Applied Soft Computing 68 (2018) 733–746

Contents lists available at ScienceDirect

Applied Soft Computing

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

OSFSMI: Online stream feature selection method based on mutual information

Maryam Rahmaninia [a], Parham Moradi [b,*]

[a] Faculty of Engineering, Sanandaj Branch, Islamic Azad University, Sanandaj, Iran
[b] Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran

ARTICLE INFO

Article history:
Received 28 March 2017
Received in revised form 24 July 2017
Accepted 16 August 2017
Available online 24 August 2017

Keywords:
Online streaming feature selection
Mutual information
Dimensionality reduction
Filter method

ABSTRACT

Feature selection is used to choose a subset of the most informative features in pattern identification based on machine learning methods. However, in many real-world applications such as online social networks, it is either impossible to acquire the entire feature set or to wait for the complete set of features before starting the feature selection process. To handle this issue, online streaming feature selection approaches have been recently proposed to provide a complementary algorithmic methodology by choosing the most informative features. Most of these methods suffer from challenges such as high computational cost, stability of the generated results and the size of the final features subset. In this paper, two novel feature selection methods called OSFSMI and OSFSMI-k are proposed to select the most informative features from online streaming features. The proposed methods employ mutual information concept in a streaming manner to evaluate correlation between features and also to assess the relevancy and redundancy of features in complex classification tasks. The proposed methods do not use any learning model in their search process, and thus can be classified as filter-based methods Several experiments are performed to compare the performance of the proposed algorithms with the state-of-the-art online streaming feature selection methods The reported results show that the proposed methods performs better than the others in most of the cases.

1. Introduction

Rapid improvement of storage and information processing technologies has led to appearance of large-scale datasets with large number of patterns and features [1]. The presence of high dimensional data – known as the curse of the dimensionality problem – reduces the performance of many machine learning methods [2]. A popular approach to tackle this problem is to reduce dimensionality of the feature space [3]. Feature selection is a well-known and effective dimensionality reduction approach that aims at selecting a parsimonious feature subset by identifying and eliminating those of redundant and irrelevant features.

Up to now, many feature selection methods have been proposed to improve the interpretability, efficiency and accuracy of the learning models. Most of these methods require to access the entire feature set to perform their search process [4–11]. However, in many real-world applications it is either impossible to acquire the entire data or it is impractical to wait for the complete data before feature selection starts [12–15]. In other words, in these types of applications, data arrives sequentially and novel features or instances may appear incrementally. For example, in online social networks such as Twitter, in the case of presenting a new hot topic, a set of new keywords appears which leads to increase the dimensionality of the data over time. Traditional feature selection methods need to load the entire training dataset in the memory, which leads to exceeding the memory capacity for many real-world applications. These limitations make the traditional batch feature selection techniques impractical for emerging big data applications. To overcome these problems, online streaming feature selection methods (OSF) have been recently proposed to provide a complementary algorithmic methodology to addresses high dimensionality in big data analytics by choosing the most informative features [15–18].

Considering the fact that the whole data is unavailable, a successful OSF method needs an efficient incremental update rule in its search process. To this end, several methods have been recently proposed to select a best feature subset from online data streams. These methods can be classified into two categories: instance-based and feature-based OSF methods. In instance-based OSF methods, the number of instances increases over the time, while the number of features is assumed to be fixed [16,19–21]. This type of methods can be employed in some applications such

* Corresponding author.
E-mail addresses: ma.rahmaninia@gmail.com (M. Rahmaninia), p.moradi@uok.ac.ir (P. Moradi).

https://doi.org/10.1016/j.asoc.2017.08.034
1568-4946/

as traffic network monitoring, financial analysis of stock data streams and Internet query monitoring, where all feature space is available from the beginning but the number of instances increase over time. For example, the method proposed in [22] uses an incremental learning algorithm to select prominent features as new instances arrive. Therefore, the scope of these methods is limited to the problems where all features are given before the learning process. On the other hand, feature-based OSF methods assume that the feature space is unavailable or is infinite before starting the feature selection process [17,23–27]. In some real-world applications, the features are often expensive to generate (e.g., a lab experiment), and thus may appear in streaming manner. Generally, in feature-based OSF methods a criterion is defined to decide whether or not a newly arrived feature should be added to the model. For example, in [23] a statistical analysis is performed to evaluate the importance of a so-far-seen feature. This method requires prior knowledge about the entire feature space. Also, it does not have any strategy to remove those features that later found to be redundant. In [26] a conditional independent test is used to evaluate both relevancy and redundancy of a newly arrived feature. Although this method considers both relevancy and redundancy analysis in its online process, in each iteration it compares a new feature with all subsets of previously selected features, and thus it requires a high computational cost to process a so-far-seen feature. Recently, in [24] the authors proposed a method called SOALA that uses a pairwise mutual information to evaluate the relevance and redundancy of features in an online manner. In this method, deciding to include or ignore the features depends on some predefined parameters that setting their precise value requires to know the whole feature space. Also, the authors of [17] proposed an online feature selection method based on rough set theory to assess the importance of features in an online manner. Although their method does not require any domain knowledge, it only operates effectively with datasets containing discrete values, and therefore it is necessary to perform a discretization step for real-valued attributes. This method also suffers from high computational complexity when dealing with large-scale datasets.

In this paper, two novel feature-based OSF methods are proposed which aims to achieve high classification accuracy with a reasonable running time. We suppose that the features appear incrementally over time while the number of instances is considered to be fixed. The first method selects a prominent subset of feature with the minimal size, while the other method chooses a set of prominent features in constant size k. These methods employ mutual information concept to evaluate the relevancy and redundancy of features without using any learning model. Thus, the proposed methods can be classified into filter-based feature selection methods. A number of mutual information based methods have been successfully applied to feature selection tasks, see a review in [4]. However, most of these algorithms consider batch feature selection problem and cannot be applied for online feature selection scenarios. For example, in [28] mutual information is used to choose relevant features and eliminate redundant ones in an offline manner, and one needs to know the whole feature space at the beginning of the search process. The framework of the proposed methods consists of two steps. In the first step, the relevancy of each newly arrived feature with the target class is evaluated. The relevant features are included to the selected subset and the others are ignored. In the second step, the goal is to identify and eliminate those of ineffective features through several iterations. In this step, a specific strategy is used to eliminate redundant features. Using this strategy, if a previously selected feature is identified as a redundant feature, it is removed from the selected subset. To this end, a measure is introduced to evaluate the effectiveness of a newly arrived feature considering both of relevancy and redundancy concepts. Using this measure, those of ineffective features compared to a so far seen feature are eliminated from the feature set and the process is continued iteratively until there is no more ineffective in the feature set. Using this process, one more chance is given to a new arrived feature to be processed in the further steps. Also, by removing a feature, it is ensured that there are exists some other effective features that has higher relevancy and lower redundancy value than it. This process also leads to decreasing the risk of discounting a feature. The proposed methods have several novelties compared to the previous feature-based OSF methods [17,23–27], as:

1. The proposed methods employ mutual information to analysis relevancy and redundancy of features. Compared to [29,30] which uses rough set theory in their processes, using mutual information has several advantages. While those based on rough sets require $O\left(n^2m\right)$ time steps (n and m show the number of instances and the number of features, respectively), it is only $O(nm)$ for the proposed methods. Moreover, mutual information can be used for both discrete and continues features [31], while rough sets can only be applied on data with discrete variables.
2. The proposed methods do not employ any adjustable user-defined parameters. Thus, compared to [23,26], they can generate more robust results over various information sources.
3. The proposed methods employ an elimination strategy to remove redundant features in further steps, even if they have been previously selected. Compared to [23], this strategy results in returning a set of features with minimal redundancy.
4. The redundancy analysis step of the proposed methods is only performed on relevant features, while in [26,27], in each step the redundancy of so-far-seen feature is computed with all other previously selected features that needs a high computational cost. Also, [26,27] use a k-greedy search strategy to eliminate redundant features by checking all subsets of selected features. Thus, their complexity for evaluating the redundancy and relevancy of each feature is $O(|S_t|2^{|S_t|})$, where S_t denotes the selected feature at time t. Our algorithms take only $O\left(|S_t|^2\right)$ time steps to identify and eliminate redundant features.
5. Compared with the algorithm proposed in [24], the proposed methods calculate the redundancy of features considering the target class. [24] uses the pairwise mutual information to discover redundant features without considering the target class. Two features may be dependent on each other, while each sharing different information about the target class, and thus cannot not considered as redundant features.
6. Compared to [22], the proposed methods are filter-based feature-based OSF methods and do not use any learning model in their processes. In [22] the data samples arrive one-by-one and a learning method is used to evaluate features, and thus it is a wrapper and instance-based OSF method. Therefore, it is much slower than the proposed methods.
7. Although the method proposed in [28] uses the mutual information concept for relevancy and redundancy analysis, it is an offline feature selection method and needs to access the whole feature space and cannot be used online streaming feature selection.

The efficiency of the proposed methods is evaluated on two complex scenarios over 29 datasets in different categories. Our experiments show superiority of the proposed methods over others.

2. Related work and background

2.1. Related work

The aim of feature selection is to select a set of prominent features to improve interpretability and efficiency of the learning

model without degrading model accuracy. Considering the type of data arrival, feature selection approaches can be either offline or online. Offline methods, also known as traditional feature selection methods, need to access the entire feature space to perform their global search [32]. These methods can be categorized into filter, wrapper, embedded and hybrid methods. Filter methods aim to evaluate and rank features based on statistical characteristics of data without using any learning model [4–9]. Then a subset of features with the highest ranks is selected and other are removed from dataset. These types of feature selection methods can broadly be classified into univariate and multivariate approaches. Univariate filter methods assume that the features are independent and use a specific statistical criteria such as information gain [33] or Gini index [34] to evaluate the relevance of each feature individually. Although these methods select the relevant features, they do not consider the relation between the features and cannot identify redundant features. Theoretical and empirical studies showed that redundant features affect accuracy and computational time of the classification [29]. Multivariate filter model has been developed to take into account dependency between the features. Examples of such methods include MIFS-U [35], MRMR [36] and FCBF [28].

Traditional methods need to know about the entire dataset at the start of the algorithm. However, in many real-world applications, it is either impossible to acquire the entire dataset or to wait for the complete set before starting the feature selection process. Offline methods are not often scalable enough for big data applications and data streams. To overcome these issues, several research efforts have been made to iteratively select features in an online manner. Some of these efforts assume that the number of features are fixed and the instances flow one-by-one [22,37,38], while some other methods assume that the number of instances is fixed [39]. The method proposed in [37] uses an incremental technique to weight the features, where features with small weights have little impact on the learning process. These weights can be continuously updated during the incremental learning phase with newly loaded data instances based on an evolving fuzzy classifier. The method proposed in [40] uses an incremental learning algorithm to rank the features when new instances arrive. Recently, [41] proposed a learning strategy based on switching to a neighboring model, which uses local input selectors in the antecedent and regressor vectors in the consequent part. Each feature is associated with a local input selector and once the input selector value becomes zero, the corresponding input variable will be locally eliminated.

In this paper we focus on models in which the number of instances is fixed and the new features arrive one-by-one. We provide a review of similar methods in the following. In [39], the authors proposed an incremental gradient descent algorithm, which maps the feature selection task to a regularized risk minimization problem. They also proposed online incremental version of their algorithm [42]. Zhou et al. [23,25] proposed two streaming feature selection methods, called Information-investing and Alpha-investing, based on streamwise regression. In these methods a statistical criterion is used to analyze the relevancy of newly arrived features. OSFS and fast-OSFS are two other methods that use mutual information in their process [26,27]. Both of them have two phases, one for checking the relevance of features and another one to check their redundancy. In [24] two other methods, called SAOLA and group-SAOLA, were proposed that employ an online pairwise comparison techniques to address high dimensionality and scalability issues. SAOLA uses mutual information to measure pairwise correlation to analysis relevancy and redundancy of features. For a newly arrived feature, if the dependency between it and any of the previously selected features is higher than a predefined threshold, the algorithm removes the one with the lower relevancy with the target class. In group-SAOLA algorithm, when a group of features arrive, the algorithm first removes redundant and irrelevant features from the newly arrived group of features, and then remove the redundant features from other group. In both of these algorithms, deciding to include a feature in the final pool depends on the threshold parameter and setting an inappropriate value for this parameter needs a priori knowledge about the entire dataset. Furthermore, redundancy between two features is calculated based on the values of two features, without any consideration of target class. In [43] a new method, called OGFS, was proposed in which the features are included in a group, and the computations are performed in online intra-group and inter-group selection fashions.

Recently, rough set theory has also been adopted for online feature selection [17,18]. In [17] an online feature selection method, called OS-NRRSAR-SA, was proposed. This algorithm adopts the classical rough set theory to analysis the relevancy of features and removes irrelevant ones. In [18] an incremental feature selection method was proposed for datasets with dynamically-increasing features. This method employs rough set theory in an incremental strategy to reduce computational complexity. Although methods based on rough sets are successful in considering relevancy and redundancy of features for online data streams, they are often computationally costly making their application limited for large-scale datasets.

2.2. Mutual information

Mutual information has been frequently used as a criterion to characterize both the relevancy and redundancy in feature selection methods. In this section, the principles of information theory are discussed with a focus on entropy and mutual information. Both of these concepts are based on Shannon's information theory [44]. Using this concept, the initial uncertainty in the target feature is measured by the entropy, defined as:

$$H(C) = -\sum_{c=1}^{Nc} P(c)\ \log\ P(c) \tag{1}$$

where N_c is the number of classes and $P(c)$ is define as follows:

$$P(c) = \frac{|number\ of\ samples\ in\ class\quad c|}{|Total\ number\ of\ classes|}$$

On the other hand, the average uncertainty after knowing the feature vector f (with N_f components) is defined as the conditional entropy $H(C|f)$ as follows:

$$H(C|f) = -\sum_{f=1}^{N_f} P(f)\left(\sum_{c=1}^{Nc} P(c|f)\ \log\ P(c|f)\right) \tag{2}$$

Generally, the conditional entropy is equal to or less than the initial entropy. It is equal if and only if the features and the target class are independent. The amount by which the uncertainty is decreased, is $I(C;f)$ that is defined as follows:

$$H(C|f) = H(C) - I(C;f) \tag{3}$$

In other words, the mutual information measures the mutual dependence between two variables as:

$$I(C;f) = \sum_{i=1}^{N}\sum_{j=1}^{M} p(c_i,f_j)\log_2\left(\frac{p(c_i,f_j)}{p(c_i)p(f_j)}\right) \tag{4}$$

where $p\left(c_i,f_j\right)$ refers to joint probability of c_i and f_i. The aim of employing mutual information in feature selection is to compute the correlation between the features and classes. A feature with the highest mutual information with the class is considered as the most informative one. Other measures such as conditional mutual information $I\left(C;f/S\right)$ and interaction gain $I(C;f;S)$ are used to compute

the correlation between a feature f, set of feature S and target class C as follow:

$$I(C;f/S) = H(C/S) - H(C/f,S) \tag{5}$$

$$I(C;f;S) = I(C,f;S) - I(C;S) - I(C,f) \tag{6}$$

The interaction information can be set to negative, zero and positive values. It is positive when both features can provide information that cannot be provided individually. On the other hand, it is negative when each feature provides the same information and zero while two features are independent in the context of the class label.

3. Proposed methods

In this section we provide the details of the proposed feature selection methods for online stream feature selection problem, called OSFSMI and OSFSMI-k. These methods are based on mutual information to take into account relevancy and redundancy concepts in their processes. In both of these methods it is assumed that the entire dataset is not accessible at the beginning of the process and features appear incrementally in an online manner.

3.1. Online stream feature selection based on mutual information (OSFSMI)

The aim of this section is to describe the details of the first proposed method. Let's define a data stream by $D = (F_t, C)$ where $F_t = \{f_1, f_2, \ldots, f_t\}$ is a sequence of features presented till time t and C is the class attribute. Also, suppose S_t and f_t show a set of selected features and the newly arrived feature in time t, respectively. In this method, the goal is to select an optimal subset $S_t \subseteq \{S_{t-1} \cup f_t\}$ with minimum number of features that maximizes relevancy to the target feature and minimizes redundancy between the selected features. Thus, the objective function is defined as follows:

$$S_t = \underset{S_t'}{arg\ \min}\{|S_t'| : S_t' = \underset{\chi \subseteq \{S_{t-1} \cup f_t\}}{arg\ \min} H(C|\chi)\} \tag{7}$$

In other words, the aim is to find a subset $S_t \subseteq \{S_{t-1} \cup f_t\}$ with minimum number of features that minimizes the following equation:

$$H(C|S_t) = H(C) - I(C;S_t) \tag{8}$$

Finding an exact solution for the above objective function needs a high computational, since one has to consider all possible subsets that requires computing mutual information between a numbers of $O(2^{|S_t|})$ pairs. Motivated by this, in this paper our aim is to find an approximate solution for the above objective function. To this end, adding or removing a newly arrived feature f_t to S_{t-1} is decided by the relevance of f_t to the target feature C, which ensures that the uncertainty of the target class decreases. Besides, to minimize the size of the selected features, some of highly redundant and less relevance features are decided to be removed from the previously selected features S_t.

The first proposed method, called Online Streaming Feature Selection based on Mutual Information (OSFSMI) consists of two main steps. The goal of the first step is to identify those of relevant features and to consider them in the search process. In this step, the relevancy of a newly arrived feature is first assessed, and then it is added to the selected feature set if it is relevant to target class. In the second step, the aim is to identify and eliminate those of ineffective features and also to minimize the size of selected feature set through several iterations. Including effective feature f in target class C results in decreasing its uncertainty. The effectiveness of feature f is computed by a linear combination between its relevancy and redundancy values. A higher effectiveness value of a feature means that the feature has higher relevancy to the target class and lower redundancy with other selected features. The features that have lower effectiveness value than the effectiveness value of the newly arrived feature are removed from the feature set. This process is continued iteratively until there is no more ineffective feature than the newly arrived feature in the feature set. The details of the OSFSMI is presented in Fig. 1.

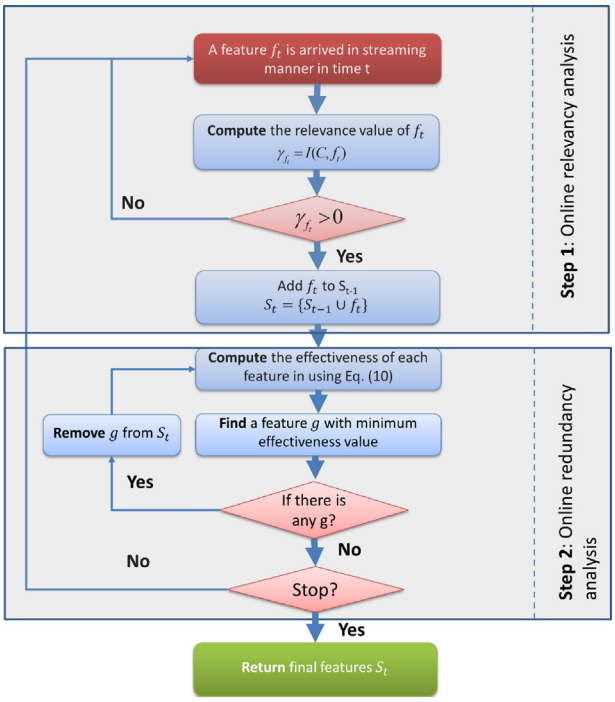

Fig. 1. The general framework of OSFSMI method.

The algorithm starts with an empty subset and the following steps are performed by arriving a feature f_t at time t.

3.1.1. Step 1: online relevancy analysis

The aim of this step is to compute the relevancy value of a feature f_t with the target feature as follows:

$$\gamma_{f_t} = I(C, f_t) \tag{9}$$

where γ_{f_t} is computed using Eq. (4). If $\gamma_{f_t} > 0$, the newly arrived feature f_t carries predictive information to the target feature C and the algorithm add f_t to the selected features S_t, otherwise it is discarded. Therefore, it can be concluded that only those of strictly non-relevant features ($\gamma_{f_t} = 0$) are discarded and the others are considered in the further evaluation processes. When a feature is discarded, it cannot be recalled later, and thus setting the value of 0 to assess the relevancy of a feature leads to decrease the risk of discontinuity and instability of the algorithm. On the other hand, the aim is to give a chance to the newly arrived feature if it has some relevancy with the target class.

3.1.2. Step 2: online redundancy analysis

The goal of this step is to minimize the number of selected features and also to identify and eliminate the redundant ones. To this end, relevancy and redundancy values of each feature $g \in S_t = \{S_{t-1} \cup f_t\}$ are computed and the features that have lower effectiveness value than f_t are removed. The following equation is used to measure the effectiveness of each feature $g \in S_t S_t = \{S_{t-1} \cup f_t\}$ with respect to

$$\lambda_{g,S}t = I(C,g) - \beta\ I(C;g;\{S_t \backslash g\}) \qquad \forall g \in S_t \tag{10}$$

f_t: the newly arrival feature f at time t,
S_t: the selected feature subset till time t, S_0: {}.
1. **Repeat**
2. f_t←newly arrived feature at time t
// Step1: Checking for relevance of new arrival feature f_t
3. **Compute** γ_{f_t} using Eq.(9)
4. **If** $\gamma_{f_t} > 0$, then $S_t \leftarrow S_{t-1} \cup f_t$, **Else If** $\gamma_{f_t} = 0$, **Go to** line 14;
//Step 2: Checking for redundancy features in S_{t-1}
5. **For** each feature $g \in S_t$ //, also including the new arrival feature f_t
6. **Compute** λ_{g,S_t} using Eq.(12)
7. **End For**
8. **Repeat**
9. **Find** feature $g \in S_t \backslash \{f_t\}$ with minimum λ_{g,S_t} which $\lambda_{g,S_t} < \lambda_{ft,S_t}$
10. **If** there is no feature g
11. Go to line 16.
12. **Else** Remove feature g from S_t
13. **End If**
14. **Update** λ_{g,S_t} for each feature $g \in S_t$
15. **Until** there is no feature $g \in S_t \backslash \{f_t\}$ with less redundancy than redundancy of f_t or *size*(S_t)<=2
16. **Until** no new feature are available
17. **Return** S_t;

where $\beta \in (01]$ is a parameter that controls redundancy penalty. It is worth mentioning that the first and second terms in the right hand side of Eq. (10) denote, respectively, the relevance and redundancy of feature g with regards to a set of previously selected features $S_t \backslash g$. These terms are computed using Eqs. (4) and (6). In each iteration, if g has effectiveness lower than that of f_t(i.e., $\lambda_{g,S_t} < \lambda_{f_t,S_t}$), it is removed from the feature set. If there is not any ineffective feature, the iteration is stopped and the method waits for a new feature. If there are two or more features that have lower effectiveness values than that of newly arrived feature, the one with the minimum effectiveness value is removed. If two or more features have equal effectiveness values, one of them with lower relevancy value is removed from the feature set. Also, by removing a feature the effectiveness value of the remaining ones must be computed again for the next iterations. The iteration process is repeated until no more ineffective feature is identified or the feature set S_t contains only one feature (i.e.$|S_t| = 1$). Using this strategy, a relevant feature is not removed at the first step and it is given by one more chance to be proceed in the further steps. Also, by removing a feature, it is ensured that there exist some other effective features that have higher relevancy and lower redundancy values that it. This process also leads to decrease the risk of discounting a feature.

Since computing $I(C;g;\{\})$ requires a large amount of storage, we use an alternative method to compute any interaction gain $I(C;g;S)$, where g is a feature and S is a set of features. In [35] Kwak and Choi proposed a method to estimate $I(C;g;S)$ as follows:

$$I(C;g;S) = \sum_{f_s \in S} \frac{I(C,f_s)}{H(f_s)} I(g;f_s) \tag{11}$$

Thus, considering their idea, the Eq. (10) is rewritten as follows:

$$\lambda_{g,S_t} = I(C;g) - \beta \sum_{f_s \in S_t \backslash g} \frac{I(C;f_s)}{H(f_s)} I(g,f_s) \tag{12}$$

It should be noted that considering the relevancy of the features in computing the effectiveness of each feature in Eq. (12) results in assigning a low effectiveness value to those of features which have no additional useful information. The details of the proposed OSFSMI method is presented in Algorithm 1.

Algorithm 1. OSFSMI: Online Stream Feature Selection based on Mutual Information

3.2. Online stream feature selection based on mutual information with fixed number of features (OSFSMI-k)

OSFSMI seeks to select a set of features with minimal size and high accuracy. However, in some real-world applications one needs to select fixed number of features. To this end, we modify OSFSMI in such a way that returns top-k features. The pseudo-code of the second proposed method, called OSFSMI-k, is presented in Algorithm 2. A simple way to do this is to let the OSFSMI select those of correlated features (i.e., $I(f_t;C) > 0$) until the size of the selected features S_t reaches constant k (i.e., $|S_t| = k$). Otherwise, when a relevant feature arrives, the algorithm eliminates the feature with the least effectiveness value. If two features have the same effectiveness value, the algorithm eliminates the one with lower relevancy value.

Algorithm 2. OSFSMI-k. Online Feature Selection with fix number of features

f_t: the newly arrived feature f at time t,
S_{t-1}: the selected feature subset till time t, S_0: {}.
k: Predefined number of selected features.
1. Repeat
2. f_t←newly arrived feature at time t
3. **Compute** γ_{f_t} using Eq.(9).
4. **If** $\gamma_{f_t} > 0$, $S_t = S_{t-1} \cup f_t$, **Else If** $\gamma_{f_t} = 0$, discard it and **go to** 2;
5. **If** $size(S_t) \succ k$
6. **Compute** λ_{f_t,S_t} using Eq. (12)
7. $min = \gamma_{f_t,S_t}$
8. $D = f_t$
9. **For** each feature $g \in S_t$
10. **Compute** λ_{g,S_t} using Eq.(12)
11. **If** $(min > \lambda_{g,S_t})$ then
12. $min = \lambda_{g,S_t}$
13. $D \leftarrow g$
14. End if
15. **If** $(min == \lambda_{g,S_t})$ then
16. **If**$(\gamma_g \prec \gamma_D)$
17. $min = \lambda_{g,S_t}$
18. $D = g$
19. End if
20. End if
21. End for
22. **Remove** feature in D from S_t.
23. **Update** λ_{g,S_t} for each feature $g \in S_t$
24. **Until** (no new feature is available)
25. Return S_t;

3.3. Analyzing computation complexity

The time complexity of OSFSMI method depends on computing effectiveness of the newly arrived feature f_t and the previously selected features $g \in S_{t-1}$. Based on Eq. (12), computing effectiveness of each newly arrived feature f_t takes$O(|S_{t-1}|)$ times. There are $|S_{t-1}|$ features at time t, and finding the feature with minimum effectiveness lower than f_t and also updating the effectiveness of all features in S_t have complexity of $O(S_t)$ each. It can be concluded that the overall time complexity of OSFSMI method is$O\left(|S_{t-1}|^2\right)$, which depends only on the number of selected features and is independent from the total number of features. OSFSMI-k requires $O(|S_{t-1}|)$ time steps to compute the effectiveness values and identifying the less effective features. In many real-world applications, only a small portion of the feature space is informative and should be selected, which means $|S_t|$ is often much smaller than the overall feature space.

4. Experimental results

In this section the effectiveness of the proposed algorithms are assessed and compared with two different types of feature selection methods. First, we compare the first proposed method (OSFSMI) with five state-of-the-art online feature selection methods including; SAOLA [24], group-SAOLA [43], OSFS [27], fast-OSFS [26] and Alpha-investing [23] algorithms. In order to provide streaming feature selection scenario, it is supposed that the features are not available and they are presented one-by-one. Then, the second proposed method (OSFSMI-k) is compared with 11 traditional univariate and multivariate feature selection methods including Information Gain (IG) [33], Symmetrical Uncertainty (SU) [45], Fisher Score (FS) [46], Gini index (GI) [34], Term of Variance [47], MRMR [36], MIFS [48], MIFS_U [35], NJMIM [49], DISR [50] and NMIFS [51] algorithms. In these types of the algorithms, the entire datasets are available at the beginning of the search process. Each method includes several adjustable parameters that need to be tuned before starting the feature selection process. The statistical significance level of α for SAOLA, group-SAOLA, OSFS and fast-OSFS is set to 0.1. Also, the parameter β for the proposed methods, which controls the redundancy penalty in Eq. (10), is set to 0.1.

In order to evaluate the performance of the proposed methods, 29 different datasets[1] are used in the experiments. The datasets are chosen from different categories such as face images, biological, maicroarray and artificial datasets which are listed with sufficient details in Table 1. To compare the efficiency of the algorithms the average results of DT [45], KNN [52], Naive Bayes [53] classifiers over ten independent runs are reported in the experiments. The parameter k for KNN classifier is set to 3 and each dataset is randomly split into a training set (66%) and a test set (34%). The experimental results are reported in terms of the classification accuracy, subset selected size or compactness, running time and stability measures. Also, a nonparametric Friedman test [54] is performed in the experiments to statistically compare the significance of the results obtained by different methods.

We used the source code of LOFS data mining software toolbox for implementation of online streaming feature selection such as SAOLA, group-SAOLA, OSFS, fast-OSFS and Alpha-investing [55]. Besides, we used the Arizona State University feature selection toolbox[2] for implementation of Gini index, Fisher Score, Information Gain and MRMR. Note that the other methods are implemented by the authors using MATLAB programming language. Also, for features continuous values, the Fisher's Z-test [56] is adopted to calculate correlations between features. All methods are implemented in Matlab software and carried out on a computer with Windows8, 8GB memory, and 2.4 GHz CPU.

4.1. Comparison of OSFSMI with online stream feature selection methods

In this section, OSFSMI is compared to SAOLA, group-SAOLA, OSFS, fast-OSFS and Alpha-investing algorithms. Note that the full feature space is not available at the start of the search process. As the streaming order of the features may affect the final results, several random streaming orders are generated for each dataset and the average results are reported.

4.1.1. Classification accuracy

In the experiments, the classification accuracy of OSFSMI is compared with others, and the results are shown in Tables 2–4 when using NB, KNN and DT classifiers, respectively. These results show that the average classification accuracy over all datasets with ten different orders of features. In all Tables the columns show online feature selection methods and each row denotes a dataset. The best value in each row is shown by bold and underlined. The last two rows show the average values and Friedman test over all datasets, respectively.

Table 2 shows the average classification accuracy of NB classifier. The cases indicated by start (*) represent those cases where the within-class variance of the features in the training set is too small to classify the test set. From the results it can be seen that in most cases OSFSMI obtained higher classification accuracy compared to the other online stream feature selection methods. Friedman test shows that the proposed method generally achieved the best classification accuracy. For example, the average value of SAOLA, group-SAOLA, OSFS, fast-OSFS and Alpha-investing were 56.97, 57.11, 53.23, 57.40 and 52.38, compared to 58.64 for the proposed method. This means that the proposed method is better than SAOLA, group-SAOLA, OSFS, fast-OSFS and Alpha-investing about 2.84%, 2.6%, 9.22%, 2.11% and 10.67%, respectively. Similar results are also reported for KNN and DT classifiers in Tables 3 and 4, respectively.

4.1.2. Comparison of the selected subset size

Table 5 reports the average number of selected features identified by five algorithms over ten different orders of features. The results indicate that OSFS, fast-OSFS and Alpha-investing algorithms always choose the smallest number of features compared to others. For instance, in the case of Prostate_GE dataset, SAOLA, group-SAOLA, OSFS, fast-OSFS, Alpha-investing and OSFSMI methods select the average number of 8.6, 7.66, 1.4, 2.6, 3.6 and 9.2 features, respectively. The results also show that in some cases the number of selected features for OSFS, fast-OSFS and Alpha-investing are the highest compared to others. For example, for CINA0 dataset, SAOLA, group-SAOLA, OSFS, fast-OSFS, Alpha-investing and OSFSMI methods select 9, 6.4, 19, 21.33, 71.33 and 7 features, respectively. The average number of selected features for the proposed method was 7.84, which indicates that the proposed method performs much better than SAOLA, group-SAOLA and Alpha-investing methods, however OSFS and fast-OSFS are better than the proposed method in this senses.

4.1.3. Comparison of running times

The methods were also compared based on their execution time and the results are shown in Table 6. It can be seen from the results that in most cases the lowest running time is for fast-OSFS and Alpha–investing. However, the average running times over all datasets indicates that the proposed method is faster than OSFS,

[1] http://featureselection.asu.edu/datasets.php and http://csse.szu.edu.cn/staff/zhuzx/Datasets.html.

[2] http://featureselection.asu.edu/old/software.php.

Table 1
Details of datasets used in the experiments.

	Data Set	#Instances	#Features	#Classes	keywords	type
1	ALLAML	72	7129	2	continuous, binary	Biological Data
2	GLIOMA	50	4434	4	continuous,multi-class	Biological Data
3	Prostate_GE	102	5966	2	continuous, binary	Biological Data
4	SMK_CAN_187	187	19993	2	continuous, binary	Biological Data
5	ORL	240	1024	40	continuous, binary	Image dataset
6	GLA-BRA-180	180	49151	4	continuous, multi-class	Biological Data
7	OHSUMED-F	13929	1024	2	continuous, binary	Artificial
8	Arrhythmia	452	279	13	continuous,multi-class	Biological Data
9	MLL	72	12582	3	continuous, binary	Microarray Data
10	Carcinom	174	9182	11	continuous,multi-class	Biological Data
11	Arcene	200	10000	2	continuous, binary	Mass Spectrometry
12	Madelon	2600	500	2	continuous, binary	Artificial
13	orlraws10P	100	10304	10	continuous,multi-class	Face Image Data
14	pixraw10P	100	10000	10	continuous,multi-class	Face Image Data
15	warpAR10P	130	2400	10	continuous,multi-class	Face Image Data
16	warpPIE10P	210	2420	10	continuous,multi-class	Face Image Data
17	Yale	165	1024	15	continuous,multi-class	Face Image Data
18	Extended Yale	203	3312	5	continuous,multi-class	Biological Data
19	CINA0	16033	132	2	continuous, binary	Artificial
20	Breast	97	24400	2	continuous, binary	Microarray Data
21	SRBCT	83	2308	4	continuous, multi-class	Microarray Data
22	CNS	60	7130	2	continuous, binary	Microarray Data
23	Lung cancer	203	12601	5	continuous, multi-class	Biological Data
24	Dexter	420	20001	2	continuous, binary	Biological Data
25	Lymphoma	66	4026	3	continuous, multi-class	Microarray Data
26	Ovarian	253	15154	2	discrete, binary	Microarray Data
27	ColonTumor	62	2000	2	discrete, binary	Microarray Data
28	Nci9	60	9712	9	discrete,multi-class	Biological Data
29	Lung	203	3312	5	discrete,multi-class	Biological Data

Table 2
Naïve Bayes (NB) classification accuracy results at the end of online feature selecting methods applied on different datasets.

	Dataset	SAOLA	group-SAOLA	OSFS	Fast-OSFS	Alpha-investing	OSFSMI
1	ALLAML	**95 ± 4.07**	91.42 ± 3.57	86.42 ± 5.29	86.42 ± 5.86	60.71 ± 5.05	90 ± 7.31
2	GLIOMA	64.81 ± 6.41	**66.66 ± 5.31**	59.25 ± 6.41	55.55 ± 5.51	50 ± 11.1	62.96 ± 8.48
3	Prostate_GE	88.5 ± 4.18	90 ± .20	88.5 ± 4.54	92 ± 3.25	81.5 ± 10.09	**92.5 ± 4.33**
4	SMK_CAN_187	**68.1 ± 7.49**	68.01 ± 7.44	63.51 ± 6.55	64.86 ± 6.11	50.27 ± 6.64	65.67 ± 6.16
5	ORL	27.37 ± 6.63	26.87 ± 8.75	22 ± 10.45	36.75 ± 2.91	39.16 ± 4.06	**45.75 ± 4.2**
6	GLA-BRA-180	53.52 ± 3.98	53.80 ± 7.34	51.26 ± 3.80	65.53 ± 7.08	50.7 ± 1.99	**61.97 ± 4.10**
7	OHSUMED-F	84.75 ± .05	86.14 ± 2.59	84.18 ± .06	84.11 ± .03	83.27 ± .04	**92.44 ± .08**
8	Arrhythmia	*	*	*	*	*	*
9	MLL	85.71 ± 5.64	84.52 ± 2.06	75.71 ± 8.52	83.57 ± 10.8	59.28 ± 10.59	**86.1 ± 3.91**
19	Carcinom	*	*	*	*	*	*
11	Arcene	60 ± 7.19	60.33 ± 4.79	62.78 ± 6.91	67.34 ± 6.16	62.53 ± 4.52	**69.62 ± 4.19**
12	Madelon	59.65 ± 1.83	58.68 ± 1.94	59.73 ± .99	59.5 ± 1.07	61.09 ± 1.05	**62.11 ± 1.24**
13	orlraws10P	48 ± 9.25	49.16 ± 11.54	37.5 ± 5	39 ± 8.9	**64.5 ± 7.37**	61.5 ± 4.18
14	pixraw10P	*	*	*	*	*	*
15	warpAR10P	30.8 ± 3.64	28.4 ± 5.68	23.2 ± 3.03	**32.8 ± 7.56**	**32.8 ± 16.4**	28.4 ± 1.67
16	warpPIE10P	48.5 ± 6.03	50.25 ± 10.54	46.75 ± 7.88	61.5 ± 4.54	**64.25 ± 6.93**	41.75 ± 6.03
17	Yale	16 ± 302	16 ± 2.4	14 ± 5.2	16.33 ± 4.91	*	**24.66 ± 5.19**
18	Extended Yale	8.25 ± 3.81	8.34 ± 2.25	9.31 ± 2.8	13.54 ± 3.81	24.1 ± 4.59	**16.85 ± 5.38**
19	CINA0	85.31 ± .99	87.76 ± .56	90.9 ± .2.0	90.84 ± .34	**92.57 ± .05**	89.40 ± .11
20	Breast	62.28 ± 10.95	64.91 ± 11.86	64.21 ± 8.64	**66.84 ± 5.12**	50.12 ± 3.5	63.15 ± 3.22
21	SRBCT	90.62 ± 5.41	**91.25 ± 5.13**	72.5 ± 12.77	83.12 ± 4.73	68.75 ± 13.07	85.62 ± 12.22
22	CNS	**60.86 ± 9.22**	60 ± 9.42	58.26 ± 6.52	56.52 ± 6.14	54.78 ± 9.01	58.26 ± 6.59
23	lung	**91.13 ± 4.26**	90.12 ± **3.39**	74.93 ± **8.92**	85.06 ± **4.32**	79.74 ± **3.90**	85.06 ± **3.83**
24	Dexter	*	*	*	*	*	*
25	Lymphoma	**96 ± 4**	**96 ± 4**	88 ± 4	94.66 ± 6.11	72 ± 8	85.33 ± 6.11
26	Ovarian	**99.33 ± .57**	**99.33 ± .57**	98 ± 3.46	**99.33 ± .57**	97.66 ± 1.52	97 ± 1.73
Average classification Accuracy		56.97	57.11	53.23	57.40	52.38	**58.64**
Fridman rank		3.24	3.08	4.32	3.26	4.24	**2.86**

fast-OSFS and Alpha-investing methods while being slower than the SAOLA and group-SAOLA methods.

4.1.4. Comparison of the stability of methods

Stability of an algorithm refers to how the algorithm identifies similar feature set over various independent runs and various orders of features. The aim of stability analysis is to quantify between two selected feature set. Generally, a stable algorithm generates similar results in each iteration. In order to evaluate the stability of the online feature selection algorithms, we use the stability measure which proposed in [57]. To obtain the similarity between two feature sets S_i and S_j, this measure creates a complete weighted bipartite graph $G = \langle V, E, W \rangle$ between them where $V = S_i \cup S_j$ and $E = \{(v_1, v_2) | v_1 \in S_i, v_2 \in S_j\}$, and the weights of edges are determined by normalized mutual information between the corresponding features. The generated graph between two node sets is matched using Hungarian algorithm and a maximum

Table 3
KNN classification accuracy results at the end of online feature selection methods applied on different datasets and.

	Dataset	SAOLA	group-SAOLA	OSFS	Fast-OSFS	Alpha-investing	OSFSMI
1	ALLAML	**91.42 ± 3.19**	91.23 ± 2.06	85.71 ± 7.14	85 ± 5.86	56.42 ± 6.38	86.42 ± 5.68
2	GLIOMA	66.6 ± 8.4	61.44 ± 14.16	55.55 ± 9.62	**72.22 ± 7.85**	44.44 ± 7.85	69.66 ± 23.57
3	Prostate_GE	**95 ± 1.76**	90.83 ± 2.88	85.5 ± 3.25	90.5 ± 3.25	93 ± 4.10	93.5 ± 3.35
4	SMK_CAN_187	65.67 ± 5.19	**66.21 ± 3.57**	59.45 ± 2.86	63.51 ± 2.70	51.89 ± 5.11	60 ± 5.02
5	ORL	27.87 ± 7.36	28.12 ± 9.92	17.87 ± 12.26	41.12 ± 4.9	39.87 ± 4.4	**53.37 ± 3.32**
6	GLA-BRA-180	55.77 ± 5.59	52.39 ± 5.84	49.29 ± 8.56	53.8 ± 5.4	48.73 ± 5.5	**64.22 ± 8.36**
7	OHSUMED-F	98.62 ± .08	98.62 ± .08	98.72 ± .04	98.71 ± .02	**98.73 ± .01**	98.69 ± .02
8	Arrhythmia	45.97 ± 2.5	45.76 ± 4.32	43.78 ± 1.73	44.56 ± 1.29	45.12 ± 2.08	**47.45 ± 2.25**
9	MLL	**88.57 ± 6.86**	85.71 ± 6.18	82.14 ± 9.78	82.85 ± 9.91	52.85 ± 6.86	82.85 ± 5.86
19	Carcinom	78.71 ± 4.08	**79.06 ± 7.29**	46.28 ± 4.08	56.64 ± 5.77	43.55 ± 10.70	59.96 ± 7.96
11	Arcene	62.59 ± 2.91	62.24 ± 6.37	63.54 ± 4.04	62.78 ± 7.73	**65.31 ± 5.34**	65.06 ± 4.06
12	Madelon	55.76 ± 1.79	54.90±.16	**56.4 ± 1.97**	55.96 ± 2.15	58.03 ± 3.91	**56.4 ± 1.58**
13	orlraws10P	52 ± 6.93	53.33 ± 8.77	43 ± 5.41	36.5 ± 5.75	**65 ± 4.67**	61 ± 9.28
14	pixraw10P	76.25 ± 12.37	75.5 ± 5.75	58.75 ± 12.37	78.25 ± 15.9	80.5 ± 3.53	**82.5 ± 7.07**
15	warpAR10P	42.4 ± 5.17	42 ± 5.91	28.4 ± 4.77	36 ± 6.63	38.4 ± 6.84	**48 ± 6.78**
16	warpPIE10P	65.25 ± 8.72	76.80 ± 5.4	57.75 ± 8.58	75.25 ± 4.54	76.75 ± 8.03	**77.8 ± 9.03**
17	Yale	17.33 ± 2.52	13.86 ± 4.31	13 ± 7.58	19.33 ± 4.01	26 ± 2.78	**28.33 ± 6.51**
18	Extended Yale	7.38 ± 1.2	4.50 ± 1.75	12.12 ± 4.82	13.48 ± 2.88	**52.33 ± .86**	30.79 ± 1.91
19	CINA0	89.47 ± .50	89.57 ± .78	90.8 ± .40	**90.99 ± .25**	90.73 ± .42	89.19 ± .58
20	Breast	61.40 ± 8.45	62.15 ± 8.52	63.15 ± 8.52	**63.68 ± 4.7**	57.36 ± 7.3	60 ± 9.59
21	SRBCT	**90.62 ± 4.94**	88.12 ± 4.07	70.62 ± 9	81.87 ± 5.13	51.87 ± 7.52	85.62 ± 6.48
22	CNS	**61.73 ± 3.63**	60 ± 3.63	57.39 ± 4.76	58.26 ± 6.59	60 ± 8.36	60 ± 8.36
23	lung	**84.30 ± 1.91**	84.05 ± **2.91**	75.18 ± **4.15**	81.77 ± **4.95**	80.50±**.69**	78.22 ± **2.26**
24	Dexter	**85.91 ± 2.81**	86.11 ± 2.93	85.31 ± 5.96	84.12 ± 1.81	54.56 ± 3.58	72.61 ± 0
25	Lymphoma	96	**97.33 ± 2.3**	88 ± 6.92	94.66 ± 6.11	69.33 ± 2.30	84 ± 6.92
26	Ovarian	99 ± 1.41	**99** ± 1.00	98 ± 1.34	99.8 ± .44	**99.4 ± .54**	67.8 ± 1.3
Average classification Accuracy		67.75	67.30	60.98	66.21	61.56	**68.97**
Fridman Rank		3.00	3.28	4.63	3.40	3.76	**2.90**

Table 4
Decision Tree (DT) classification accuracy results at the end of online feature selection methods applied on different datasets.

	Dataset	SAOLA	group-SAOLA	OSFS	Fast-OSFS	Alpha-investing	OSFSMI
1	ALLAML	87.85 ± 6.96	91.66 ± 2.06	86.42 ± 8.89	88.57 ± 6.86	63.57 ± 4.65	**91.42 ± 5.41**
2	GLIOMA	55.55 ± 7.85	**61.11 ± 10.39**	47.22 ± 11.78	47.22 ± 3.92	44.4 ± 7.85	47.22 ± 19.6
3	Prostate_GE	87.5 ± 5.86	85.83 ± 3.81	82.5 ± 3.06	83 ± 4.10	84.5 ± 7.37	**92.5 ± 3.06**
4	SMK_CAN_187	57.29 ± 5.93	**65.31 ± 8.79**	59.45 ± 4.58	59.72 ± 3.36	56.48 ± 4.09	58.37 ± 4.09
5	ORL	24.87 ± 4.53	20.83 ± 7.4	15.37 ± 6.44	28.37 ± 4.83	26.87 ± 5.07	**31.5 ± 1.68**
6	GLA-BRA-180	50.98 ± 9.92	46.19 ± 6.92	48.45 ± 2.35	51.83 ± 6.56	45.91 ± 5.85	**55.49 ± 2.13**
7	OHSUMED-F	98.66 ± .7	98.65 ± .02	98.43 ± .20	98.4 ± .19	98.01 ± .05	**98.71±.01**
8	Arrhythmia	45.4 ± 4	48.50 ± 3.35	**49.01 ± 3.49**	47.24 ± 3.66	44.27 ± 4.23	48.87 ± 2.95
9	MLL	77.85 ± 4.65	71.42 ± 3.51	77.85 ± 5.86	**79.28 ± 2.98**	54.28 ± 11.95	72.14 ± 8.14
19	Carcinom	**56.06 ± 7.49**	53.12 ± 9.63	44.33 ± 7.6	44.92 ± 5.71	38.47 ± 9.52	49.41 ± 3.53
11	Arcene	61.51 ± 7.67	63.81 ± 5.06	63.54 ± 4.93	62.53 ± 5.63	63.79 ± 4.86	**64.3 ± 5.01**
12	Madelon	55.34 ± .84	54.80 ± 1.60	54.28 ± 1.96	54.92 ± 2	54.88 ± 3.62	**56.55 ± 1.97**
13	orlraws10P	45 ± 7.21	46.66 ± 17.73	40.5 ± 4.8	32 ± 5.86	**60.5 ± 4.1**	51 ± 6.98
14	pixraw10P	76.25 ± 1.76	80.3 ± 8.50	65 ± 17.67	63.75 ± 15.90	83.75 ± 1.76	**84.5 ± 7.07**
15	warpAR10P	36.8 ± 4.60	35 ± 6.09	28.8 ± 6.26	34.8 ± 4.38	34.8 ± 4.38	**37.6 ± 9.31**
16	warpPIE10P	52 ± 2.59	54.25 ± 5.56	48 ± 6.76	58.25 ± 6.53	57.5 ± 1.53	**59.75 ± 8.54**
17	Yale	16 ± 2.52	15.33 ± 4.31	12.66 ± 4.8	18.66 ± 5.19	**29.66 ± 5.7**	21 ± 3.65
18	Extended Yale	7.95 ± 1.78	15.60 ± 1.84	15.11 ± 3.55	18.2 ± 2.95	**47.85 ± 1.61**	30.79 ± 1.51
19	CINA0	92.55 ± .16	**92.86 ± .29**	91.9 ± .1	91.57 ± .28	90.85 ± .31	90.24 ± .09
20	Breast	**61.40 ± 11.86**	55.78 ± 12.81	63.68 ± 6	57.36 ± 7.3	58.94 ± 10.94	54.21 ± 6.33
21	SRBCT	80 ± 4.191	83.12 ± 5.22	71.25 ± 8.67	75.62 ± 8.08	67.5 ± 13.54	**85 ± 10.68**
22	CNS	53.04 ± 9.91	54.78 ± 7.7	53.91 ± 6.59	**60.86 ± 6.14**	59.13 ± 9.01	59.13 ± 6.59
23	lung	73.67 ± **3.39**	74.93 ± **2.88**	72.40 ± **6.95**	**77.97 ± 6.11**	76.96 ± **3.02**	72.40 ± **3.27**
24	Dexter	86.11 ± 1.49	84.32 ± 2.68	83.92 ± 3.57	**85.91** ± 1.23	61.70 ± 9.22	75.19 ± .68
25	Lymphoma	84 ± 8	85.33 ± 8.32	84 ± 0	**88 ± 6.92**	66.66 ± 6.11	84 ± 0
26	Ovarian	95.4 ± 2.3	95.4 ± 2.3	96 ± 1.41	**97 ± .7**	96.4 ± 1.14	95.2 ± 1.48
Average classification Accuracy		62.27	62.88	59.76	61.76	60.29	**64.09**
Freidman Rank		3.46	3.17	4.38	3.17	4.09	**2.71**

weighted bipartite graph is selected. Finally, the overall similarity between two feature sets is the final matching cost.

To calculate the stability values, each dataset is first split into four folds with equal number of samples, and then a feature selection method is applied on three out of four folds over several independent iterations. In each iteration, features are presented one-by-one with various orders. The process is repeated for five independent iterations to generate different feature sets and the average stability values are reported in Fig. 2. It can be seen that OSFSMI achieved the most stable results in most cases.

4.2. Comparing OSFSMI-k with offline feature selection methods

Several experiments were performed to compare the second proposed method (OSFSMI-k) with well-known and state-of-the-art traditional feature selection methods including IG, SU, FS, GI, TV, MRMR, MIFS, MIFS-U, NJMIM, DISR and NMIFS. These methods, also known as batch feature selection methods, need to access the entire feature space at the beginning of their processes. The comparison results are reported in terms of classification accuracy using NB and DT classifiers and the running time. The average classification

Table 5
The selected subset size for SAOLA, group-SAOLA, OSFS, fast-OSFS, Alpha-investing and OSFSMI methods.

	Dataset	SAOLA	group-SAOLA	OSFS	Fast-OSFS	Alpha-investing	OSFSMI
1	ALLAML	22.2	18	2	4.2	1.2	6.6
2	GLIOMA	16	12.8	2	4	1	5.5
3	Prostate-GE	8.6	7.66	1.4	2.6	3.6	9.2
4	SMK_CAN_187	9.6	8.33	1.6	4.4	1.2	6.6
5	ORL	3	2.66	2.2	5	6	11
6	GLA-BRA-180	25.8	16.4	2.6	6.2	2.2	8.4
7	OHSUMED-F	10.66	9.4	24.33	25.66	24.7	6.66
8	Arrhythmia	4.62	3.6	2.25	3.5	4.12	8
9	MLL	27	22	2	4.6	4.2	8.6
19	Carcinom	35.12	29.8	3.87	7	3.7	11.37
11	Arcene	19.4	15.33	2.2	5	6	8
12	Madelon	7.2	6.33	4.6	5.2	3.2	3.8
13	orlraws10P	3.4	5.33	1	2.8	3	10
14	pixraw10P	8	7	1.5	4	2	15
15	warpAR10P	4.2	4	2	3.6	5.2	7.2
16	warpPIE10P	3.4	4	3	4.2	12	6
17	Yale	1.6	1.6	1.2	1.6	7.2	5.6
18	Extended Yale	2	1	2.4	3	6.74	5.8
19	CINA0	9	6.4	19	21.33	71.33	7
20	Breast	21.33	13.8	1.2	3.8	4.2	8.4
21	SRBCT	19.4	18.8	2.4	4.6	3.6	7.8
22	CNS	8.2	6.8	1	1.8	2.8	5.4
23	lung	34	27.4	3.4	7	12	6.2
24	Dexter	20.33	19.33	8	10	8.33	4.33
25	Lymphoma	21.66	23.66	2	4	1	13.66
26	Ovarian	8.8	6.8	2.2	5	20	7.8
Average selected subset size		13.63	11.47	3.89	5.92	8.48	7.84

Table 6
Running times of SAOLA, group-SAOLA, OSFS, fast-OSFS, Alpha-investing and OSFSMI methods in seconds.

	Dataset	SAOLA	group-SAOLA	OSFS	Fast-OSFS	Alpha-investing	OSFSMI
1	ALLAML	3.33	3.38	5.21	2.28	.94	4.75
2	GLIOMA	3.38	4.86	8.21	.07	.43	11.41
3	Prostate_GE	2.29	2.34	3.7	1.88	.77	5.05
4	SMK_CAN_187	6.59	7.22	9.67	5.51	5.98	9.71
5	ORL	3.06	1.00	3.69	1.02	.11	3.28
6	GLA-BRA-180	25.24	39.91	68.5	20.21	33.12	65.65
7	OHSUMED-F	.62	1.15	715.89	84.7	120.15	.72
8	Arrhythmia	.13	.45	.35	.12	.05	.14
9	MLL	10.98	13.35	22.94	5.42	2.78	22.55
19	Carcinom	8.1	9.51	57.18	7.11	2.22	13.19
11	Arcene	5.28	4.98	9.33	3.42	2.07	.77
12	Madelon	.16	.23	.43	.23	.08	.17
13	orlraws10P	3.2	3.07	3.2	2.74	1.88	4.86
14	pixraw10P	4.49	7.81	7.63	4.04	1.71	14.93
15	warpAR10P	.68	.95	.73	.66	.21	.77
16	warpPIE10P	.81	1.34	1.36	.83	.29	1.41
17	Yale	.3	.34	.35	.28	.09	.41
18	Extended Yale	.53	.74	1.96	.67	2.41	2.68
19	CINA0	.27	.41	1101.85	86.38	4.40	.87
20	Breast	8.33	7.11	7.01	6.44	10.23	8.02
21	SRBCT	1.06	1.41	2.01	.86	.21	1.28
22	CNS	1.93	2.02	1.79	1.80	1.02	1.92
23	lung	8.66	11.29	60.28	7.80	3.30	17.95
24	Dexter	2.25	2.31	12.31	3.29	47.91	2.09
25	Lymphoma	5.08	5.46	8.84	2.35	.47	8.92
26	Ovarian	9.87	9.96	37.73	10.17	8.42	30.49
Average running time		4.48	5.48	82.77	10.01	9.73	9.13

accuracy and average running time were reported for 5 independent runs, the obtained results are reported in Tables 7–10. It can be seen that OSFSMI-k is still very competitive with all univariate and multivariate filter-based feature selection methods and in most cases it achieved the best or the second best results. Note that unlike batch feature selection methods, OSFSMI-k does not require to access the whole feature space in its process. Table 7 summarizes the average classification accuracy (in%) of OSFSMI-k compared to univariate filter methods (IG, FS, SU, GI and TV) using KNN and DT classifiers. It is clear from the results that in most cases OSFSMI-k obtained the best classification accuracy. Furthermore, Friedman ranks of OSFSMI-k for KNN and DT classifiers were respectively 1.08 and 1.33, which were the best compared to all the other feature selection methods.

In another comparison, several experiments were performed to compare the performance of OSFSMI-k with multivariate filter methods using KNN and DT classifiers and the results are reported in Table 8. The results indicate that in most cases OSFSMI-k archived the highest classification accuracy compared to others. The average classification accuracy and Friedman test over all datasets are reported in Table 8, indicating that OSFSMI-k method gained the highest average classification accuracy among all other multivari-

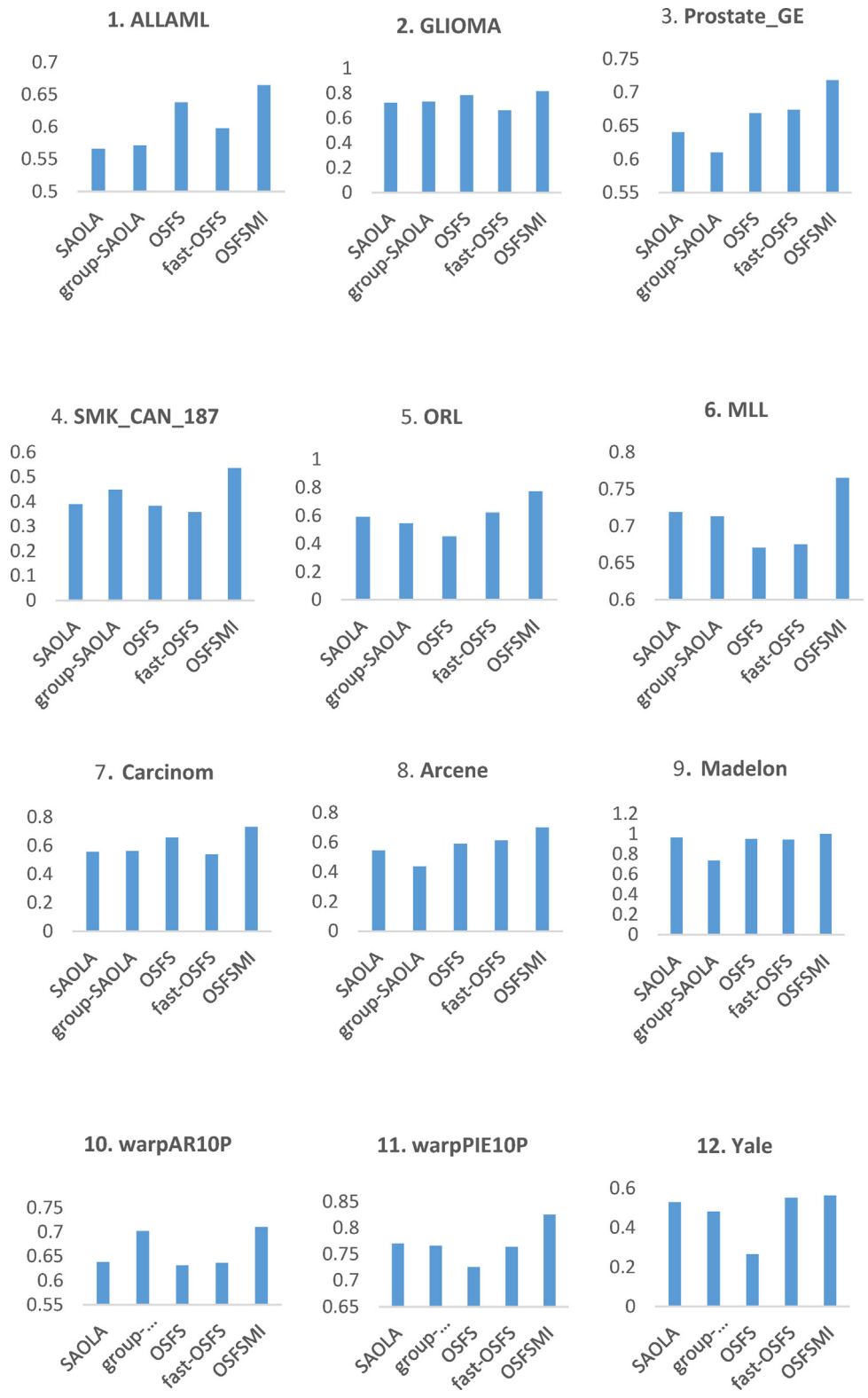

Fig. 2. The stability of SAOLA, group-SAOLA, OSFS, fast-OSFS and OSFSMI methods over different datasets.

ate filter methods. Also, based on Friedman test, the rank of the proposed method was 2.50 which is the best among all the methods

The execution time of OSFSMI-k (in seconds) is compared with univariate and multivariate feature selection methods and the obtained results are reported in Tables 9 and 10, respectively. Table 9 shows that the proposed method is slower than univariate methods, which is due to the fact that the proposed method needs to compute both relevancy and redundancy values while univariate methods do not consider the dependency between selected features. On the other hand, the proposed method considers the dependency between features and chooses more effective features and thus in needs a little more execution time compared to univariate methods.

Table 7
Classification accuracy of OSFSMI-k compare to univariate filter feature selection methods using KNN and DT classifiers.

		KNN classifier						DT classifier					
		IG	FS	SU	GI	TV	OSMI	IG	FS	SU	GI	TV	OSFSMI-k
Colon	6	51.04	**72.91**	51.04	75	66.66	77.08	58.33	67.70	58.33	71.87	65.62	**73.95**
	8	63.88	73.61	63.88	79.16	61.11	**84.72**	65.27	73.61	65.27	**81.94**	61.11	76.38
	10	56.94	75	56.94	**83.33**	61.11	80.55	72.22	72.22	72.22	76.38	69.44	**80.55**
	12	63.54	68.75	63.54	75	69.79	**75.2**	67.70	71.87	67.70	72.91	60.41	**76.04**
Lung	6	53.70	54.62	53.70	52.77	52.77	**70.37**	37.03	43.51	37.03	44.44	43.51	**51.85**
	8	56.4	55.55	56.48	58.33	50	**61.11**	50	51.85	50	50.92	42.59	**54.62**
	10	65.74	58.33	65.74	69.44	63.88	**73.14**	46.29	42.59	46.29	50	39.81	**50**
	12	60.18	61.11	60.18	62.96	64.81	**70.37**	47.22	41.66	47.22	50.92	48.14	**50.92**
Nci9	6	25	31.25	25	21.25	22.5	**35**	22.5	**35**	22.5	18.75	25	28.77
	8	32.5	30	32.5	21.25	17.5	**41.25**	27.5	**37.5**	27.5	25	26.25	35
	10	32.5	37.5	32.5	26.25	30	**43.75**	21.25	37.5	21.25	21.25	20	**38.75**
	12	37.5	27.5	37.5	21.25	28.75	**40**	27.5	26.25	27.5	20	21	**30**
Average		49.91	53.84	49.91	53.83	49.07	62.71	45.23	50.10	45.23	48.69	43.57	53.90
Friedman rank		4.12	3.58	4.04	3.54	4.62	1.08	4.25	3.12	4.25	3.08	4.95	1.33

Table 8
Classification accuracy of OSFSMI-k compared to multivariate filter feature selection methods using KNN and DT classifiers.

		KNN							DT						
		MIFS	MRMR	NMIFS	NJMIM	MIFS-U	DISR	OSFSMI-k	MIFS	MRMR	NMIFS	NJMIM	MIFS-U	DISR	OSFSMI-k
Colon	6	75.08	80	79.16	77.5	80	79.16	**80.11**	73.33	67.5	68.33	**78.33**	74.16	**76.66**	**76.66**
	8	80.54	75.83	79.16	61.66	81.66	76.66	**82.66**	**80**	**78.33**	75	68.33	71.66	72.5	71
	10	**80.21**	79.16	77.5	70.83	79.16	80	80	70	72.5	**72.5**	70	**72.5**	**73.33**	**72.5**
	12	77.5	80.33	77.5	68.33	78	80	**83.33**	77.5	73.83	**78.33**	73.33	73.33	**78.54**	74.5
Lung	6	59.25	**64.44**	60	52.59	59.25	59.25	**64.44**	42.96	40	**47.4**	**48.14**	39.25	44.44	42.96
	8	71.11	68.88	68.14	50.37	68.14	69.62	**71.85**	**55.55**	**60**	53.33	43.70	49.62	48.14	51.85
	10	68.14	**70.37**	**70.37**	56.29	68.14	66.66	**70.37**	**55.55**	54.07	51.11	40	50.37	53.33	51.11
	12	**77.07**	66.66	65.18	49.62	66.66	65.98	71.11	42.22	44.44	46.66	42.22	50.44	**54.07**	**48.14**
Nci9	6	**53.33**	36.66	43.33	28.33	33.33	40	31.66	**41.66**	33.33	28.33	31.66	28.33	26.66	**35.33**
	8	**46.21**	37.58	39.69	28.1	32.22	42.88	37.42	34.55	28.55	31.25	**34.65**	34.88	33.77	**35.74**
	10	34.25	39.08	34.54	38.46	34.06	38.12	**34.75**	30.56	29.55	24.15	28.25	29.85	**33.54**	**31.22**
	12	34.33	46.2	**52.78**	26.84	48	42	46.98	32.45	34.65	39.11	26.59	**39.54**	31.21	**39.85**
Averag		63.08	62.09	62.27	50.74	60.71	61.69	62.89	53.027	51.39	51.29	48.76	51.16	52.18	52.57
Fidman		3.58	3.25	3.95	6.50	4.2	3.91	2.50	3.54	4.20	4.12	5.20	4.37	3.37	3.16

Table 9
Execution times comparison (in second) of IG, FS, SU, GI, TV and OSFSMI-k methods.

		IG	FS	SU	GI	TV	OSFSMI-k
Colon	K = 6	4.40	.23	3.59	.51	.18	9.38
Tumor	K = 8	3.58	.21	3.66	.5	.17	12.05
	K = 10	3.7	.22	3.77	.5	.17	15.62
	K = 12	3.65	.21	3.56	.50	.19	17.55
Lung	K = 6	.55	.17	.55	.1	.02	1.30
	K = 8	.55	.17	.58	.11	.021	1.71
	K = 10	.58	.17	.57	.10	.02	2.12
	K = 12	.56	.16	.57	.11	.02	2.41
Nci9	K = 6	17.23	6.84	17.09	3.95	.75	42.64
	K = 8	16.63	7.08	17.41	3.93	.77	54.85
	K = 10	17.6	6.74	17.29	3.91	.74	70.05
	K = 12	17.01	7.31	17.62	4.03	.75	84.68
Average		7.17	2.45	7.18	1.52	.31	26.19

Table 10 shows that the average running time of the proposed method is much lower than MIFS, NJMIM and DISR methods, while being close to MRMR, NMIFS and MIFS-U. Indeed, it is a trade-off between the classification accuracy and the running time. For example, because of computing the redundancy values, multivariate methods require higher computational costs compared to the univariate methods. However, multivariate methods achieved higher classification accuracies compared to univariate methods.

4.3. Analyzing the effect of different orders of features

To evaluate the effectiveness of OSFSMI, a number of trials were generated, where each trial represents a random ordering of the features in the input feature set. Then, OSFSMI was applied on each randomized trial and the corresponding results are reported in Figs. 3–5. In these results, x-axis represents the trials and the y-axis represents classification accuracy. The results are reported on four datasets including Madelon, ALLAML, ORL and SMK-C-AN-187.

Table 10
Execution time compassion (in second) of MIFS, MRR, NMIFS, NJMIM, MIFS-U, DISR and OSFSMI-k methods.

		MIFS	MRMR	NMIFS	NJMIM	MIFS-U	DISR	OSFSMI-k
Colon	K = 6	6.64	6.89	7.03	27.31	6.78	24.99	8.09
Tumor	K = 8	9.32	9.29	9.81	37.78	9.26	34.95	10.39
	K = 10	11.11	11.3	11.76	45.95	11.35	43.34	12.43
	K = 12	19.77	20.24	21.01	81.09	20.56	76.56	21.55
Lung	K = 6	1.06	1.10	1.15	4.83	1.14	4.09	1.26
	K = 8	1.44	1.50	1.54	6.68	1.48	5.68	1.64
	K = 10	1.81	1.83	1.93	8.47	1.86	7.17	2.02
	K = 12	2.16	2.22	2.31	2.16	2.22	8.68	2.37
Nci9	K = 6	66.7	51.62	53.45	206.19	47.43	181.28	57.00
	K = 8	74.71	52.81	55.33	215.95	51.88	197.83	57.95
	K = 10	74.30	56.71	59.03	235.43	55.79	224.49	65.79
	K = 12	128.39	84.51	89.70	331	80.71	315.25	82.12
Average		33.11	25.00	26.17	100.23	24.20	93.69	26.88

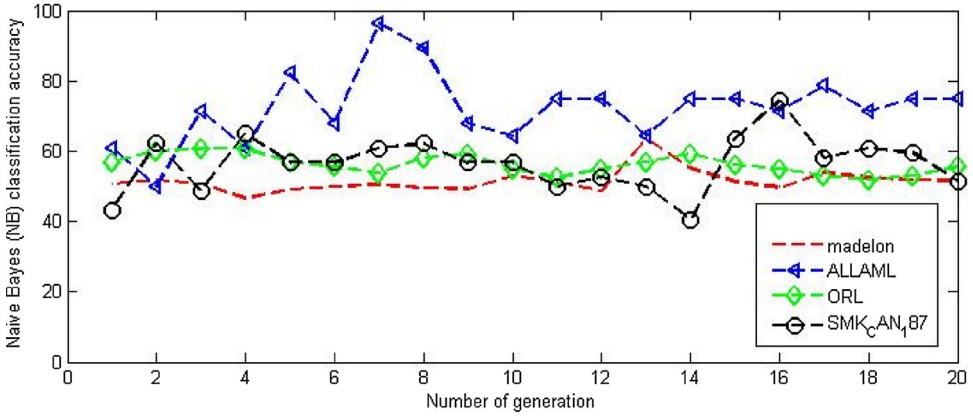

Fig. 3. Classification accuracy on varied order of features using NB classifier.

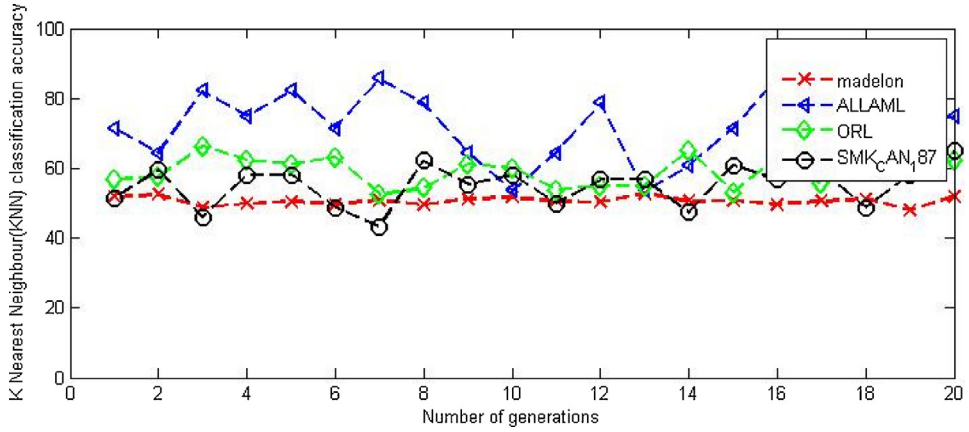

Fig. 4. Classification accuracy on varied order of features using KNN classifier.

The results reveal that different orders of the incoming features do not affect much the final outcomes.

4.4. Discussion

Here we proposed two novel methods for online streaming feature selection, and compared their performance with a number of state-of-the-art traditional and streaming feature selection methods. The results depict that the proposed methods perform better than both online and batch feature selection methods in terms of classification accuracy, size of selected set, running time and stability of results. From the results obtained by the performed experiments, the following interesting points summarized:

- One has to make some trade-offs between running time, number of selected features and classification accuracy, when designing a feature selection method. Our proposed methods consider this and the experimental results indicate that it can identify informative features in an acceptable time.
- From Tables 2–4, it can be concluded that the classification accuracy of the proposed OSFSMI method is higher than state-of-the-art online feature selection methods (i.e., SAOLA, group-SAOLA, OSFS, fast-OSFS and Alpha-investing) using different classifiers and datasets. Also, the results of Table 5 reveal that OSFSMI identifies fewer number of features compared to online feature selection methods excepts OSFS and fast-OSFS. The results of Table 6 show that the proposed method is faster

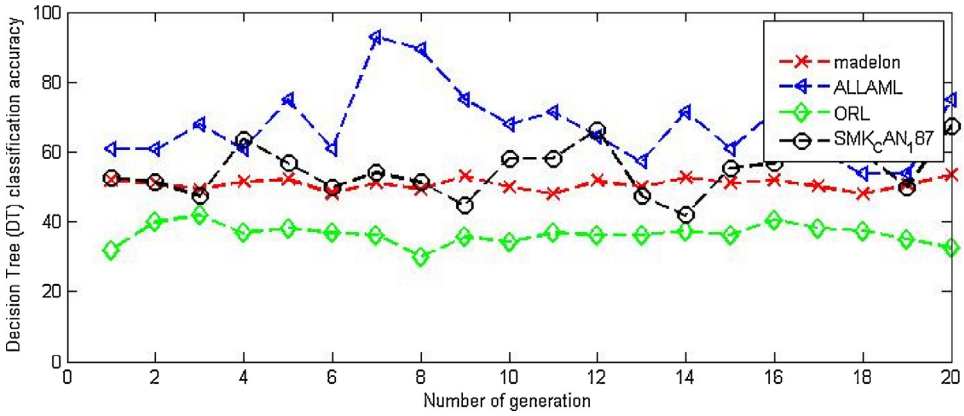

Fig. 5. Classification accuracy on varied order of features using DT classifier.

than OSFS, fast-OSFS and Alpha-investing, while having comparable run time to SAOLA and group-SAOLA methods. Additionally, based on Fig. 2, the stability of OSFSMI is higher than SAOLA, group-SAOLA, OSFS, fast-OSFS in all cases. Since Alpha-investing always chooses one or two features without removing previously selected features, it is more stable than OSFSMI method. Form all the reported results it can be concluded that the proposed methods make a trade-off between classification accuracy, running time, stability and the size of identified feature set. Additionally, since time complexity of the proposed method is linear in terms of the number of selected features, we can conclude that OSFSMI is more scalable than OSFS, fast-OSFS and Alpha-investing.

- The performance of the propose OSFSMI-k method is compared with univariate filter method, and the results (Tables 7 and 8) indicate that the classification accuracy of OSFMI-k method is higher than all univariate filter methods.
- OSFSMI-k is compared with multivariate filter methods and the results (Tables 9 and 10) indicate that it gained higher classification accuracy compared to all the other multivariate filter methods excepts MIFS. Furthermore, the running time of the proposed method is much lower than MIFS, NJMIM, DISR and very close to MRMR, NMIFS and MIFS-U algorithms.
- The results show that the various orders of incoming features do not havesignificant impact on the classification accuracy. This is due to the fact that the proposed methods consider most of the features in both relevancy and redundancy analysis. The proposed methods give higher chance to the features to proceed to further steps. Using this strategy, a feature is eliminated from the feature set if there exists at least a similar and effective feature that it in the feature set.

5. Conclusion

Feature selection aims to select informative features by removing redundant and irrelevant features. In this paper two novel feature selection methods, called OSFSMI and OSFSMI-k, were proposed for feature selection of online data streams. The main idea is to use mutual information to compute both relevancy and redundancy of so-far-seen features in an online manner in order to select most informative and non-redundant feature set that are highly relevant to the target class. The search process of the proposed method consists of two main steps. In the first step, the relevancy of each newly arrived feature is evaluated and those of strictly non-relevant features are discarded and the others will be processed in the next step. Then in the second step, a novel measure is used to evaluate the effectiveness of the relevant features by using both relevancy and redundancy concepts through several iterations. In each iteration of this step, those of features that their effectiveness values are lower than the effectiveness of the so-far-seen feature are eliminated from the feature set. Using this strategy, if a feature is eliminated, it is ensured that there is at least a feature with a higher effectiveness value than it. Also, unlike the other methods, by facing with effective features in further steps, those of previously selected features can be eliminated even they are selected in the previous step. Also, the proposed methods do not use any learning algorithm in their search process and thus they can be classified in filter-based methods.To evaluate the efficiency of the proposed algorithms several experiments were performed to compare them with several well-known and state-of-the-art online and offline feature selection methods over 29 frequently used datasets. The methods were compared in terms of different evaluation measures including classification accuracy, running time, stability of the results and the number of identified features. The obtained results showed that in most cases the proposed methods outperformed both offline and online feature selection methods. Also, Friedman statistical test showed that in most cases the proposed method attained the best rank among the other methods.

References

[1] X. Deng, B. Wang, H. Wei, M. Chen, The key data mining models for high dimensional data, in: G. Yang (Ed.), Proceedings of the 2012 International Conference on Communication, Electronics and Automation Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 321–327 (%@ 978-3-642-31698-2).
[2] L. Huan, M. Hiroshi, Computational Methods of Feature Selection (Chapman \\& Hall/Crc Data Mining and Knowledge Discovery Series), Chapman \\& Hall/CRC, 2007.
[3] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[4] J.R. Vergara, P.A. Estévez, A review of feature selection methods based on mutual information, Neural Comput. Appl. 24 (1) (2014) 175–186.
[5]. G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (2014) 16–28.
[6] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, A review of feature selection methods on synthetic data, Knowl. Inf. Syst. 34 (2013) 483–519.
[7] S. Tabakhi, A. Najafi, R. Ranjbar, P. Moradi, Gene selection for microarray data classification using a novel ant colony optimization, Neurocomputing 168 (2015) 1024–1036.
[8] P. Moradi, M. Rostami, A graph theoretic approach for unsupervised feature selection, Eng. Appl. Artif. Intell. 44 (2015) 33–45.
[9] S. Tabakhi, P. Moradi, Relevance-redundancy feature selection based on ant colony optimization, Pattern Recogn. 48 (2015) 2798–2811.
[10] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh, Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing), Springer-Verlag New York Inc., 2006.
[11] A. Moayedikia, R. Jensen, U.K. Wiil, R. Forsati, Weighted bee colony algorithm for discrete optimization problems with application to feature selection, Eng. Appl. Artif. Intell. 44 (2015) 153–167.
[12] J. Zhou, D.P. Foster, R.A. Stine, L.H. Ungar, Streamwise feature selection, J. Mach. Learn. Res. 7 (2006) 1861–1885.
[13] M. Sayed-Mouchaweh, E. Lughofer, Learning in Non-Stationary Environments: Methods and Applications, Springer Publishing Company, Incorporated, 2012.

[14] J. Gama, Knowledge Discovery from Data Streams, Chapman \& Hall/CRC, 2010.
[15] J. Dean, Big Data, Data Mining, and Machine Learning: Value Creation for Business Leaders and Practitioners, CreateSpace Independent Publishing Platform, 2014.
[16] G. Lang, D. Miao, T. Yang, M. Cai, Knowledge reduction of dynamic covering decision information systems when varying covering cardinalities, Inf. Sci. 346 (2016) 236–260.
[17] S. Eskandari, M.M. Javidi, Online streaming feature selection using rough sets, Int. J. Approx. Reasoning 69 (2016) 35–57.
[18] F. Wang, J. Liang, Y. Qian, Attribute reduction: a dimension incremental strategy, Knowl.-Based Syst. 39 (2013) 95–108.
[19] S.C.H. Hoi, J. Wang, P. Zhao, R. Jin, Online feature selection for mining big data, in: Proceedings of the 1 st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, ACM Beijing, China, 2012, pp. 93–100.
[20] F. Hu, G. Wang, H. Huang, Y. Wu, Incremental attribute reduction based on elementary sets, in: D. Ślęzak, G. Wang, M. Szczuka, I. Düntsch, Y. Yao (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing: 10th International Conference, RSFDGrC 2005, Regina, Canada,August 31 – September 3, 2005, Proceedings, Part I, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 185–193.
[21] J. Wang, P. Zhao, S.C.H. Hoi, R. Jin, Online feature selection and its applications, IEEE Trans. Knowl. Data Eng. 26 (2014).
[22] M. Pratama, G. Zhang, M.J. Er, S. Anavatti, An incremental type-2 meta-cognitive extreme learning machine, IEEE Trans. Cybern. 47 (2017) 339–353.
[23] J. Zhou, D. Foster, R. Stine, L. Ungar, Streaming feature selection using alpha-investing, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, ACM, Chicago, Illinois, USA, 2005, pp. 384–393.
[24] K. Yu, X. Wu, W. Ding, J. Pei, Scalable and accurate online feature selection for big data, ACM Trans. Knowl. Discov. Data 11 (2016) 1–39.
[25] L.H. Ungar, J. Zhou, D.P. Foster, B.A. Stine, Streaming feature selection using IIC, in: AI&Statistics'05, 2005, pp. 384–393.
[26] X. Wu, K. Yu, W. Ding, H. Wang, X. Zhu, Online feature selection with streaming features, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1178–1192.
[27] X. Wu, K. Yu, H. Wang, W. Ding, Online streaming feature selection, in: International Conference on Machine Learning, Haifa, Israel, 2010, pp. 1159–1166.
[28] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (2004) 1205–1224.
[29] G. Wang, Q. Song, B. Xu, Y. Zhou, Selecting feature subset for high dimensional data via the propositional FOIL rules, Pattern Recogn. 46 (2013) 199–214.
[30] S. Eskandari, M.M. Javidi, Online streaming feature selection using rough sets, Int. J. Approximate Reasoning 69 (2016) 35–57.
[31] B.C. Ross, Mutual information between discrete and continuous data sets, PLOS ONE 9 (2014) e87357.
[32] G. Brown, A. Pocock, M.-J. Zhao, M. Luján, Conditional likelihood maximisation: a unifying framework for information theoretic feature selection, J. Mach. Learn. Res. 13 (2012) 27–66.
[33] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[34] L.E. Raileanu, K. Stoffel, Theoretical comparison between the gini index and information gain criteria, Ann. Math. Artif. Intell. 41 (2004) 77–93.
[35] N. Kwak, C. Choi, Input feature selection for classification problems, IEEE Trans Neural Netw (2002).
[36] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell 27 (2005) 1226–1238.
[37] E. Lughofer, On-line incremental feature weighting in evolving fuzzy classifiers, Fuzzy Sets Syst. 163 (2011) 1–23.
[38] H. Huang, S. Yoo, S.P. Kasiviswanathan, Unsupervised feature selection on data streams, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, Melbourne, Australia, 2015, pp. 1031–1040.
[39] S. Perkins, K. Lacker, J. Theiler, Grafting: fast incremental feature selection by gradient descent in function space, J. Mach. Learn. Res. 3 (2003) 1333–1356.
[40] M. Pratama, J. Lu, S. Anavatti, E. Lughofer, C.-P. Lim, An incremental meta-cognitive-based scaffolding fuzzy neural network, Neurocomputing 171 (2016) 89–105.
[41] S. Alizadeh, A. Kalhor, H. Jamalabadi, B.N. Araabi, M.N. Ahmadabadi, Online local input selection through evolving heterogeneous fuzzy inference system, IEEE Trans. Fuzzy Syst. 24 (2016) 1364–1377.
[42] S. Perkins, J. Theiler, in: T. Fawcett, N. Mishra (Eds.), Online Feature Selection Using Grafting, ICML, 2003, pp. 592–599.
[43] H. Li, X. Wu, Z. Li, W. Ding, Online group feature selection from feature streams, in: M. desJardins, M.L.E. Littman (Eds.), Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, AAAI Press Bellevue, Washington, 2013.
[44] C.E. Shannon, A mathematical theory of communication, SIGMOBILE Mob. Comput. Commun. Rev. 5 (2001) 3–55.
[45] I.H. Witten, E. Frank, M.A. Hall, in: John M. Chambers (Ed.), Data Mining Practical Machine Learning Tools and Techniques, 3rd Edition, Morgan Kaufmann, 2011.
[46]. T. Lin, H. Li, K. Tsai, Implementing the fisher's discriminant ratio in a k-means clustering algorithm for feature selection and dataset trimming, J. Chem. Inf. Comput. Sci. 44 (2004) 76–87.
[47] D. Zhang, S. Chen, Z.-H. Zhou, Constraint Score A new filter method for feature selection with pairwise constraints, Pattern Recogn. 41 (2008) 1440–1451.
[48] R. Battiti, Using mutual information for selecting features in supervised neural net learning, Trans. Neur. Netw. 5 (1994) 537–550.
[49] M. Bennasar, Y. Hicks, R. Setchi, Feature selection using joint mutual information maximisation, Expert Syst. Appl. 42 (2015) 8520–8532.
[50] P.E. Meyer, G. Bontempi, On the use of variable complementarity for feature selection in cancer classification, in: F. Rothlauf, J. Branke, S. Cagnoni, E. Costa, C. Cotta, R. Drechsler, E. Lutton, P. Machado, J.H. Moore, J. Romero, G.D. Smith, G. Squillero, H. Takagi (Eds.), Applications of Evolutionary Computing: EvoWorkshops 2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, and EvoSTOC, Budapest, Hungary, A Pril 10–12, 2006. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 91–102.
[51] L.T. Vinh, S. Lee, Y.-T. Park, B.J. d'Auriol, A novel feature selection method based on normalized mutual information, Appl. Intell. 37 (2012) 100–120.
[52] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Am. Statistician 46 (1992) 175–185.
[53] P. Domingos, M. Pazzani, On the optimality of the simple bayesian classifier under zero-one loss, Mach. Learn. 29 (1997) 103–130.
[54] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1940) 86–92.
[55] K. Yu, W. Ding, X. Wu, Library of online streaming feature selection, Knowl.-Based Syst. 113 (2016) 1–3.
[56] J.M. Peña, Learning gaussian graphical models of gene networks with false discovery rate control, in: E. Marchiori, J.H. Moore (Eds.), Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics: 6th European Conference, EvoBIO 2008, Naples, Italy,March 26-28, 2008. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 165–176.
[57] L. Yu, C. Ding, S. Loscalzo, Stable feature selection via dense feature groups, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Las Vegas, Nevada, USA, 2008, pp. 803–811.