Knowledge-Based Systems 226 (2021) 107157

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Online group streaming feature selection considering feature interaction

Peng Zhou [a,b,c], Ni Wang [b], Shu Zhao [a,b,c,*]

[a] Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui Province 230601, PR China
[b] School of Computer Science and Technology, Anhui University, Hefei, Anhui Province 230601, PR China
[c] Information Materials and Intelligent Sensing Laboratory of Anhui Province, Anhui Province 230601, PR China

ARTICLE INFO

Article history:
Received 16 June 2020
Received in revised form 11 November 2020
Accepted 15 May 2021
Available online 18 May 2021

Keywords:
Feature selection
Streaming feature selection
Streaming groups
Mutual Information
Elastic net

ABSTRACT

In real-world applications, features can be generated continuously one by one or by groups, such as image analysis and physical examination. Online streaming feature selection deals with streaming features on the fly. Existing streaming feature selection methods focus on removing irrelevant and redundant features and selecting the most relevant features, but they ignore the interaction between features. Interacting features appear to be irrelevant or weakly relevant to the class individually. However, if they are combined, they may highly correlate with the class. Features within the same group are more likely to interact with each other. Therefore, in this paper, we focus on feature interaction within and between the streaming groups and propose an Online Group Streaming Feature Selection method that can select Features to Interact with each other, named OGSFS-FI. OGSFS-FI consists of two stages: online intra-group selection and online inter-group selection. For intra-group selection, we design a new pair selection strategy that can select features interacting with each other. For inter-group selection, we use the regularization and variable selection method elastic net, which encourages a grouping effect. Extensive experiments conducted on synthetic and real-world datasets demonstrate our new method's efficiency and effectiveness.

1. Introduction

Feature selection aims to select a minimal size subset of the feature space, which can retain the optimum salient characteristics necessary from the original data sets [1]. Traditional feature selection methods assume that all features are presented to a learner before learning takes place [2]. Nevertheless, in real-world applications, such as image analysis [3] and Mars crater detection [4], not all features can be presented before learning. Streaming features are defined as features that flow in one by one or by groups over time, whereas the number of training examples is fixed [5,6]. Streaming feature selection, which deals with streaming features online, has attracted much attention in recent years [7].

Furthermore, streaming feature selection can be divided into two categories: individual streaming feature selection and group streaming feature selection, such as OSFS (Online Streaming Feature Selection) [5] and OGFS (Online Group Feature Selection) [8] respectively. In real-world applications, features man be generated in groups and can be required in streams. The various inspection items in a physical examination project is a typical example of this. For the same person, different inspection items generate different groups of features, such as blood routine examination, urine routine examination, ECG (Electrocardiograph), etc. Each inspection generates its features independently at a different time. For the whole examination project, features are generated separately and flow in a group by a group over time. Another example is the environmental monitoring and analysis [9]. Researchers may deploy many sets of observation stations in different areas. Each station may have several monitors treated as a group of objects in the data collection. In other words, the number of data groups (objects) is fixed. However, the number of features in temporal domains keeps increasing.

Fig. 1 shows the details of traditional feature selection and online group streaming feature selection. For online group streaming feature selection, the features arrived by groups. At each timestamp, we have the currently selected feature subset S' and gets the new arriving streaming group G. With the intra-group selection, we gets the new selected feature subset G'. Then, we process the inter-group selection and retain the selected feature subset S. The algorithm terminated and returned the final selected features until no more feature groups available.

In general, feature selection focuses on removing irrelevant and redundant features from the feature space and selecting

* Correspondence to: 111 Jiulong Road, Hefei, Anhui, China.
E-mail addresses: doodzhou@hotmail.com (P. Zhou), wangni613@hotmail.com (N. Wang), zhaoshuzs2002@hotmail.com (S. Zhao).

https://doi.org/10.1016/j.knosys.2021.107157
0950-7051/

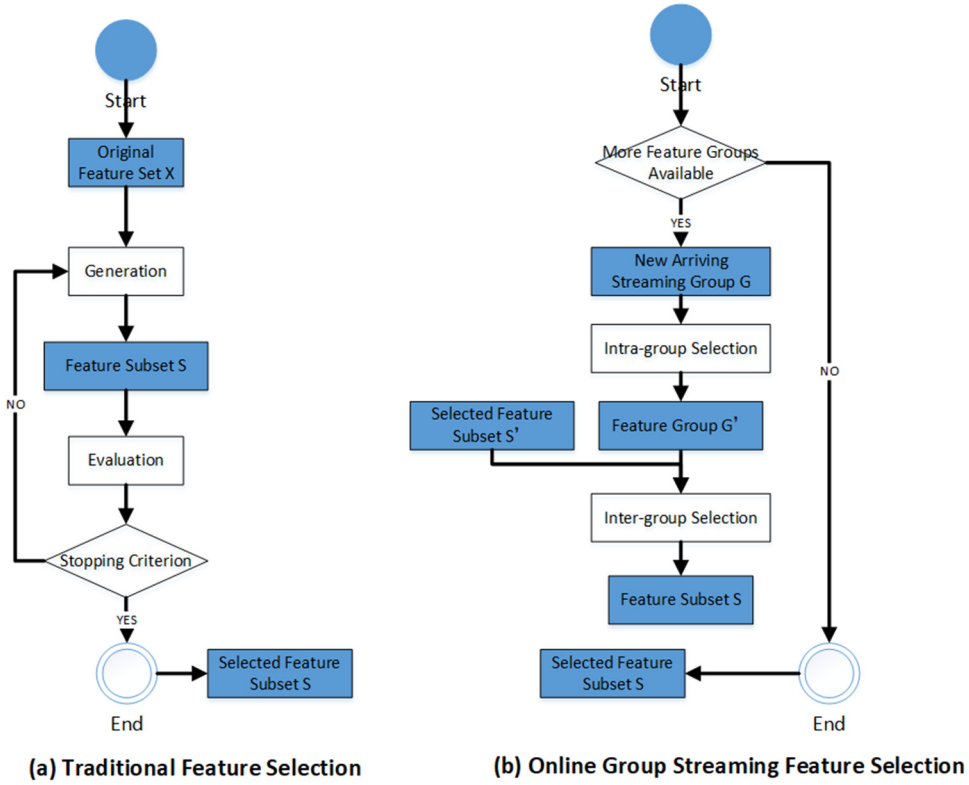


Fig. 1. The traditional feature selection(a) and online group streaming feature selection(b).

Table 1
An example of feature interaction problem.

f_1	f_2	f_3	f_4	D
0	0	0	0	0
1	0	0	1	1
0	1	0	0	1
1	1	0	1	1
0	0	1	0	1
1	0	1	1	1
0	1	1	0	0
1	1	1	1	1

the most relevant and informative ones. Features can be further categorized into three disjoint groups: strong relevance, weak relevance, and irrelevance [10,11]. Irrelevant features provide no information for the outcome in any context. Redundant features provide information for the outcome in some context (also called weak relevance), but they are not necessary for optimal prediction. Besides, an essential but usually being ignored issue is feature interaction [12]. Interacting features are those that appear to be irrelevant or weakly relevant with the class individually, but when the feature is combined with other features, it may positively correlate to the class [13].

An illustration of this interaction phenomenon is as shown in Table 1. Let f_1, f_2, f_3 be independent binary random variables. The output of a given system is built through the function $D = f_1 + (f_2 \oplus f_3)$, where $f_4 = f_1$, '+ ' stands for the OR logic function and '$\oplus$' represents the XOR logic function. We use the Mutual Information [14] $I(f_i; D)$ to evaluate the strength of the relevance between a feature f_i and the class D. In Table 1, we can see that f_2 and f_3 have a null relevance individually, i.e., $I(f_2; D) = I(f_3; D) = 0$. However, the joint information of $I(\{f_2, f_3\}; D) = 0.3113 > I(f_2; D) + I(f_3; D) = 0$, in which we call there is an interaction between f_2 and f_3 on D. By contrast, $I(f_1; D) = 0.3113$, $I(f_4; D) = 0.3113$, but $I(\{f_1, f_4\}; D) = 0.3113 < I(f_1; D) + I(f_4; D)$. Then, we denotes that there is a redundancy between f_1 and f_4 on D. In brief, feature interaction means "$1 + 1 > 2$" while feature redundancy denotes "$1 + 1 < 2$".

For streaming feature selection, we do not know the feature space before learning. Individual streaming feature selection methods process features individually during feature selecting and cannot handle the feature interaction, such as Alpha-investing [15], OSFS [5], and SAOLA [9]. Meanwhile, existing group streaming feature selection approaches do not consider feature interaction within the groups, such as GFSSF [6], OGFS [8], and Group-SAOLA [9]. Besides, many effective and efficient learning algorithms assume the independence of features. However, they may fail badly when the degree of feature interaction becomes critical [16]. Most of the above-mentioned streaming feature selection methods validate the features individually. However, features may influence the class by grouping rather than by the individual.

Motivated by this, we focused on feature interaction within and between streaming groups and proposed a new online Group Streaming Feature Selection method, named OGSFS-FI. Our main contributions are as follows:

- Based on Mutual Information theory, we gave the formal definition of feature interaction and discussed the relationship between interaction, redundancy, and relevance.
- We present the OGSFS-FI that consists of two components: intra-group selection and inter-group selection. For intra-group selection, we designed a new pair selection strategy that can guarantee the selected features relevant to the class and interact with each other. For inter-group selection, we use the regularization and variable selection method, elastic net, which prefers to select a group of features. Based on this, OGSFS-FI can efficiently select relevant and interactive features and remove redundancy ones.

- To investigate our new method's effectiveness, experimental results on synthetic datasets and real-world datasets demonstrated that OGSFS-FI could select relevant and interactive features on the fly.

The rest of this paper is organized as follows. In Section 2, we describe the related work. In Section 3, some basic information-theoretic notions are reviewed. Section 4 provides formal definitions of relevance, redundancy, and interaction in the framework of Mutual Information theory and proposes a new group streaming feature selection algorithm in Section 5. Experimental analyses are presented in Section 6, and we make a brief conclusion in Section 7.

2. Related work

Generally speaking, feature selection removes irrelevant and redundant features and selects relevant features from a given candidate feature space. For example, [17] proposed a novel neighborhood multi-granulation rough sets based feature selection method using Lebesgue and entropy measures in incomplete neighborhood decision systems. From the data perspective, we can divide feature selection into feature selection with static data and feature selection with dynamic data [18]. For feature selection with static data, there are few works considering feature interaction during the feature selection. More specifically, Jakulin et al. [12] first formally define the degree of interaction between attributes through the deviation of the best possible "voting" classifier from the true relation between the class and the features in a domain. Furthermore, they introduced an operational definition of a generalized n-way interaction by highlighting two models: the reductionistic part-to-whole approximation and the holistic reference model [16]. Zhao et al. [19] took up the challenge of feature interaction to design a particular data structure for the feature quality evaluation and to employ an information-theoretic feature ranking mechanism to handle feature interaction in the subset selection efficiently. Zeng et al. [13] proposed a novel feature selection algorithm considering feature interaction. They defined the interaction weight factor, reflecting whether a feature is redundant or interactive, and designed an interaction weight-based feature selection algorithm.

The feature selection with dynamic data can be further divided into feature selection with data stream and feature selection with feature stream. In this paper, we focus on the latter. As compared to traditional feature selection methods that deal with static data, streaming feature selection assumes that the features flow in one by one over time. We cannot require the information of the whole feature space before learning. All these methods mentioned above are designed for traditional feature selection, and they need the information of the whole feature space before learning. Therefore, they cannot handle the streaming feature selection.

There are two significant reasons for streaming feature selection: (1) the feature space is unknown or even infinite, and (2) the feature space is known, but feature streaming offers many other advantages. More specifically, Grafting [20] first considered the problem of online feature selection and treats feature selection as an integral part of learning a predictor within a regularized framework. Information-investing and alpha-investing [15] were two penalized likelihood ratio methods based on streamwise regression for online feature selection. OSFS [5] is an online streaming feature selection framework that contained two major steps: online relevance analysis (discarding irrelevant features) and online redundancy analysis (eliminating redundant features). SAOLA [9] was a Scalable and Accurate Online feature selection Approach for high dimensional data which employed novel online pairwise comparison techniques and maintained a parsimonious model over time in an online manner. OS-NRRSARA-SA [21] was a Rough Set based method for online streaming feature selection which considered both the boundary and positive regions. [22] was an extension to the OS-NRRSAR-SA algorithm that containing two major steps: online redundancy analysis that discards redundant features, and online significance analysis which eliminates non-significant features. K-OFSD [23] was proposed for high-dimensional and class-imbalanced data in online streaming feature selection, which is based on the dependency between condition features and decision classes. OFS-A3M [24] was a new neighborhood rough relation based streaming feature selection method which can select features with high correlation, high dependency, and low redundancy in terms of the maximal-dependency, maximal-relevance, and maximal-significance evaluation criteria. ROSFSMI [25] employed Mutual Information in a streaming manner to evaluate the relevancy and redundancy of features.

Nevertheless, all these streaming feature selection methods mentioned above are individual streaming feature selection, which cannot handle the feature interaction problem. Meanwhile, considering the group structures of feature streams, group streaming feature selection was proposed. GFSSF (Group Feature Selection with Streaming Features) [6] first performed group feature selection with streaming features that can work at both the group and individual feature levels for streaming feature selection exploiting entropy and Mutual Information in information theories. OGFS (Online Group Feature Selection) [8] was an efficient online feature selection framework using the prior knowledge of group information, which consisted of two stages as the intra-group feature selection and inter-group features selection. Group-SAOLA [9] extends the SAOLA algorithm, and it can select feature groups that are sparse at the levels of both features and groups.

However, all these group streaming feature selection approaches do not consider the interaction between features. Thus, contrary to the approaches mentioned above, we propose a new group streaming feature selection method considering the feature interaction during streaming feature selection.

3. Definitions of relevance, redundancy and interaction

In this section, we give a formal definition of feature interaction in terms of Mutual Information theory. Information theory, proposed by Shannon, provides a way to measure the information of random variables [26]. The mutual information (MI) is a measure of the amount of information that one random variable has about another variable [14]. Multi-information is an extension of MI that can measure the interaction among more than two variables [27]. For the case of three variables, the multi-information is defined as follows:

$$I(X;Y;Z) = \begin{cases} I(Y;Z) - I(Y;Z|X) \\ I(X;Y) - I(X;Y|Z) \\ I(X;Z) - I(X;Z|Y) \end{cases} \tag{1}$$

The multi-information is symmetrical, such as $I(X;Y;Z) = I(Y;X;Z)$.

Mutual information tends to favor features with more values. In order to solve this problem, a normalized measure of mutual information called the symmetrical uncertainty is used as follows [28]:

$$SU(X;Y) = \frac{2 * I(X;Y)}{H(X) + H(Y)} \tag{2}$$

The symmetrical uncertainty can compensate for Mutual Information's bias toward features with more values and restricts its values to the range [0, 1] by penalizing inputs with large

entropies. In our algorithm, we use symmetrical uncertainty for the information calculating between a feature subset and the decision feature in Algorithm 1.

For data with continuous values, we adopt the best-known measure of Fisher's Z-test [29] to calculate correlations between features. In a Gaussian distribution $Normal(\mu, \Sigma)$, the population partial correlation $P(X, Y|S)$ between feature X and the feature Y given a feature subset S is calculated as follows:

$$P(X, Y|S) = \frac{-((\sum_{XYS})^{-1})_{XY}}{((\sum_{XYS})^{-1})_{XX}((\sum_{XYS})^{-1})_{YY}} \quad (3)$$

In Fisher's Z-test, under the null hypothesis of conditional independence between X and Y given S, $P(X, Y|S) = 0$. With the given significance level α and the p-value returned by Fisher's Z-test p, under the null hypothesis of the conditional independence, if $p > \alpha$, X and Y are uncorrelated; otherwise, if $p \leq \alpha$, X and Y are correlated to each other.

For an information system, let C denotes the condition feature set, and D denotes the decision class. The features in C can be categorized into three disjoint groups: strong relevance, weak relevance, and irrelevance as follows [11]. Based on Markov blankets, Yu and Liu [30] further divided weakly relevant features into redundant and non-redundant features. First of all, let us define strong relevance, weak relevance, and irrelevance based on the Mutual Information-theoretic framework.

Definition 1 (*Strong Relevance, Weak Relevance, and Irrelevance*)**.** Given C and D, $f \in C$,

(1) f is strongly relevant to D iff $\forall S \subseteq C\backslash\{f\}$ s.t. $I(D; S) \neq I(D; \{S, f\})$;

(2) f is weakly relevant to D iff it is not strongly relevant, and $\exists S \subset C\backslash\{f\}$ s.t. $I(D; S) \neq I(D; \{S, f\})$.

(3) f is irrelevant to D iff it is neither strongly nor weakly relevant, and $\forall S \subseteq C\backslash\{f\}$ s.t. $I(D; S) = I(D; \{S, f\})$.

For interactive features, the information of the combined feature set is bigger than the sum of each feature. Thus, the formal definition of feature interaction as follows.

Definition 2 (*Interaction*)**.** Given C and D, $INT \subseteq C$. For $\forall f \in INT$. If $I(D; INT) > \sum_{f_i \in INT} I(f_i; D)$, the features in INT are said to have an interaction with each other on D, and we call INT a interactive feature set.

Let us use the dataset Monks3 in Section 6.2.1 as an example to illustrate the definition of feature interaction. Meanwhile, we use Mutual Information to calculate the information between features. We calculate some values as follows.

$I(a_1; c) = 0.0071$, $I(a_2; c) = 0.2937$, $I(a_3; c) = 8.3111e-04$, $I(a_4; c) = 0.0029$, $I(a_5; c) = 0.2559$, $I(a_6; c) = 0.0071$. $I(\{a_2, a_4\}; c) = 0.3579 > I(a_2; c)+I(a_4; c) = 0.2966$, $I(\{a_2, a_4, a_5\}; c) = 0.8678 > I(a_2; c) + I(a_4; c) + I(a_5; c) = 0.5565$. According to Definition 2, the feature subsets $\{a_2, a_4\}$ and $\{a_2, a_4, a_5\}$ are interactive feature sets.

In general, feature selection aims to select relevant and non-redundant features from the condition feature set. Nevertheless, for high-dimensional datasets, the definitions of strong relevance, weak relevance, and irrelevance are hard to be applied directly, for we cannot test all the subsets of C. Thus, many feature selection methods always use a low-dimensional evaluating approach to judge the feature type. For example, for many Mutual Information based feature selection methods, if $I(f; D) = 0$ or $I(f; D) < \alpha$ (where α is a user-defined parameter), then f will be considered as an irrelevant feature [6,9]. As shown in Table 1, these methods ignore the interaction between features. For the convenience of discussion, let us consider the relationship between three variables at first.

Theorem 1. *Given C and D, $f_1 \in C$, $f_2 \in C$. If f_2 has an interaction with f_1 on D, then $I(\{f_1, f_2\}; D) > I(f_1; D) + I(f_2; D)$.*

Proof. According to Definition 2, we make $INT = \{f_1, f_2\}$. Because f_2 has an interaction with f_1 on D, thus, $I(D; INT) = I(D; \{f_1, f_2\}) > \sum_{f_i \in INT} I(f_i; D) = I(f_1; D) + I(f_2; D)$.

Theorem 2. *Given C and D, $f_1 \in C$, $f_2 \in C$. If f_2 has an interaction with f_1 on D, then $I(f_1; D|f_2) > I(f_1; D)$.*

Proof. For f_2 has an interaction with f_1 on D, According to Theorem 1, $I(\{f_1, f_2\}; D) > I(f_1; D) + I(f_2; D)$. $I(\{f_1, f_2\}; D) = I(f_2; D) + I(f_1; D|f_2) > I(f_1; D) + I(f_2; D)$. Therefore, $I(f_1; D|f_2) > I(f_1; D)$.

For a random feature f_1 in C, $I(f_1; D)$ is a measure of the amount of information that f_1 has about D. If there exist feature $f_2 (f_2 \in C)$ which can increase the information of f_1 about D when f_2 is known, f_2 is said to have an interaction with f_1 on D.

Theorem 3. *Given C and D, $f_1 \in C$, $f_2 \in C$. If f_2 has an interaction with f_1 on D, $I(f_1; f_2; D) < 0$.*

Proof. For f_2 has an interaction with f_1 on D, $I(f_1; D|f_2) > I(f_1; D)$, $I(f_1; D) - I(f_1; D|f_2) < 0$. According to Eq. (1), $I(f_1; f_2; D) = I(f_1; D) - I(f_1; D|f_2)$. Thus, $I(f_1; f_2; D) < 0$.

According to Theorem 1, when f_1 and f_2 are interactive with each other on D, they provide more information than the sum of their individual mutual information. In contrast with interaction, we give the definition of redundancy as follows.

Definition 3. Given C and D, $f_1 \in C$, $f_2 \in C$. If $I(\{f_1, f_2\}; D) < I(f_1; D) + I(f_2; D)$, f_1 and f_2 are said to have a redundancy on D.

In another words, when f_1 and f_2 both provide a part of the same information about D, there exists redundancy.

Theorem 4. *If $I(f_1; f_2; D) > 0$, f_1 and f_2 have a redundancy on D.*

Proof. According to Eq. (1), $I(f_1; f_2; D) = I(f_1; D) - I(f_1; D|f_2)$. For $I(f_1; f_2; D) > 0$, then $I(f_1; D|f_2) - I(f_1; D) < 0$, $I(\{f_1, f_2\}; D) - I(f_1; D) - I(f_2; D) = I(f_2; D) + I(f_1; D|f_2) - I(f_1; D) - I(f_2; D) = I(f_1; D|f_2) - I(f_1; D) < 0$, $I(\{f_1, f_2\}; D) < I(f_1; D) + I(f_2; D)$. Thus, f_1 and f_2 have a redundancy on D.

A given feature is relevant to the class when either individually or together with other variables provides information about decision attribute D. Thus, relevance should not just consider the target feature and class only, but it is conditionally dependent on S. Thus, we define feature relevance as below.

Definition 4 (*Relevance*)**.** Given C and D, $f_i \in C$ and $\neg f_i = C \setminus \{f_i\}$. Feature f_i is relevant to the class D if and only if $\exists S \subset \neg f_i$, such that

$$I(f_i; D|S) > 0. \quad (4)$$

Otherwise, feature f_i is **irrelevant**.

Theorem 5. *Given C and D, $f \in C$, if $\exists f' \in C$, such that $I(f; f'; D) < 0$, f is relevant to class D.*

Proof. For $I(f; f'; D) = I(f; D) - I(f; D|f') < 0$, $I(f; D|f') > I(f; D)$. Because $I(f; D) \geq 0$, then $I(f; D|f') > 0$. Assume $S = \{f'\}$. According to Definition 4, f is relevant to class D.

4. Our new online group streaming feature selection approach

In this paper, we focus on the problem of feature interaction during online group streaming feature selection. We first give the problem formalization of online group streaming feature selection. For features in the same streaming groups, we design a new intra-group selection approach that selects pairs of interactive features at each time in terms of multi-information theory. We use the elastic net for features between different groups, which encourages a grouping effect and tends to select strongly correlated features as a group together. We present a new online group streaming feature selection method that considers feature interaction within and between streaming groups.

4.1. Problem formalization

Let $OGSFS = (G, h, t)$ denote a online group streaming feature selection framework, where $G = \langle G_1, G_2, \ldots, G_k \rangle$ is the streaming feature set, h is the mapping function from features to classes, and t is the time stamp. $G_i = [f_1, f_2, \ldots, f_m]^T$ is a group of features in G which contains m features. It is worth mentioning that not every G_i needs to contain the same number of features. At each time stamp t, we get a new group of features G_t without knowing the exact number of feature space in advance. The problem of online group streaming feature selection aims to select an optimal feature subset S by using groups of features that have arrived so far when the algorithm terminates.

We use the data in Table 1 as an example to illustrate the feature interaction issue during streaming feature selection. For individual streaming feature selection, f_2 and f_3 are arriving one by one at each time and will be discarded one by one for null relevance to D. We suppose f_2 and f_3 in the same group for group streaming feature selection. If the group streaming feature selection method considers the group's features individually, both f_2 and f_3 will not be selected. Thus, we need a new method that can consider the feature interaction within and between the groups.

4.2. Our new algorithm

Our new online group streaming feature selection method can be divided into the intra-group selection and inter-group selection.

4.2.1. Intra-group selection

Suppose $F = \{f_1, f_2, \ldots, f_m\}$ contains $m (m > 2)$ features. For a big feature set F (m is a large value), according to Definition 1, it is hard to know whether a feature $f (f \in F)$ is strongly relevant to class D, because we cannot test all the subsets in $F \setminus \{f\}$.

For the new arriving streaming feature group G_i which contains m features on a feature space $F = \{f_1, f_2, \ldots, f_m\}$, suppose the selected feature subset is S^* which is initialized to $\{\}$ at first. At each iteration i, we select pairs of features $\{f_i, f_i'\}$ which satisfied $I(f_i; f_i'; D) < 0$. Thus, $S^i = \{f_1, f_1', f_2, f_2', \ldots, f_i, f_i'\}$.

Theorem 6. *Features in S^i are relevant to D.*

Proof. For each feature $f' \in S^i$, there exists feature f'' which satisfied $I(f'; f''; D) < 0$. We make $S' = \{f''\}$, then according to Theorem 4, f' is relevant to class D. Thus, the features in S^i are relevant to class D.

According to Theorem 6, if we select a pair of features f' and f'' which satisfies $I(f'; f''; D) < 0$ at each iteration, both f' and f'' are relevant to class D.

In addition to considering feature interaction, we also need to select a compact feature subset. Thus, for features f_i and f_j, if

$$I(f_i; D) > I(f_j; D) \quad \& \quad I(f_i; f_j; D) > 0 \tag{5}$$

which means f_i contains more information about D than that of f_j, and f_i and f_j have a redundancy on D. Then, we can keep f_i and discard f_j.

Meanwhile, the main target of our new method is to select a feature subset that can maintain maximal information about the decision class. If

$$I(f_i; f_j; D) < 0 \quad \& \quad I(S^t \cup f_i \cup f_j; D) > I(S^t; D) \tag{6}$$

which means there is an interaction between f_i and f_j on D. Meanwhile, the adding of f_i and f_j can increase the information of selected feature subset, then both f_i and f_j will be considered to be added into the candidate feature subset.

Besides, for feature f_i, if there is no feature f_j satisfying Eq. (6), but satisfying Eq. (7)

$$I(S^t \cup f_i; D) > I(S^t; D) \tag{7}$$

which means the adding of f_i can increase the information of the candidate feature subset, we will select f_i too.

Based on these, we proposed the new intra-group streaming feature selection algorithm, named $OGSFS - FI_{intra}$, as shown in Algorithm 1.

Algorithm 1 $OGSFS - FI_{intra}$

Require:
F: the condition attribute set in streaming group G_t;
D: the decision class of G_t;
Ensure:
S_t: the selected feature subset of G_t;
1: $S_t = \{\}$;
2: While (F is not empty)
3: find feature f_i in F with the maximal value of $I(f_i; D)$;
4: $INT = \{\}, F = F - \{f_i\}$;
5: For each feature f_j in F
6: If f_j satisfies Eq. (5) Then $F = F - \{f_j\}$;
7: If f_j satisfies Eq. (6) Then $F = F - \{f_j\}, INT = INT \cup \{f_j\}$;
8: End For
9: IF INT is not Empty
10: $S_t = S_t \cup \{f_i\}$;
11: For each feature f' in INT
12: If f' satisfies Eq. (7) Then $S_t = S_t \cup \{f'\}$;
13: End For
14: Else
15: If f_i satisfies Eq. (7) Then $S_t = S_t \cup \{f_i\}$;
16: End If
17: End While
18: **return** S_t;

At timestamp t, suppose the new arriving group is G_t and we let F denote all the features in G_t. At step 3, we find the feature f_i in S with the maximal value of $I(f_i; D)$. For each feature $f_j \in S$, we calculate the multi-information of $I(f_i; f_j; D)$. At step 6, if f_j satisfies Eq. (5), which means f_j has a redundant with f_i. Then, f_j will be removed from S. At step 7, if f_j has an interaction with f_i and can increase the candidate feature subset information, f_j will be added into INT. From Step 9 to Step 16, if INT is not empty, we find all the features which interacted with f_i and can increase the candidate feature subset information. Otherwise, if INT is empty, we will consider selecting a single feature f_i at Step 15. When there are no more features in F, the algorithm will terminate and return the selected feature set S^t.

Let us use the data in Table 1 to illustrate our algorithm.

Suppose all the features in Table 1 in the same streaming group. First of all, we initialize $F = \{f_1, f_2, f_3, f_4\}$, $S_t = \{\}$. Then

we calculate each feature's Mutual Information value in F as: $I(f_1, D) = 0.3113$, $I(f_2; D) = 0$, $I(f_3; D) = 0$, $I(f_4; D) = 0.3113$. Both f_1 and f_4 have the maximal value. Thus, we can select f_1 or f_4 as the first feature. Let us select f_1 as the first feature. Then, $S_t = \{f_1\}$ and $F = \{f_2, f_3, f_4\}$. For each feature in S, $I(f_1; f_2; D) = 0$, $I(f_1; f_3; D) = 0$, $I(f_1; f_4; D) = 0.3113$. Thus, f_1 and f_4 are redundant with each other, and we remove f_4 from S. For there are no features interact with f_1, we calculate $I(f_1; Y) = 0.3113 > 0$. Thus, $S_t = \{f_1\}$, $F = \{f_2, f_3\}$.

Then, we consider f_2 as the next candidate feature. $S_t = \{f_1, f_2\}$ and $F = \{f_3\}$. For $I(f_2; f_3; D) = -0.3113$ and $I(\{f_1, f_2, f_3\}; D) = 0.8113 > I(f_1; Y)$, then $INT = INT \cup \{f_3\} = \{f_3\}$. For there is only one feature f_3 in INT, thus, $S_t = \{f_1, f_2, f_3\}$, $F = \{\}$.

Therefore, the final selected feature subset is $\{f_1, f_2, f_3\}$.

4.2.2. Inter-group selection

This section introduces the inter-group selection algorithm, which aims to obtain a group of features on global group information. We propose to solve this problem with the regularization and variable selection method, elastic net [31].

Given the selected features in the intra-group selection step $S = \{G'_1, G'_2, \ldots, G'_t\}$, where $G'_i = \{f_{i1}, f_{i2}, \ldots, f_{ik}\}$. Suppose $X = S \in R^{m' \times n}$ denotes the condition feature set, $y \in R^n$ denotes the class label vector, $\beta = [\beta_1, \beta_2, \ldots, \beta_m]$ denotes the projection vector. The naive elastic net criterion is defined as:

$$L(\lambda_1, \lambda_2, \beta) = \|y - X\beta\|_2 + \lambda_1\|\beta\|_2 + \lambda_2\|\beta\|_1 \tag{8}$$

where $\| \sim \|_2$ stands for l_2 norm, and $\| \sim \|_1$ stands for l_1 norm of a vector.

The naive elastic net estimator $\hat{\beta}$ is the minimizer of Eq. (8):

$$\hat{\beta} = arg\ min_\beta\{L(\lambda_1, \lambda_2, \beta)\}. \tag{9}$$

For an $\alpha \in (0, 1]$, and a nonnegative λ, elastic net can be rewrite as:

$$min_\beta\{\|y - X\beta\|_2 + \lambda P_\alpha(\beta)\} \tag{10}$$

where $P_\alpha(\beta) = \frac{(1-\alpha)}{2}\|\beta\|_2 + \alpha\|\beta\|_1$.

Elastic net is the same as lasso when $\alpha = 1$. As α shrinks toward 0, elastic net approaches simple ridge regression.

In the optimization methods, the value of λ is usually determined by cross-validation. By setting several β_i to 0, the features corresponding to non-zero coefficients will be selected. With the simultaneously automatic variable selection and continuous shrinkage, the elastic net can select groups of correlated variables.

4.2.3. OGSFS-FI

Our new online group streaming feature selection algorithm (OGSFS-FI) is shown as Algorithm 2. OGSFS-FI is divided into two components: intra-group selection and inter-group selection. Details are as follows.

4.3. Time complexity of OGSFS-FI

Suppose the dataset is $\mathbb{D}$, the number of instances in $\mathbb{D}$ is n, the number of features in $\mathbb{D}$ is m, and the number of features in each group is m_G. At timestamp t, suppose the selected feature subset is S^{t-1} and the new arriving streaming group is $G_t(|G_t| = m_G)$.

For intra-group selection, each feature in G_t will be check once from step 5 to step 8, and the worst case of this stage is $O(|G_t|)$. From step 9 to step 16, if INT is not empty, this stage's worst time complexity is $O(|G_t|)$. Thus, the worst time complexity of intra-group selection is $O(|G_t|)$. Suppose the selected feature subset for G_t is S^t and $S = S^{t-1} \cup S^t$. For elastic net, if we stop the algorithm after k steps, then it requires $O(k^3 + |S| * k^2)$ operations.

Algorithm 2 *OGSFS − FI*

Require:
G: the condition streaming feature set;
Ensure:
S: the selected feature subset;
1: $S = \{\}$;
2: **Repeat**
3: $G_t \leftarrow$ get a new streaming group of features;
4: /* online intra-group selection */
5: $S_t = OGSFS - FI_{intra}(G_t)$;
6: $S = S \cup S_t$;
7: /* online inter-group selection */
8: $S \leftarrow$ find the global optimal subset by elastic net algorithm;
9: **Utile no more groups arrive**
10: **return** S;

In total, the worst time complexity of OGSFS-FI is $O(k^3 + |S| * k^2)$. The experimental results in the next section show that the final selected subset's size is tiny for OGSFS-FI. Thus, the time complexity of our method for real-world applications is far less than the worst case. Meanwhile, we set the maximum number of iterations to 1000, and it can converge very fast for some datasets.

5. Experimental results

5.1. Experiment setup

This section applies the proposed group streaming feature selection algorithm on several synthetic datasets and real-world datasets. We compare OGSFS-FI with eight streaming feature selection methods, including: Grafting [20], Alpha-investing [15], OSFS and Fast-OSFS [5], SAOLA and Group-SAOLA [9], OS-NRRSARSA [21] and OGFS [8]. The significance level α is set to 0.01 for OSFS, SAOLA, and Group-SAOLA. For Alpha-investing, the parameters are set to the values used in [15]. For OGFS, the parameters of ε and γ are set to 0.001 and 0.45 as the values used in [8]. All algorithms mentioned above are implemented in MATLAB [32], where OS-NRRSARA-SA is implemented by ourselves.

We use three basic classifiers, KNN(k=3), SVM, and CART in Matlab R2015b, to evaluate a selected feature subset in our experiments. We perform 5-fold cross-validation on each dataset. Feature selection is training on 4/5 data samples and testing on the rest 1/5 data. All competing algorithms use the same training and testing data for each fold. All experimental results are conducted on a PC with AMD(R) 3700X, 3.6 GHz CPU, and 32 GB memory.

To validate whether OGSFS-FI and its rivals have a significant difference in prediction accuracy, we conduct the Friedman test at a 95% significance level [33], under the null-hypothesis. If the null-hypothesis at the Friedman test is rejected, we proceed with the Nemenyi test [33] as a post-hoc test. Besides, the win/tie/loss (W/T/L for short) counts are summarized for the experiment results.

5.2. Experimental on synthetic data sets

To demonstrate the features selected by our new method are interactive with each other, we apply OGSFS-FI on six synthetic datasets with all the irrelevant, redundant, and interactive features known in advance.

Table 2
Synthetic datasets.

Data set	Instances	Features	Target concept c
Data1	100	10	$c = (\bar{a}_2 \wedge a_7) \vee (a_1 \wedge a_6)$
Data2	100	10	$c = \bar{a}_6 \vee (\bar{a}_2 \wedge \bar{a}_7 \wedge \bar{a}_9)$
Data3	100	5	$c = (a_1 \wedge \bar{a}_3 \wedge \bar{a}_5) \vee (a_2 \wedge a_3) \vee a_3$;
MONK1	432	6	$c = (a_1 = a_2) \vee (a_5 = 1)$
MONK2	432	6	c =two of $\{a_1 = 1, a_2 = 1, \ldots, a_6 = 1\}$
MONK3	432	6	$c = (a_5 = 3 \wedge a_4 = 1) \vee (a_5 \neq 4 \wedge a_2 \neq 3)$

Table 3
Features selected on the six synthetic datasets.

Methods	Data1	Data2	Data3	Monk1	Monk2	Monk3
OGSFS-FI	$\{a_2, a_7\}$	$\{a_2, a_6, a_7\}$	$\{a_2, a_3\}$	$\{a_1, a_2, a_5\}$	$\{a_1 \sim a_6\}$	$\{a_2, a_4, a_5\}$
Grafting	$\{a_1 \sim a_{10}\}$	$\{a_1 \sim a_{10}\}$	$\{a_1 \sim a_5\}$	$\{a_1 \sim a_6\}$	$\{a_1 \sim a_6\}$	$\{a_1 \sim a_6\}$
Investing	$\{a_1, a_3, a_4, a_6, a_7\}$	$\{a_1 \sim a_7, a_{10}\}$	$\{a_1, a_2, a_3\}$	$\{a_1 \sim a_6\}$	$\{a_1, a_2, a_3\}$	$\{a_1, a_2, a_5\}$
OSFS	$\{a_2, a_7\}$	$\{a_2, a_6\}$	$\{a_1, a_3\}$	$\{a_5\}$	{}	$\{a_2, a_5\}$
Fast-OSFS	$\{a_2, a_7\}$	$\{a_2, a_6\}$	$\{a_1, a_3\}$	$\{a_5\}$	{}	$\{a_2, a_5\}$
SAOLA	$\{a_1, a_2, a_5, a_6, a_7\}$	$\{a_2, a_6\}$	$\{a_1, a_2, a_3\}$	$\{a_2, a_5\}$	$\{a_4, a_5, a_6\}$	$\{a_2, a_4, a_5, a_6\}$
OS-SA	$\{a_1, a_2, a_6, a_7, a_9\}$	$\{a_2, a_6, a_7, a_9, a_{10}\}$	$\{a_1 \sim a_5\}$	$\{a_1, a_2, a_5, a_6\}$	$\{a_1 \sim a_6\}$	$\{a_2, a_4, a_5, a_6\}$

5.2.1. Synthetic data sets

The first three datasets, Data1, Data2, and Data3, are generated by the data generation tool RDG1 of the data mining toolkit WEKA. The other three datasets about MONKs problems are available from the UCI Machine Learning Repository. The six datasets are described as follows:

For each dataset, the features appearing in the target concept's definition are all relevant, while the absent features are either redundant or irrelevant. The conjunctive terms in the definition of the target concept imply interactive features. For example, in dataset Data1, there are ten features $a_1, \ldots, a_{10}$ and $c = (\bar{a}_2 \wedge a_7) \vee (a_1 \wedge a_6)$. This means a_2, a_7, a_1, a_6 are relevant features, and the others are redundant or irrelevant features. Meanwhile, $\{a_2, a_7\}$ and $\{a_1, a_6\}$ are both interactive features. Besides, for MONK3, 5% class noise was added to the training set.

5.2.2. Results on the synthetic data sets

We apply OGSFS-FI and other six streaming feature selection methods, including Grafting [20], Alpha-investing [15], OSFS [5], Fast-OSFS [5], SAOLA [9] and OS-NRRSAR-SA [21] on these six synthetic datasets. All these competing algorithms do not consider the feature interaction during feature selection. For datasets MONK1, MONK2, and MONK3, we use 1/2 samples for training and the rest for testing. The results of selected features for each algorithm can be seen in Table 3.

From Tables 2 and 3, we observe that OGSFS-FI can select the interactive features from datasets, while SAOLA, Grafting, Alpha-investing, and OS-NRRSAR-SA select more features compared to our new method. This indicates that they cannot discriminate irrelevant and redundant features well. Meanwhile, OSFS and Fast-OSFS select fewer features that indicate they cannot select interactive features from datasets.

5.3. Experiments on real world data sets

In this subsection, we will apply our new method on ten real-world datasets, including five DNA microarray datasets (PROSTATE, LEUKEMIA, COLON, DLBCL, BREAST), three NIPS 2003 datasets (MADELON, ARCENE, GISETTE,) and two WCCI 2006 Performance Prediction Challenge datasets (GINA, HIVA), as shown in Table 4. For continuous value datasets, the values of each feature are discretized into ten equal intervals.

Table 4
Real-world datasets.

Index	Data set	Instances	Features
1	PROSTATE	102	6033
2	LEUKEMIA	72	7129
3	COLON	62	2000
4	DLBCL	77	7129
5	MADELON	2600	500
6	ARCENE	200	10000
7	GISETTE	7000	5000
8	BREAST	97	24481
9	SRBCT	63	2308
10	HIVA	4229	1617

5.3.1. Analysis of parameter alpha and group sizes

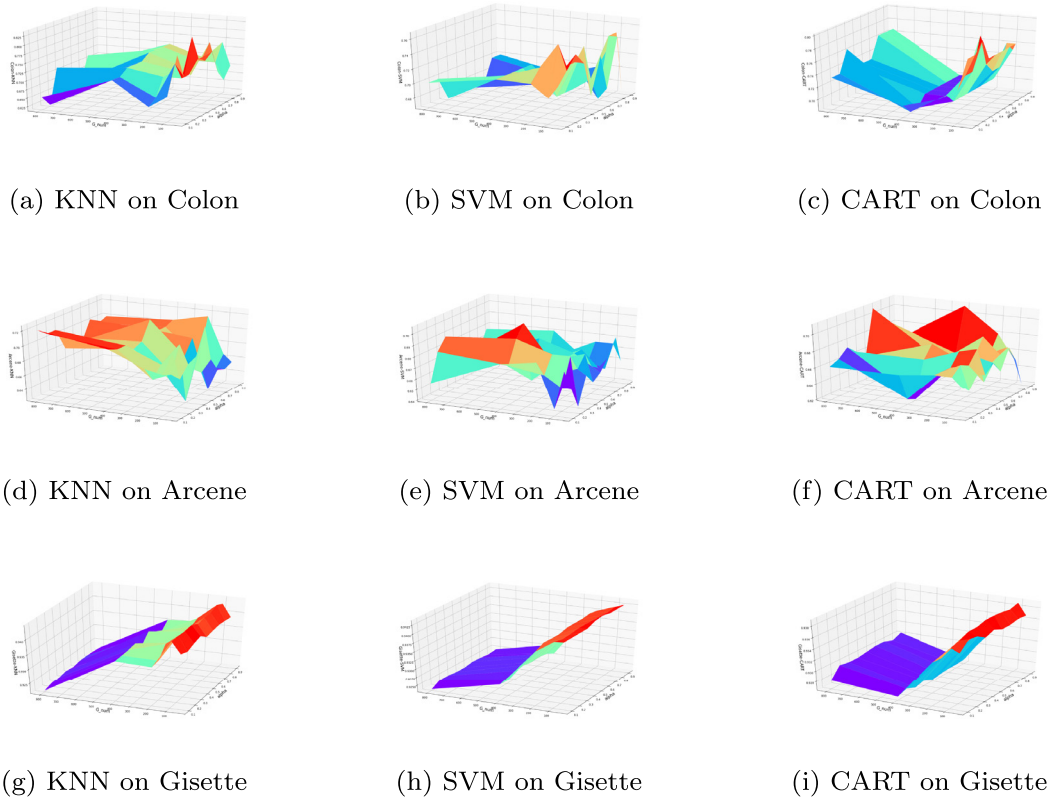
There are two parameters: α and λ, in the inter-group selection component of OGSFS-FI. In this section, we test different values of α and analyze the influence of it in our new method. For λ, we determine the value of it by cross-validation with the minimum MSE (Mean Square Error). We choose the values of α from 0.1 to 0.9 with 0.1 intervals.

Meanwhile, for the datasets mentioned above do not have known group structures, we try to apply our new method on these datasets with different group sizes and analyze the influence of group size in our new method. We choose the group sizes of 50, 100, 200, 400, and 800.

Due to space constraints, we use three datasets: Colon, Arcene, and Gisette to test the performance of different values of α and group sizes. The experimental results of predictive accuracy, running time and the mean number of selected features can be seen from Fig. 2 to Fig. 4.

From Figs. 2–4, we can observe that:

- For a specific value of group size, the predictive accuracy highly depends on α for datasets Colon and Arcene. Meanwhile, on dataset Gisette, all different values of α almost get the same performance. According to Fig. 4, the alpha values seriously affect the final number of selected features for datasets Colon and Arcene, while having a minor impact for Gisette. The fundamental reason for this phenomenon is the relationship between the features in each dataset. In other words, there is nearly no group effecting on dataset Gisette, no matter what the group sizes we specified.
- For a specific value of α, the performance of different values of group sizes varies greatly. For dataset Colon, a small group size performances better than others. For dataset Arcene, big group size performances better than others. Except for

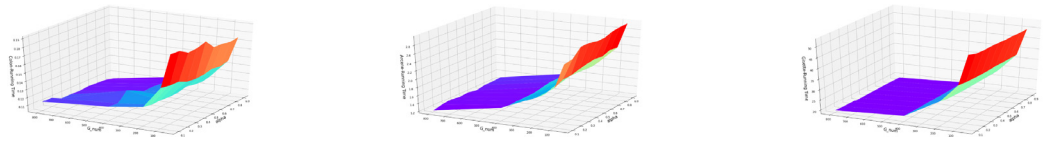


(a) KNN on Colon (b) SVM on Colon (c) CART on Colon

(d) KNN on Arcene (e) SVM on Arcene (f) CART on Arcene

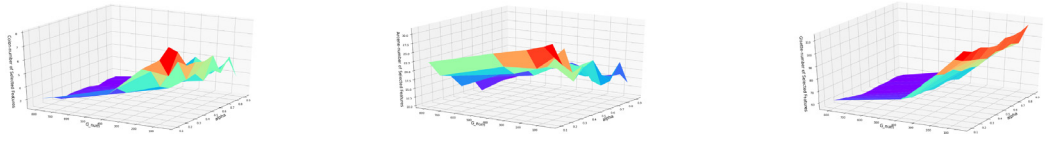
(g) KNN on Gisette (h) SVM on Gisette (i) CART on Gisette

Fig. 2. Predictive accuracy varying with different values of α and group sizes.



(a) Colon (b) Arcene (c) Gisette

Fig. 3. Running time varying with different values of α and group sizes.



(a) Colon (b) Arcene (c) Gisette

Fig. 4. Mean number of selected features varying with different values of α and group sizes.

those no group effecting datasets, such as Gistette, the final performance of our new algorithm determined by both the values of α and group size. Smaller group sizes prefer to select more features during intra-group selection, then for inter-group selection, a bigger value of α performs better. On the other side, bigger group sizes prefer to select fewer features during intra-group selection, while a smaller value of α tends to perform better in inter-group selection. Thus, for predictive accuracy, the selection of these two parameter values is related. A smaller group size indicates more consideration of feature interaction within feature groups, while a smaller value of alpha means more feature interaction between feature groups.

- On running time, the performance between all these different values of α is exceedingly small. Meanwhile, the running time increases as the group size decreases. Smaller group sizes will lead to more calls of the intra-group selection and inter-group selection. Thus, it will spend more running time in total with a small group size.
- Besides, on the mean number of selected features, smaller α and group sizes select more features than others on average. The main reason is the symmetrical uncertainty criteria we used for stopping the selection in our method. When the group size is smaller, it tends to select more features to satisfy each group's symmetrical uncertainty constraint and consume more time.

Table 5
The KNN predictive accuracy of OGSFS-FI VS. competing algorithms.

Dataset	OGSFS-FI	Alpha-investing	OSFS	SAOLA	OGFS	Group-SAOLA
PROSTATE	**0.876**	0.796	0.616	0.812	0.492	0.81
LEUKEMIA	**0.8686**	0.7771	0.6486	0.84	0.62	0.8029
COLON	**0.82**	0.7067	0.5767	0.6333	0.5867	0.6367
DLBCL	**0.912**	0.8187	0.7547	0.7707	0.3253	0.7733
MADELON	**0.5958**	0.5437	0.5526	0.5247	0.4978	0.5247
ARCENE	**0.7075**	0.7	0.565	0.595	0.52	0.575
GISETTE	0.9327	**0.9635**	0.5939	0.89	0.4978	0.8903
BREAST	**0.6263**	0.6105	0.5421	0.5421	0.5	0.5421
SRBCT	**0.8159**	0.694	0	0.6411	0.3799	0.6103
HIVA	0.9664	0.9645	0.9648	**0.9671**	0.9646	**0.9671**
W/T/L	8/0/2	1/0/9	0/0/10	1/0/9	0/0/10	1/0/9
AVG.	**0.8121**	0.7574	0.5814	0.7216	0.5384	0.7132
AVG. RANKS	**1.3**	2.8	4.7	3.2	5.7	3.3

Table 6
The SVM predictive accuracy of OGSFS-FI VS. competing algorithms.

Dataset	OGSFS-FI	Alpha-investing	OSFS	SAOLA	OGFS	Group-SAOLA
PROSTATE	**0.884**	0.848	0.646	0.876	0.582	0.866
LEUKEMIA	**0.9057**	0.7371	0.68	0.8314	0.6571	0.7943
COLON	**0.82**	0.7167	0.5967	0.6633	0.6333	0.66
DLBCL	**0.9147**	0.7813	0.768	0.8187	0.7547	0.8213
MADELON	0.5715	0.5872	**0.5982**	0.5883	0.4845	0.5883
ARCENE	0.6525	**0.7425**	0.5425	0.6275	0.56	0.6
GISETTE	0.9348	**0.97**	0.6064	0.8953	0.4884	0.8905
BREAST	**0.6105**	0.6053	0.4789	0.5895	0.5579	0.5895
SRBCT	**0.8364**	0.7242	0	0.635	0.3494	0.6042
HIVA	0.9651	0.9411	0.9643	**0.9663**	0.9648	**0.9663**
W/T/L	6/0/4	2/0/8	1/0/9	1/0/9	0/0/10	1/0/9
AVG.	**0.8095**	0.7653	0.5881	0.7491	0.6032	0.7380
AVG. RANKS	**1.8**	3.0	5.0	2.65	5.4	3.15

Table 7
The CART predictive accuracy of OGSFS-FI VS. competing algorithms.

Dataset	OGSFS-FI	Alpha-investing	OSFS	SAOLA	OGFS	Group-SAOLA
PROSTATE	**0.838**	0.806	0.626	0.826	0.602	0.796
LEUKEMIA	**0.8886**	0.7057	0.5914	0.8086	0.56	0.7829
COLON	**0.7567**	0.64	0.57	0.6133	0.6033	0.63
DLBCL	**0.872**	0.7947	0.7093	0.792	0.7547	0.7787
MADELON	0.5887	**0.7174**	0.5418	0.6122	0.5054	0.6122
ARCENE	**0.665**	0.6625	0.58	0.6525	0.59	0.605
GISETTE	0.9181	**0.9441**	0.5943	0.8963	0.4886	0.8964
BREAST	**0.6105**	0.5211	0.5579	0.5842	0.5737	0.5842
SRBCT	**0.7944**	0.6727	0	0.621	0.367	0.581
HIVA	0.9652	0.9547	0.9643	0.9677	0.9648	**0.9679**
W/T/L	7/0/3	2/0/8	0/0/10	0/0/10	0/0/10	1/0/9
AVG.	**0.7897**	0.7418	0.5735	0.7373	0.6009	0.7234
AVG. RANKS	**1.6**	2.9	5.4	2.8	5.2	3.1

Table 8
The running time(s) of OGSFS-FI VS. competing algorithms.

Dataset	OGSFS-FI	Alpha-investing	OSFS	SAOLA	OGFS	Group-SAOLA
PROSTATE	1.0929	0.6869	4.6988	0.227	0.0005	0.4212
LEUKEMIA	0.8375	1.0327	5.5542	0.2564	0.0001	0.5035
COLON	0.361	0.0611	1.4779	0.0662	0.0001	0.1409
DLBCL	1.0134	0.6746	4.7218	0.2224	0.0001	0.4384
MADELON	2.3293	0.4316	0.4624	0.0242	0.0002	0.0239
ARCENE	16.9252	1.2248	8.8917	0.4337	0.0001	0.6618
GISETTE	190.8926	218.3727	93.9858	2.0754	0.0004	2.3962
BREAST	3.8581	3.8069	17.2513	0.9009	0.0002	1.7095
SRBCT	1.1172	0.0741	1.8685	0.0791	0.0001	0.1644
HIVA	2.321	27.6255	1.8818	0.2637	0.0002	0.3183
AVG.	22.0748	25.3990	14.0794	0.4549	0.0002	3.4811
AVG. RANKS	4.5	3.7	5.0	2.2	1.0	4.6

In general, for some datasets that have very weak group effecting, all different values of α almost get the same performance no matter what the group sizes we specified. For other datasets, the selection of these two parameter values is related and varies greatly. On running time and number of selected features, the values of group sizes have a more significant effect than α. Smaller group sizes tend to select more features and consume more running time.

Table 9
The mean number of selected feature of OGSFS-FI VS. competing algorithms.

Dataset	OGSFS-FI	Alpha-investing	OSFS	SAOLA	OGFS	Group-SAOLA
PROSTATE	16.28	19.2	1	7.24	1	5.6
LEUKEMIA	12.44	20.12	1	1.44	1	1.04
COLON	5.32	5.96	1	1.28	1	1
DLBCL	15.8	16.56	1	9.4	1	5.2
MADELON	3.56	57.36	1.88	1	1	1
ARCENE	6.5	15.1	1	16.2	1	9
GISETTE	92.24	416.84	2	19.52	1	16.2
BREAST	14.8	11.9	1	1	1	1
SRBCT	17.6	7.84	0	1.92	1	1.8
HIVA	17.28	295	1	6.68	1	4.4
AVG.	20.1	86.5	1.0	6.5	1	4.6
AVG. RANKS	5.0	5.7	1.9	3.85	1.7	2.85

5.3.2. OGSFS-FI vs. Other streaming feature selection methods

We compare our algorithm with three state-of-the-art individual streaming feature selection methods (Alpha-investing [15], OSFS [5], and SAOLA [9]) and two group streaming feature selection algorithms (Group-SAOLA [9] and OGFS [8]). For OGFS and Group-SAOLA, we set the group size(G=100) as the same as OGSFS-FI .

Tables 5–9 summarize the predictive accuracy, running time and mean number of selected features of GSFS-FI against the other five algorithms. We report the best performance for each dataset. The p-values of Friedman test on KNN, SVM, CART, running time and the mean number of selected features are 5.4478e−10, 3.3444e−07, 5.5492e−09, 2.8285e−11, and 1.8496 e−13 respectively. Thus, there is a significant difference among these competing algorithms on predictive accuracy, running time and number of selected features respectively. According to the Nemenyi test, the value of CD (critical difference) is 2.3841.

From Table 5 to Table 9 , we have the following observations.

- OGSFS-FI vs. Alpha-investing. According to the average ranks and the value of critical difference, there is no significant difference between OGSFS-FI and Alpha-investing on predictive accuracy in KNN, SVM, and CART cases. However, OGSFS-FI performs better than Alpha-investing with KNN, SVM, and CART on average. Alpha-investing spends more running time on average than OGSFS-FI. Meanwhile, Alpha-investing selects many more features than our new algorithm. In general, compared with Alpha-investing, OGSFS-FI can select fewer features but performs better.
- OGSFS-FI vs. OSFS. There is a significant difference between OGSFS-FI and OSFS on predictive accuracy in KNN, SVM, and CART cases. Thus, OGSFS-FI performs significantly better than OSFS on predictive accuracy. OGSFS-FI is competing with OSFS on running time and selects much more features than OSFS. OSFS considers features individually and only selects one or two features on some datasets, which leads to the loss of much important information.
- OGSFS-FI vs. SAOLA. There is no significant difference between OGSFS-FI and SAOLA on predictive accuracy. However, OGSFS-FI outperforms SAOLA on average in cases of these three classifiers. SAOLA is faster than OGSFS-FI and selects fewer features. Similar to OGSFS-FI, SAOLA also uses Mutual Information for feature selection but does not consider the interaction between features. Thus, OGSFS-FI selects more features than SAOLA and performs better.
- OGSFS-FI vs. OGFS. There is a significant difference between OGSFS-FI and OGFS on predictive accuracy. OGFS is an online streaming group feature selection method that considers the underlying structure of the feature stream. However, OGFS only selects one or two features on these datasets and performs worst among these comparing algorithms. There are two probable reasons: 1) we do not find good parameter values for OGFS; 2) OGFS is proposed for image analysis and maybe not fit these datasets.
- OGSFS-FI vs. Group-SAOLA. There is no significant difference between OGSFS-FI and Group-SAOLA on predictive accuracy. However, OGSFS-FI performs better than Group-SAOLA on average. Group-SAOLA runs faster than OGSFS-FI and selects fewer features. Group-SAOLA is an online streaming group feature selection method and runs fast on extremely high dimension datasets. However, Group-SAOLA does not consider the feature interaction within streaming groups and selects much fewer features on these datasets, which leads to the loss of important information.

In sum, OGSFS-FI gets the highest average predictive accuracy and lowest average ranks in KNN, SVM, and CART cases. With the considering of feature interactions, OGSFS-FI selects more features and spends more running times. However, OGSFS-FI performs best on average among these competing algorithms.

6. Conclusion

In this paper, considering the feature interaction within and without streaming group features, we proposed a new online group streaming feature selection method to handle it, name OGSFS-FI. OGSFS-FI can be divided into two parts: online intra-group selection and online inter-group selection. For online intra-group selection, we designed a new approach that selects pairs of interactive features in multi-information theory. Meanwhile, we demonstrated that features in such a selected subset are relevant to the class and interact with each other. For inter-group selection, we used the elastic net that prefers to select a group of features. Experiments conducted on six synthetic datasets and ten real-world datasets indicated the effectiveness of our new method. In future work, we will deeply analyze the foundations of feature interaction and why it can improve prediction accuracy.

CRediT authorship contribution statement

Peng Zhou: Conceptualization, Methodology, Software, Writing - original draft, Funding acquisition. **Ni Wang:** Software, Validation, Investigation, Writing - review & editing. **Shu Zhao:** Formal analysis, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under grants 61906056, 61876001.

References

[1] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, Acm Comput. Surv. 50 (6) (2017) 1–45.

[2] L. Sun, L. Wang, W. Ding, Y. Qian, J. Xu, Feature selection using fuzzy neighborhood entropy-based uncertainty measures for fuzzy neighborhood multigranulation rough sets, IEEE Trans. Fuzzy Syst. (2020) 1, http://dx.doi.org/10.1109/TFUZZ.2020.2989098.

[3] 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.

[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. (TIST) 2 (4) (2011) 1–22.

[5] 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.

[6] H.G. Li, X.D. Wu, Z. Li, W. Ding, Group feature selection with streaming features, in: IEEE 13th International Conference on Data Mining, 2013, pp. 1109–1114.

[7] X. Hu, P. Zhou, P. Li, J. Wang, X. Wu, A survey on online feature selection with streaming features, Front. Comput. Sci. 12 (3) (2018) 479—493.

[8] J. Wang, M. Wang, P. Li, L. Liu, Z. Zhao, X. Hu, X. Wu, Online feature selection with group structure analysis, IEEE Trans. Knowl. Data Eng. 27 (2015) 3029–3041.

[9] 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) 1–39.

[10] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: Eleventh International Conference on Machine Learning, 1994, pp. 121–129.

[11] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324.

[12] A. Jakulin, I. Bratko, Analyzing attribute dependencies, Lecture Notes in Comput. Sci. 2838 (2003) 229–240.

[13] Z. Zeng, H. Zhang, R. Zhang, Y. Zhang, A novel feature selection method considering feature interaction, Pattern Recognit. 48 (8) (2015) 2656–2666.

[14] J.R. Vergara, P.A. Estévez, A review of feature selection methods based on mutual information, Neural Comput. Appl. 24 (1) (2014) 175–186.

[15] J. Zhou, D.P. Foster, R.A. Stine, L.H. Ungar, Streamwise feature selection, J. Mach. Learn. Res. 3 (2) (2006) 1532–4435.

[16] A. Jakulin, I. Bratko, Testing the significance of attribute interactions, in: The 21st International Conference on Machine Learning, 2004, pp. 52–60.

[17] L. Sun, L. Wang, W. Ding, Y. Qian, J. Xu, Neighborhood multi-granulation rough sets-based attribute reduction using Lebesgue and entropy measures in incomplete neighborhood decision systems, Knowl.-Based Syst. 192 (2020) 105373, http://dx.doi.org/10.1016/j.knosys.2019.105373.

[18] W. Shu, W. Qian, Y. Xie, Incremental feature selection for dynamic hybrid data using neighborhood rough set, Knowl.-Based Syst. 194 (2020) http://dx.doi.org/10.1016/j.knosys.2020.105516.

[19] Z. Zhao, H. Liu, Searching for interacting features in subset selection, Intell. Data Anal. 13 (2) (2009) 207–228.

[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] S. Eskandari, M. Javidi, Online streaming feature selection using rough sets, Internat. J. Approx. Reason. 69 (C) (2016) 35–57.

[22] M.M. Javidi, S. Eskandari, Online streaming feature selection: a minimum redundancy, maximum significance approach, Pattern Anal. Appl. 22 (2019) 949–963.

[23] P. Zhou, X. Hu, P. Li, X. Wu, Online feature selection for high-dimensional class-imbalanced data, Knowl.-Based Syst. 136 (2017) 187–199.

[24] P. Zhou, X. Hu, P. Li, X. Wu, Online streaming feature selection using adapted neighborhood rough set, Inform. Sci. 481 (2019) 258–279.

[25] M. Rahmaninia, P. Moradi, OSFSMI: Online stream feature selection method based on mutual information, Appl. Soft Comput. 68 (2018) 733–746.

[26] C.E.A. Shannon, A mathematical theory of communication, Acm Sigmob. Mob. Comput. Commun. Rev. 5 (1) (2001) 3–55.

[27] G. Bontempi, P.E. Meyer, Causal filter selection in microarray data, in: Proceedings of International Conference on Machine Learning, 2010, pp. 95–102.

[28] M. Hall, M. Hall, Benchmarking attribute selection techniques for discrete class data mining, IEEE Trans. Knowl. Data Eng. 15 (6) (2003) 1437–1447.

[29] J.M. Peña, Learning Gaussian graphical models of gene networks with false discovery rate control, in: European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBIO'08), 2008, pp. 165–176.

[30] L. Yu, H. Liu, Eficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (12) (2004) 1205–1224.

[31] H. Zou, T. Hastie, Regularization and variable selection via the elastic-net, J. R. Stat. Soc. 67 (2) (2005) 301–320.

[32] K. Yu, W. Ding, X. Wu, LOFS: Library of online streaming feature selection, Knowl.-Based Syst. 113 (2016) 1–3.

[33] J. Demˇ sar, Statistical comparisons of classifiers over multiple data sets., J. Mach. Learn. Res. 7 (1) (2006) 1–30.