2011.
[24] K. Yang, Z. Cai, J. Li, and G. Lin, "A stable gene selection in microarray data analysis," *BMC Bioinformatics*, vol. 7, p. 228, 2006.
[25] L. Yu, C. Ding, and 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.
[26] M. Wasikowski and X. Chen, "Combating the small sample class imbalance problem using feature selection," *IEEE Transactions On Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1388–1400, 2010.
[27] K. Yu, W. Ding, and X. Wu, "Lofs: Library of online streaming feature selection," *Knowledge-Based Systems*, vol. 113, no. 1-3, 2016.