A survey on online feature selection with streaming features

Article *in* Frontiers of Computer Science (electronic) · August 2017

DOI: 10.1007/s11704-016-5489-3

5 authors, including:

Hu Xuegang
Fujian Medical University
129 PUBLICATIONS **1,920** CITATIONS
SEE PROFILE

Peng Zhou
Anhui University
13 PUBLICATIONS **360** CITATIONS
SEE PROFILE

Peipei Li
Hefei University of Technology
110 PUBLICATIONS **1,940** CITATIONS
SEE PROFILE

Xindong Wu
UVM
332 PUBLICATIONS **23,514** CITATIONS
SEE PROFILE

Frontiers of Computer Science CN 10-1014/TP
DOI 10.1007/s11704-016-5489-3 ISSN 2095-2228
REVIEW ARTICLE

A Survey on Online Feature Selection with Streaming Features

Xuegang HU[1], Peng ZHOU[1], Peipei LI[1], Jing WANG[1], Xingdong WU[2]

1 Hefei University of Technology, Hefei 230009, China
2 University of Louisiana, Louisiana 70504, USA

Front. Comput. Sci., **Just Accepted Manuscript** • 10.1007/s11704-016-5489-3
http://journal.hep.com.cn on August 25, 2016

Front.Comput.Sci.
DOI

REVIEW ARTICLE

A Survey on Online Feature Selection with Streaming Features

Xuegang HU [1], Peng ZHOU [1], Peipei LI [1], Jing WANG [1], Xingdong WU (✉)[2]

1 Hefei University of Technology, Hefei 230009, China
2 University of Louisiana, Louisiana 70504, USA

Abstract In the era of big data, the dimensionality of data is increasing dramatically in many domains. To deal with high dimensionality, online feature selection becomes critical in big data mining. Recently, online selection of dynamic features has received much attention. In situations where features arrive sequentially over time, we need to perform online feature selection upon feature arrivals. Meanwhile, considering grouped features, it is necessary to deal with features arriving by groups. To handle these challenges, some state-of-the-art methods for online feature selection have been proposed. In this paper, we first give a brief review of traditional feature selection approaches. Then we discuss specific problems of online feature selection with feature streams in detail. A comprehensive review of existing online feature selection methods is presented by comparing with each other. Finally, we discuss several open issues in online feature selection.

Keywords Big data, Feature selection, Online feature selection, Feature stream.

1 Introduction

We are living in the "Big Data era" [1]. Data with large volumes and high dimensionality are ubiquitous in many domains, such as geometrics, computer vision, social media and so forth. For instance, Flickr, as a public picture sharing website, receives 55.8 million photos per month and 1.83 million photos per day, on average in 2014 [2]. Assuming the size of each photo is 2 megabytes, this requires 3.66 terabyte (TB) storage every single day. At the same time, the dimensionality of data is also extremely high in some applications. For example, in complementary DNA microarray experiments, the total number of positive and negative samples is usually no more than 100, but the number of genes to be selected is usually 6,000 to 60,000 [3]. Moreover, the Web Spam Corpus 2011, is a collection of approximately 330,000 spam web pages and 16,000,000 features (attributes) [4]. Thus, how to process and get valuable information from massive and high-dimensional data has become a great challenge [5, 6].

Feature selection is one of the most important techniques in data mining and machine learning, and plays a critical role in dealing with big data problems [7]. The task of feature selection is to select a subset of relevant features for building effective prediction models. Feature selection can generate many potential benefits, such as reducing the storage requirements, saving training and modeling times, improving the prediction performance, providing a better data understanding and so on [8]. Traditional feature selection methods assume that all features are presented to a learner before feature selection takes place. For example, mRMR (minimal Redundancy and Maximal Relevance) [9] in the principle of max-dependency, max-relevance and min-redundancy, is a representative algorithm base on mutual information. It aims to find a subset, in which the features are with large dependency on the target class and with low redundancy among each other.

Meanwhile, in real-world applications, not all features can be presented before learning. For example, in image analysis [10], multiple descriptors are exacted dynamically to capture various visual information of images, including HOG (Histogram of Oriented Gradients), color histogram

Received month dd, yyyy; accepted month dd, yyyy

E-mail: *xwu@louisiana.edu*

and SIFT (Scale Invariant Feature Transform). Each of these descriptors is generated independently. It is hence very time-consuming or even unrealistic to wait until all features are generated. Meanwhile, Mars crater detection from high resolution planetary images is another real-world application example [11]. Tens of thousands of texture-based features in different scales and different resolutions can potentially be generated from high resolution planetary images. It is infeasible and time-consuming, if we are waiting for the texture features to be generated through planetary images until we have a near global coverage of the Martian surface. Thus, it is necessary to perform feature selection as the arrivals of features, called online feature selection with streaming features [12].

Online feature selection with streaming features is one branch of online feature selection [13]. It is designed to deal with feature selection without the knowledge of an entire feature space [12], that is, we cannot afford waiting until all features arrive before learning. This category of online feature selection assumes the number of data instances is fixed and the number of features changes over time. The representative works include the approaches of Grafting [14], Alpha-investing [15] and OSFS (Online Streaming Feature Selection) [16]. Meanwhile, to address the challenges of online feature selection for extremely high-dimensional data for big data, a Scalable and Accurate OnLine Approach for feature selection called SAOLA was proposed [13]. A common assumption in the aforementioned approaches lies that features are generated one by one. However, in real-world applications, features can also be generated by groups. For instance, the descriptor of each image consists of a group of features instead of an individual feature [10]. Correspondingly, many approaches of online group feature selection with streaming features have been proposed, such as GFSSF (Group Feature Selection with Streaming Features) [17] and OGFS (Online Group Feature Selection) [18]. More precisely, GFSSF can work at both the group and individual feature levels by exploiting entropy and mutual information in information theories. OGFS selects a significant feature by its distinguishing capability and reduces redundancy by a regression model.

Methods

The other branch of online feature selection assumes that the number of features on training data is fixed while the number of data instances changes over time, called online feature selection with data streams [18]. Related work includes [19–22]. In object tracking [20], success or failure depends primarily on how distinguishable an object is from its surroundings. It is most important to select a feature space that can discriminate objects and their background. As the foreground and background appearances are constantly changing, the key is online and adaptive selection of an appropriate feature space for tracking. Another example is in CBIR (Content-Based Image Retrieval) [22], the online learning process must solve a fundamental problem: which features are more representative for explaining the current query concept than the others.

Online feature selection with data streams is closely related to the data mining on data streams. More details can be found in data streaming mining [23]. However, this is beyond the scope of our paper. In this paper we only focus on online feature selection with streaming features.

The rest of the paper is organized as follows. Section 2 reviews traditional feature selection approaches. Section 3 first introduces several approaches of online feature selection and then gives the analysis. Section 4 first summarizes the benchmark data sets and evaluation measures, introduces an open experimental tool of online streaming feature selection, and then presents the experiment analysis of several representative online feature selection algorithms mentioned in this paper. Finally, Section 5 discusses some challenging issues for online feature selection and Section 6 concludes this paper.

2 Feature Selection

In this section, we first give the formalization of traditional feature selection. Let X represent the data set, denoted as $X = [x_1, x_2, ..., x_n]^T \in R^{n \times d}$ consisting of n samples (columns) over a d-dimensional feature space $F = [f_1, f_2, ..., f_d]^T \in R^d$. Let $C = [c_1, c_2, ..., c_m]^T \in R^m$ denote the class label vector that has m distinct class labels. The task of feature selection is to select a subset of features for F that can be used to derive a mapping function from x to c that is "as good as possible" according to some criterion.

Feature selection is an important technique which can be used in many real-world applications [24, 25]. Finding an optimal feature subset is usually arduous, and many problems related to feature selection have been proved to be NP-Hard [8]. A standard feature selection process consists of four basic steps, namely subset generation, subset evaluation, stopping criterion verification, and result validation [26].

According to how the label information is used, feature selection algorithms can be divided into supervised [27],

unsupervised [28] and semi-supervised [29, 30] ones. Supervised learning deals with the scenario that class labels of the data are known or they can be calculated. It can get a small subset and high accuracy. However, in practical applications, we do not always know all of the class labels of operational data or only know class labels of a few operational data. Correspondingly, unsupervised learning and semi-supervised learning have been proposed. Without knowledge of class labels, unsupervised learning tries to discover natural groupings in a set of objects. Since the examples for learning are unlabeled, there is no error or reward signal to evaluate a potential solution. Semi-supervised learning uses both labeled data and unlabeled data to modify a hypothesis obtained from labeled data alone.

Considering whether using a classifier or not in feature selection, we can further divide feature selection algorithms into the following three categories: filter, wrapper and embedded models [7].

Filter models can have a high efficiency in feature selection and evaluate features without utilizing any classification algorithms [31]. They evaluate the features by a certain criterion and select features by ranking their evaluation values [26]. Within the filter models, different feature selection approaches can further be categorized into two groups: feature ranking and subset search [32]. Feature ranking methods evaluate the goodness of features individually and obtain a ranking list of selected features ordered by their goodness. Laplacian Score [33] and Fisher Score [34] are two representative feature ranking methods. Laplacian Score is unsupervised and evaluates a feature by its power of locality preserving. Fisher Score is supervised and seeks feature subsets which preserve the discriminative ability of a classifier. Meanwhile, there are a series of feature selection methods based on Constraint Score [35–39], that is, by using pairwise constraints, they specify whether a pair of data samples belong to the same class (must-link constraints) or different classes (cannot-link constraints), they do not have to access the whole training data, and have a computational advantage on large-size data sets. Subset search methods evaluate the goodness of each candidate feature subset and select the optimal one according to some evaluation measures [40–42]. Some popular criteria include distance measures [43, 44], information measures, dependency measures, and consistency measures [45, 46].

Wrapper models depend on specific machine learning algorithms [47]. They need a search strategy to search the space of all possible feature subsets and employ a specific classifier to evaluate a subset directly. The popular search strategies used in wrapper models include Best-First, Branch and Bound, Simulated Annealing, Genetic Algorithms and so on [8]. The popular classifiers include Decision Trees, Naive Bayes, Least-square Linear Predictors and Support Vector Machines. They use the performance of the learning algorithms conducted on the selected subset to determine which features are selected. Meanwhile they evaluate the prediction accuracy of the target feature subset by cross validations on the training set. Thus, these methods are slower than filter methods on running speed, but they usually give superior performance and a smaller subset [48]. It is hence very conducive to identify the key features.

Embedded models are different form filter and wrapper models and they usually seek the subset by jointly minimizing empirical error and penalty [26]. Embedded approaches attempt to maximize classification performance and minimize the number of features used in a classification or regression model. Embedded approaches are independent of the classifier and do not need the cross-validation step, therefore they are computationally efficient. Thus, embedded models have the advantages of both wrapper models and filter models [49]. There are many approaches in embedded models, such as LASSO (least absolute shrinkage and selection operator) [50], LARS (least angle regression) [51], elastic net [52] and so on [53–55].

3 Online Feature Selection

Traditional feature selection mentioned above assumes that all candidate features are available before learning starts. However, in many real-world applications [10, 11], features are generated dynamically, and arrive one by one or group by group. It is hence not practical to wait until all features have been generated before feature selection begins. This poses great challenges to traditional feature selection approaches, called online feature selection with streaming features. We first give the formalization of online feature selection with streaming features as follows.

Let X represent the data set, denoted as $X = [x_1, x_2, ..., x_n]^T \in R^{n\times d}$ consisting of n samples (columns) over a d-dimensional feature space $F = [f_1, f_2, ..., f_d]^T \in R^d$ and let $C = [c_1, c_2, ..., c_m]^T \in R^m$ denote the class label vector. At each time j, we just get feature f_j of X and we do not know the exact number of d in advance. So the problem is to derive a x to c mapping at each time step, which is as good as possible using a subset

of the features that have arrived so far.

Considering the characteristics of a feature stream, many online feature selection approaches, including online individual feature selection approaches and online group feature selection approaches have been proposed to address this problem and they are proven to be effective and efficient with streaming features. In the following subsections, we first introduce online individual feature selection and summarize the related work, and then discuss online group feature selection. Finally, we analyze these approaches.

3.1 Online Individual Feature Selection

Online individual feature selection shares the common assumption that candidate features are generated dynamically and arrive one at a time. More specifically, Perkins and Theiler [14] considered the online feature selection problem and proposed the Grafting algorithm based on a stagewise gradient descent approach. Zhou et al. [15] proposed two algorithms of information-investing and alpha-investing, based on streamwise regression for online feature selection. Wu et al. [16] presented an online streaming feature selection framework with two algorithms OSFS (Online Streaming Feature Selection) and fast-OSFS. Wang et al. [56] proposed the algorithm OFS (Online Feature Selection) learning with full inputs and OFSp (Online Feature Selection with Partial inputs) learning with partial inputs. Yu et al. [13] proposed the SAOLA (a Scalable and Accurate OnLine Approach) for feature selection. More details are as follows.

3.1.1 Grafting

Grafting [14] treats feature selection as an integral part of learning a predictor within a regularized framework. It is oriented to binomial classification. The objective function is a binomial negative log-likelihood loss (BNLL) function as shown in Eq. (1).

$$\frac{1}{n}\sum_{i=1}^{n} ln(1+e^{-y_i f(x_i)}) + \lambda \sum_{j=1}^{k} \|w_j\|_1 \tag{1}$$

where n is the number of samples, k is the number of selected features so far, λ is a regularization coefficient, and w is the weight vector subject to $l_1$regularization. When non-zero weights w_j add to the model, $\lambda\|w_j\|$ is penalized. So, it can only add weights to the model if the reduction in the mean loss $\overline{L}$ outweighs the regularize penalty. This means, feature f_j can be selected if the following condition is satisfied:

$$\left|\frac{\partial \overline{L}}{\partial w_j}\right| > \lambda \tag{2}$$

Grafting operates in an incremental iterative fashion, and it gradually builds up a feature set while training a predictor model using gradient descent. At each iteration, a fast gradient-based heuristic is used to identify a feature that is most likely to improve the existing model. In order to determine whether feature f_j can be selected, it needs to select the value of a regularization parameter λ in advance. If a new feature f_j is selected, the algorithm repeats and reapplies the gradient test to all the selected features. Grafting can be used with both linear and non-linear predictor model classes, and it can be used for both classification and regression.

3.1.2 Alpha-investing

The Alpha-investing [15] method does not need a global model and it is one of the penalized likelihood ratio methods. When a feature f_j arrives, it is evaluated by the p-value. The p-value presents the probability whether the feature could be accepted or not. This algorithm uses a threshold α_j to measure the p-value of f_j. If the p-value of feature f_j is bigger than α_j , it will be added to the model.

Meanwhile, the threshold α_j can be adaptively adjusted each time no matter whether feature f_j is selected or not. When f_j is selected, the value of w_j will increase as shown in Eq.(3).

$$w_{j+1} = w_j + \Delta\alpha - \alpha_j \tag{3}$$

where $\Delta\alpha$ is the parameter controlling the false discovery rate and w_j represents the current acceptable number of future false positives. Otherwise, if f_j is discarded, w_j will decrease as shown in Eq. (4).

$$w_{j+1} = w_j - \alpha_j \tag{4}$$

where α_j is set to the value of $w_j/(2 \times j)$.

In sum, Alpha-investing can handle unknown or even infinite sizes of candidate feature sets. Because it does not reevaluate the selected features, it hence performs efficiently, but it will probably perform ineffectively in the subsequent feature selection for never evaluating the redundancy of selected features.

3.1.3 OSFS

OSFS [16] (Online Streaming Feature Selection) provides a framework for streaming feature selection. OSFS divides

features into three disjoint categories: strongly relevant, weakly relevant and irrelevant. First of all, we will give the definitions of these three types of feature relevance.

Definition 1. *[Strong relevance] A feature f_i is strongly relevant to C iff $\forall S \subseteq F - f_i$ s.t. $P(C|S) \neq P(C|S, f_i)$.*

Definition 2. *[Weak relevance] A feature f_i is weakly relevant to C iff it is not strongly relevant, and $\exists S \subset F - f_i$ s.t. $P(C|S) \neq P(C|S, f_i)$.*

Definition 3. *[Irrelevance] A feature f_i is irrelevant to C iff it is neither strongly nor weakly relevant, and $\forall S \subseteq F - f_i$ s.t. $P(C|S) = P(C|S, f_i)$.*

The OSFS framework contains two major steps: online relevance analysis and online redundancy analysis.

- Online relevance analysis

In this step, features are divided into three categories: strong relevance, weak relevance and irrelevance. When a new feature arrives, the algorithm calculates its relevance to the class attribute. If the feature is a strongly or weakly relevant feature, it will be added to the feature subset. If the feature is irrelevant, it will be discarded. Once a new feature is selected, it will turn to online redundancy analysis.

- Online redundancy analysis

In this step, the algorithm dynamically identifies and eliminates redundant features. Let BCF denote the set of the best candidate features so far. After a new feature is included into BCF, if a subset exists in BCF which can make any existing feature in BCF and the class attribute C conditionally independent, then the newly selected feature is redundant and will be removed from BCF. The redundancy analysis will guarantee that the newly added feature is an optimal selection for global selected features.

The most time-consuming part of OSFS is the redundancy analysis. When the selected feature set is large, this process will become very inefficient and lead to a poor performance.

Fast-OSFS divides the online redundancy analysis into two parts:

1) Determining whether an incoming new feature is redundant. This part aims to remove a new relevant but redundant feature. If the new feature is removed successfully, Fast-OSFS will deal with the next arriving feature.

2) Identifying which of the selected features may become redundant by using the Markov blanket theory once the new feature is added. In this part, Fast-OSFS reduces the computational cost by only considering the subsets within BCF that contains the new added feature instead of all subsets within BCF.

3.1.4 SAOLA

SAOLA (Scalable and Accurate OnLine Approach) [13] addresses two challenges in big data applications: extremely high dimensionality and its highly scalable requirement of feature selection. SAOLA employs novel online pairwise comparison techniques and maintains a parsimonious model over time in an online manner. This method is scalable on data sets of extremely high dimensionality. SAOLA employs a theorem as follows:

Theorem 1. *With the current feature subset $S^*_{t_{i-1}}$ at time t_{i-1} and a new feature f_i at time t_i, $\exists Y \in S^*_{t_{i-1}}$, if $I(f_i; C|Y) = 0$ holds, then the following is achieved.*

$$I(Y; C) > I(f_i; C) \text{ and } I(f_i; Y) \geqslant I(f_i; C) \tag{5}$$

where C is the class attribute and $I(A; B)$ means the mutual information between A and B.

Meanwhile, there is an important equation which is used in the algorithm as follow:

$$I(f_i; C) > I(Y; C) \text{ and } I(Y; f_i) \geqslant I(Y; C) \tag{6}$$

The SAOLA algorithm can be divided into three steps:

- Step 1: When a new feature f_i arrives at time t_i, calculate the mutual information $I(f_i; C)$ and compare to the threshold δ. If $I(f_i; C) < \delta$, the feature f_i will be discarded as an irrelevant feature. If not, go to step 2.
- Step 2: Evaluate whether f_i should be kept given the current feature set $S^*_{t_{i-1}}$. If the new feature f_i satisfies Theorem 1, it will be discarded and never considered again. If not go to step 3.
- Step 3: Once f_i is added to $S^*_{t_{i-1}}$, the current feature set will be checked whether some features within it can be removed. If $\exists Y \in S^*_{t_{i-1}}$ such that Eq.(6) holds, Y is removed.

The SAOLA algorithm performs a set of pairwise comparisons between individual features instead of conditioning on a set of features. This reduces the computational cost. Meanwhile, SAOLA employs a k-greedy search strategy to find redundant features by checking feature subsets for each feature in the current feature set $S^*_{t_{i-1}}$ which make it faster than Fast-OSFS.

3.1.5 OFS

OFS (Online Feature Selection) [56] deals with sequential training data of high dimensionality which is different from the above approaches. In the above works, features are assumed to arrive one by one while all the training instances are assumed to be available before the learning process starts. OFS considers the problem that the training instances arrive sequentially.

OFS considers the problem of online feature selection for binary classification. Let $\{(x_i, y_i)|i = 1, 2, ..., n\}$ be a sequence of input patterns received over the trials, where each $x_i \in R^d$ is a vector of d dimension and $y_i \in \{-1, +1\}$. If d is a large number, it needs to select a relatively small number of features for linear classification. In each trial i, the learner presents a classifier $L_i \in R^d$ that will be used to classify instance x_i by a linear function $sgn(L_i^T x_i)$. It requires the classifier L_i to have at most B ($B > 0$ is a predefined constant) nonzero elements instead of all the features for classification. The goal of OFS is to design an effective strategy that can make a small number of mistakes.

OFS (learning with full inputs) assumes that the learner is provided with full inputs of every training instance. The goal of OFS by learning with full inputs is to efficiently identify a fixed number of relevant features for accurate prediction. When a training instance (x_i, y_i) is misclassified, the classifier L_i will be updated by online gradient descent first. Then the classifier will be projected to a l_2 ball to ensure the norm of the classifier is bounded. OFS will simply keep the B elements in the resulting classifier L_{i+1} with the largest absolute weights, if the classifier L_{i+1} has more than B nonzero elements.

For real-world applications, the attributes of objects might be expensive to acquire. To address this challenge, OFS has another version, OFSp (learning partial inputs). To tradeoff exploration and exploitation, OFSp aims at performing online feature selection with partial input information by employing a classic technique. It spends $\varepsilon\%$ of trials for exploration by randomly choosing B attributes from all d attributes. Meanwhile, the remaining $(1 - \varepsilon)\%$ of trials on exploitation by choosing the B attributes for which classifier L_i has nonzero values.

In sum, all the approaches mentioned above can deal with the scenarios of features arriving one by one. However, in real-world applications, features possess certain group structures. It is very likely to also have scenarios of streaming group feature selection, because the features are generated and arrived group by group. It is therefore also to select features at group level when group structures exist. In the next section, we summarize several online group feature selection approaches.

3.2 Online Group Feature Selection

Online group feature selection aims to select features at the group level. It is a new processing model. Let X represent the data set, denoted as $X = [x_1, x_2, ..., x_n]^T \in R^{n\times d}$ consisting of n samples (columns) over a d-dimensional feature space $F = [f_1, f_2, ..., f_d]^T \in R^d$ and let $C = [c_1, c_2, ..., c_m]^T \in R^m$ denote the class label vector. Assume $G = \{G_1, G_2, ..., G_n\}$ represents n feature groups without overlapping, and G_i denotes the i^{th} feature group. The challenge of online group feature selection is how to simultaneously optimize selections within each group as well as between those groups to achieve a set of groups Γ_{t_i} that maximizes its predictive performance.

Several representative works include GFSSF (Group Feature Selection with Streaming Features), OGFS (Online Group Feature Selection) and group-SAOLA. GFSSF [17] is an algorithm proposed for group feature selection with streaming features and it can work at both the group and individual feature levels. OGFS [18] formulates the problem as online group feature selection, which can be divided into two stages: intra-group selection and inter-group selection. Group-SAOLA [57] is an extension of SAOLA, and it selects feature groups which are sparse at the levels of both features and groups simultaneously in an online manner. More details are below.

3.2.1 GFSSF

GFSSF (Group Feature Selection with Streaming Features) [17] performs group feature selection with streaming features. Meanwhile, GFSSF can work at both the group and individual feature levels for streaming feature selection by exploiting entropy and mutual information in information theories. GFSSF consists of the feature level and group level selections. The approach selects features by using the relationship between features. First of all, we will give the definition of irrelevance, redundancy and coverage between features. Let $I(X; Y)$ denote the mutual information between X and Y.

Definition 4. *[Irrelevance] Given two features X and Y, X is irrelevant to Y if and only if* $I(X; Y) = 0$.

Definition 5. *[Redundancy] Given two features X and Y, and a set of features F, X is redundant to F for Y if and only if*

$I((X;Y|F)) = 0$.

Definition 6. *[Coverage] Given three features X, X^* and Y, X covers X^* on Y if and only if $I(X^*;Y|X) = 0$.*

- Feature level selection

The approach processes features from the same group and seeks the best feature subset from the arrived features so far. When a new feature f_x arrives, test whether it is relevant to the target feature f_y. If f_x is irrelevant to f_y, discard f_x directly. If f_x can provide new information for f_y that any other formerly selected feature cannot, add f_x to the subset. If f_x is redundant but it can cover some other features, then replace these features with f_x. Otherwise, f_x is redundant and will be discarded.

- Group level selection

The approach seeks a set of groups that can cover as much uncertainty of the target feature f_y as possible with a minimum cost. That is to seek a group set Γ which is sparse at both the group level and the individual level, by solving the optimization problem as defined in Eq. (7).

$$min_{\Gamma}\{H(Y) - I(\Gamma;Y)\} + \{\lambda_1|\Gamma|_g + \lambda_2|\Gamma|_f\} \tag{7}$$

where $|\Gamma|_g$ is the number of groups and $|\Gamma|_f$ is the number of selected features in the selected group set Γ. This is the penalty on the number of selected groups and the number of selected features. The group level selection is similar to the feature level selection and the only differences are the selection level and the penalty. If the new information of the newly arrived group provides for Y is more than the penalty that comes with it, it will be selected.

The framework of GFSSF is able to complete feature selection at the individual feature level and the group level simultaneously. After receiving new features form the feature steam, the feature level selection is invoked to process the newly arrived feature in the current group. When all features of a group have arrived, the feature level selection for this group is done. Then the group level selection is invoked to process the new group. Thus, GFSSF can be easily set to do feature selection at the individual level, the group level, or both.

3.2.2 OGFS

OGFS (Online Group Feature Selection) [18] is an efficient online feature selection framework using the prior knowledge of group information. OGFS consists of two stages as the intra-group feature selection and inter-group features selection.

- Intra-group selection

This stage selects features dynamically by a criterion based on the spectral graph theory. Given the label of the data, two graphs G_w and G_b are created. G_w reflects the within-class or local affinity relationship and G_b reflects the between-class or global affinity relationship. The graphs G_w and G_b are characterized by the weight matrices S_w and S_b calculated as shown in Eq.(8) (9).

$$(S_b)_{ij} = \begin{cases} \frac{1}{n} - \frac{1}{n_l} & y_i = y_j = l \\ \frac{1}{n} & y_i \neq y_j \end{cases} \tag{8}$$

$$(S_w)_{ij} = \begin{cases} \frac{1}{n_l} & y_i = y_j = l \\ 0 & y_i \neq y_j \end{cases} \tag{9}$$

where n_l denotes the number of data points from class $l \in \{1, 2, ..., c\}$.

The feature selector matrix is $W = [w_i, ..., w_m]^T \in R^{d\times m}$, where d is the number of selected features and m is the dimension of the global feature space. The data matrix projected on the selected feature space is $Z = W^T X$. The best selection matrix can be achieved by maximizing the following objective function as shown in Eq. (10).

$$F(W_U) = \frac{\Sigma_{ij}||z_i - z_j||^2 S_b(ij)}{\Sigma_{ij}||z_i - z_j||^2 S_w(ij)} \tag{10}$$

When a new feature f_i arrives, the algorithm will calculate the value $|F(U \cup f_i) - F(U)|$. If the value is bigger than the small positive parameter λ, then feature f_i is assumed to be distinguished and selected. The intra-group selection selects all the significant features with distinguished ability. However, this may cause redundancy among selected features, and therefore it turns to the inter-group selection phase.

- Inter-group selection

This stage aims to get an optimal subset based on global group information. Using the linear regression model Lasso (least absolute shrinkage and selection operator), it reformulates the function as shown in Eq. (11).

$$min_{\widehat{\beta}}||y - X^T\widehat{\beta}||_2 + \lambda||\widehat{\beta}||_1 \tag{11}$$

where $||.||_2$ stands for l_2 norm, and $||.||_1$ stands for l_1 norm of a vector. λ is a parameter that controls the amount of regularization applied to estimators, and $\lambda \geqslant 0$. The above function can be solved efficiently by many optimization methods. The features with non-zero coefficients will be selected.

More specifically, on time step t, a group of features g_t is generated. In online intra-group selection, OGFS develops a novel criterion based on spectral analysis which aims to select discriminative features in g_t. Each feature in g_t is evaluated individually in this stage. Then in inter-group selection, OGFS reevaluates all the selected features so far and discards the features which are irrelevant to the class label. The process can be accomplished with a sparse linear regression model Lasso.

3.2.3 Group-SAOLA

Group-SAOLA [57] extends the SAOLA algorithm and can select feature groups which are sparse at the levels of both features and groups. At time t_i, group-SAOLA attempts to get a solution that is sparse at the levels of both intra-groups and inter-groups simultaneously for maximizing its predictive performance for classification. At first, we give the definitions of feature groups. Let $I(X;Y)$ denote the mutual information between X and Y.

Definition 7. *[Irrelevant groups] If* $\exists G_i \subset G$ *s.t.* $I(C;G_i) = 0$*, then* G_i *is considered as an irrelevant feature group.*

Definition 8. *[Group redundancy in inter-groups] If* $\exists G_i \subset G$ *s.t.* $I(C;G_i|G-G_i) = 0$*, then* G_i *is a redundant group.*

Definition 9. *[Feature redundancy in intra-groups]* $\forall F_i \in G_i$*, if* $\exists S \subset G - F_i$ *s.t.* $I(C;F_i|S) = 0$*, then* F_i *can be removed from* G_i*.*

Group-SAOLA consists of three key steps:

- Step 1: At time t_i, if G_i is an irrelevant group, then discard it. If not, go to step 2.
- Step 2: Evaluate feature redundancy in G_i to make it as parsimonious as possible at the intra-group level.
- Step 3: Remove redundant groups from the currently selected groups.

3.3 Analysis

In this subsection, we will compare and analyze the advantages and disadvantages of all online feature selection approaches mentioned above. More details are below.

- Grafting

The advantages of Grafting are as follows [14]:

1) Grafting is an embedded feature selection approach, and it treats the selection of features as an integral part of learning a predictor in a regularized learning framework.

2) Grafting can discard many irrelevant and redundant features and have a single global optimal solution, because it is based on a stagewise gradient descent and regularized risk minimization technique.

The disadvantages of Grafting are as follows [16, 18]:

1) Grafting needs the information of the global feature space to choose a good value for the important regularization parameter in advance, and it is hence weak in the handling of streaming features.

2) When the number of features is very large, it is very time-consuming because of the gradient retesting over all the selected features. This causes the runtime of Grafting to increase dramatically and eventually fail frequently.

- Alpha-investing

As compared to Grafting, Alpha-investing has the following advantages [15, 16]:

1) Alpha-investing is a streamwise feature selection approach, thus, it does not need to determine any prior parameters in advance and can handle large feature sets.

2) Alpha-investing can dynamically adjust the threshold for adding features to the model, and it is hence conducive to reduce over fitting.

3) It is extremely easy to be implemented because Alpha-investing just calculates featuresą́ p-values.

Meanwhile, Alpha-investing also has the disadvantages below [16–18]:

1) Alpha-investing does not consider the redundancy of selected features, because it only evaluates each feature once.

2) Alpha-investing needs a threshold in advance, thus, it cannot properly handle the original features without any prior information about the feature structure.

3) Alpha-investing is not computationally efficient if the size of the streaming feature set is huge and the number of features within the current model is large, because it uses the p-value of features. Moreover, it appears to be highly unstable and may fail to select any features on very sparse datasets.

- OSFS and Fast-OSFS

As compared to Grafting and Alpha-investing, OSFS and Fast-OSFS have the following advantages [16–18]:

1) OSFS and Fast-OSFS can remove redundant features from the selected candidates, thus, they select fewer features than Grafting and Alpha-investing while achieving comparable prediction accuracy. OSFS and Fast-OSFS can make sure that the newly added feature is optimal for global selected features so far.

2) Fast-OSFS divides online redundancy analysis into two parts compared to OSFS, thus, it has stronger statistical power than OSFS and it is much faster than OSFS. Meanwhile, when the feature space is unknown or significantly large, Fast-OSFS has better and more stable performance than Alpha-investing and OSFS.

Meanwhile, OSFS and Fast-OSFS also have some disadvantages below [17, 18]:

1) Considering the online redundancy analysis step of OSFS, the running time of OSFS is linear to the number of total features, but it is exponential to the number of features to be handled.

2) OSFS and Fast-OSFS are online individual feature selection approaches, therefore, they miss the group structure among features in some applications.

- SAOLA

SAOLA addresses two challenges in many big data applications: extremely high dimensionality and its highly scalable requirement of feature selection, and it hence has the advantages as follows [13]:

1) By using the strategy of online pairwise comparisons, SAOLA can handle a large size of feature space and keep scalable on data sets of extremely high dimensionality.

2) SAOLA employs a k-greedy search strategy to filter out redundant features, and it is hence conducive to select fewer features and perform faster.

Meanwhile, the disadvantage of SAOLA is that it is hard to get an optimal value for the relevance threshold.

- OFS and OFSp

OFS and OFSp have advantages as follows [56]:

1) OFS can achieve more and more significance of the gain with the training instances received, thus, it can work efficiently for large-scale data mining tasks.

2) OFS can select the exact number of features specified by users, because online learners of OFS and OFSp allow maintaining a classifier involved only a small and fixed number of features.

3) By using sparsity regularization and truncation techniques, OFS can guarantee that the sparsity level of the learner keeps unchanged during the entire online learning process.

Meanwhile, OFS and OFSp have the following disadvantages.

1) The selected features are not optimal for the global feature space arrived so far, because OFS and OFSp do not remove redundant features from the selected candidates.

2) OFS and OFSp miss the group structure among features in some applications, because they are learned on streaming features individually. .

- GFSSF

As compared to the above algorithms, GFSSF has the following advantages [17]:

1) By using feature group structures information, GFSSF can effectively identify relevant features from important groups and select features at both the group and individual levels.

2) GFSSF can select features by treating each feature as an individual group, thus, it can work without using feature group structures and it can be easily configured to do feature selection in the group level, in the individual feature level, or in both levels.

3) GFSSF removes redundant features and selects fewer features, so it is comparable or superior to the algorithms that do not deal with redundancy.

- OGFS

OGFS has the advantages below [18]:

1) OGFS can perform better than those algorithms without considering the information of the group structure, because it uses the group structure information as a type of prior knowledge on the features.

2) OGFS-Intra is a filter model, thus, it obtains a high efficiency and is linear with the number of features.

3) OGFS reevaluates selected features in the inter-group selection, so it is conducive to select sufficient features with discriminative power.

4) OGFS can get a better feature subset, because it facilitates the relationship of features within groups and the correlation between groups.

Meanwhile, the disadvantage of OGFS is that as the intra-group selection of OGFS needs to choose a small number positive parameters in advance, it is hard to choose an optimal value without any prior information.

- Group-SAOLA

As an extension of SAOLA, group-SAOLA has the advantages as follows [57]:

1) By using the strategy of online pairwise comparisons, group-SAOLA can handle a large size of feature space and is scalable to data sets with extremely high dimensionality.

2) Group-SAOLA can remove redundancy in intra-groups and inter-groups, thus, it can select feature groups which are sparse at the levels of both features and groups.

Meanwhile, the disadvantage of group-SAOLA is similar to SAOLA, that is, it is hard to get an optimal value of the relevance threshold.

4 Experiments

In this section, we first summarize several benchmark data sets in feature selection. Then we introduce evaluation measures and a Library of Online streaming Feature Selection (LOFS) [58]. Finally, we conduct experiments on online feature selection algorithms mentioned above.

4.1 Experimental Data Sets for Online Feature Selection

4.1.1 NIPS 2003 feature selection challenge

The NIPS 2003 challenge in feature selection [1)] is to find feature selection algorithms that significantly outperform methods using all features [16]. It contains five benchmark datasets including: ARCENE, MEDELON, GISETTE, DEXTER and DOROTHEA formatted for that purpose . To facilitate entering results for all five datasets, all tasks are two-class classification problems. A brief introduction of these five datasets is as follows:

- ARCENE: The task of ARCENE is to distinguish cancer versus normal patterns from mass-spectrometric data. This is a two-class classification problem with continuous input variables.
- GISETTE: The task of GISETTE is to discriminate between confusable handwritten digits, the four and the nine. This is a two-class classification problem with sparse continuous input variables.
- DEXTER: The task of DEXTER is to filter texts about "corporate acquisitions". This is a two-class classification problem with sparse continuous input variables.
- DORTHEA: The task of DOROTHEA is to predict which compounds bind to Thrombin. This is a two-class classification problem with sparse binary input variables.
- MADELON: The task of MADELON is to classify random data. This is a two-class classification problem with sparse binary input variables.

Table 1 (AT: attribute type, NI: number of instances, NF: number of features) shows more details of the above five benchmark data sets.

[1)] http://clopinet.com/isabelle/Projects/NIPS2003/

Table 1 Data sets of NIPS 2003 feature selection challenge

Data Set	AT	NI	NF
ARCENE	Non sparse	100	10000
MEDELON	Non sparse	2000	500
GISETTE	Non sparse	6000	5000
DEXTER	Sparse Integer	300	20000
DOROTHEA	Sparse Binary	800	100000

4.1.2 UCI benchmark datasets

The UCI Machine Learning Repository [2)] is a collection of databases, domain theories, and data generators that are used for empirical analysis of machine learning algorithms [16–18, 56]. By September 2015, it has 332 data sets . We introduce below some data sets used in online feature selection algorithms mentioned in Section 3.

- IONOSPHERE: Classification of radar returns from the ionosphere.
- SPECTF Heart: Data on cardiac Single Proton Emission Computed Tomography (SPECT) images. Each patient is classified into two categories: normal and abnormal.
- ARRHYTHMIA: Distinguish between the presence and absence of cardiac arrhythmia and classify it into one of the 16 groups
- NORTHIX: Northix is designed to be a schema matching benchmark problem for data integration of two entity relationship databases.
- ISOLET: Predict which letter-name was spoken, a simple classification task.

Table 2 (AT: attribute type, NI: number of instances, NF: number of features) shows more details of the above UCI benchmark data sets.

Table 2 Data sets of UCI benchmarks

Data Set	AT	NI	NF
IONOSPHERE	Integer Real	351	34
SPECTF	Integer	267	44
ARRHYTHMIA	Categorical Integer Real	452	279
NORTHIX	Integer Real	115	200
ISOLET	Real	7797	617

4.1.3 MLDATA

The MLDATA.org [3)] is a web site built as a repository for machine learning data [16]. This project is supported by PASCAL (Pattern Analysis, Statistical Modeling and

[2)] http://archive.ics.uci.edu/ml/datasets.html
[3)] http://www.mldata.org/repository/data/

Computational Learning). By September 2015, it has 861 data sets . We introduce below some data sets used in online feature selection algorithms mentioned in Section 3.

- DLBCL Tumor from Harvard: There are two kinds of classifications about diffuse large b-cell lymphoma (DLBCL) addressed in the publication.
- LUNG Cancer (Michigan): 86 primary lung adenocarcinomas samples and 10 non-neoplastic lung samples are included. Each sample is described by 7129 genes.
- OVARIAN Cancer (NCI PBSII Data): Ovarian cancer due to family or personal history of cancer.

Table 3 (AT: attribute type, NI: number of instances, NF: number of features) shows the details of these three data sets.

Table 3 Date sets of MLDATA

Data Set	AT	NI	NF
DLBCL	Integer String	77	7130
LUNG	Floating Point Integer String	96	7130
OVARIAN	Floating Point Integer String	253	15155

4.1.4 CIFAR-10 and CIFAR-100

The CIFAR-10 [18, 56] and CIFAR-100 are labeled subsets of the 80 million tiny images dataset [4]. They were collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton.

The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one testing batch, each with 10000 images. The testing batch exactly contains 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches exactly contain 5000 images from each class.

This CIFAR-100 is just like the CIFAR-10, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. The 100 classes in the CIFAR-100 are grouped into 20 super-classes.

4.1.5 CALTECH-101 and CALTECH-256

The CALTECH-101 [18] is a data set of digital images which was collected in September 2003 by Fei-Fei Li, Marco Andreetto, and Marc 'Aurelio Ranzato [5]. It contains 9144 pictures which belong to 101 categories. There are about 40 to 800 images per category and most categories have about 50 images. The size of each image is roughly 300 x 200 pixels. It is intended to facilitate Computer Vision research and techniques, and it is most applicable to techniques involving image recognition classification and categorization.

The CALTECH-256 is the latest dataset in the same website. It contains 30608 pictures which belong 256 categories. There are at least 80 images per category.

4.2 Evaluate Measures

There are three popular evaluation measures used in online feature selection [16–18], including:

- Compactness: the proportion or number of selected features.
- Prediction Accuracy: the percentage of the correctly classified testing instances which are previously unseen.
- Runtime: the time consumption of the algorithms run on a data set, generally in seconds.

4.3 Library Software

LOFS [58] (Library of Online streaming Feature Selection) is the first comprehensive open-source library for use in MATLAB that implements the state-of-the-art algorithms of online streaming feature selection. By using the library, researchers perform comparisons between new and existing methods .

LOFS contains three modules: CM (Correlation Measure), Learning and SC (Statistical Comparison).

- CM module: the library provides four measures to calculate correlations between features, including Chi-square test, G^2 test, the Fisher's Z test and mutual information.
- Learning module: it consists of two sub modules, LFI (Learning Features added Individually) and LGF (Learning Grouped Features added sequentially). The LFI module includes Alpha-investing, OSFS, Fast-OSFS and SAOLA, while the LGF module provides the group-SAOLA algorithm.
- SC module: a series of performance evaluation metrics, such as prediction accuracy, kappa statistic, compactness. In order to conduct and statistical

[4] http:// www.cs.toronto.edu/ kriz/cifar.html

[5] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

comparisons of algorithms over multiple data sets, this module also conducts the Friedman test and Nemenyi test.

4.4 Empirical Results

By using LOFS, we compare online feature selection algorithms on several benchmark datasets. MEDELON, DEXTER and DOROTHEA are from the NIPS 2003 challenge in feature selection (and also can be found in UCI benchmark datasets), LEUKEMIA is from the MLDATA, and NEWS20 and WEBSPAM with extremely high dimensionality are available at the Libsvm data set website [6]. Details of these data sets are shown in Table 4 (NI: number of instances, NF: number of features, NT: number of testing instances).

Table 4 Benchmark data sets

Data Set	NI	NF	NT
MEDELON	500	2,000	600
LEUKEMIA	7,129	48	24
DEXTER	20,000	300	300
DOROTHEA	100,000	800	300
NEWS20	1,355,191	9,996	10,000
WEBSPAM	16,609,143	20,000	78,000

We use two classifiers KNN and J48 provided in the Spider Toolbox [7] to evaluate a selected feature subset in experiments. Meanwhile, we compare these algorithms in section 3 with three evaluate measures, including prediction accuracy, number of selected features and running time. All experiments are conducted on a computer with Inter(R) i5-3470S 2.9GHz, and 8GB memory. More details are as follows.

4.4.1 Comparison of Online Individual Feature Selection Approaches

In this subsection, we select four state-of-the-art algorithms of online individual feature selection, such as Alpha-investing, Fast-OSFS, OFS and SAOLA in our experiments. The value of k for the KNN classifier is set to 1. The parameter for SAOLA is set to 0 for discrete data, and the significance level on SAOLA is set to 0.01 for Fisher's Z-test for continuous data [57]. The significance level for Fast-OSFS is set to 0.01 and parameters for Alpha-investing are set to the optimal values used in [15]. OFS adopts a user-defined parameter k to control the size of the selected feature subset. We set k to the top of 5, 10, 15,..., 100, and then select the feature set with the highest prediction accuracy as the experimental results.

Table 5 Prediction accuracy(J48)

Data Set	Alpha-investing	Fast-OSFS	OFS	SAOLA
MEDELON	0.607	0.610	0.637	0.608
LEUKEMIA	0.667	0.958	0.958	0.958
DEXTER	0.500	0.820	0.567	0.813
DOROTHEA	0.934	0.937	0.937	0.934
NEWS20	-	-	0.733	0.827
WEBSPAM	-	-	0.969	0.961

Table 6 Prediction accuracy(KNN)

Data Set	Alpha-investing	Fast-OSFS	OFS	SAOLA
MEDELON	0.577	0.528	0.643	0.562
LEUKEMIA	0.625	0.792	0.875	0.917
DEXTER	0.500	0.780	0.540	0.760
DOROTHEA	0.740	0.946	0.909	0.920
NEWS20	-	-	0.688	0.776
WEBSPAM	-	-	0.952	0.953

Table 7 Number of selected features

Data Set	Alpha-investing	Fast-OSFS	OFS	SAOLA
MEDELON	4	3	65	3
LEUKEMIA	2	5	45	17
DEXTER	1	9	85	21
DOROTHEA	113	5	60	63
NEWS20	-	-	85	212
WEBSPAM	-	-	85	51

From Tables 5-8, we have the following observations.

- Alpha-investing has the lowest accuracy among all these algorithms. When the size of features is less than 1,000, OFS performs best. While the size of features varies from 1,000 to 1,000,000, Fast-OSFS is superior to other competing algorithms. If the size of features is bigger than 1,000,000 (NEWS20 and WEBSPAM), the notation "-" denotes that Alpha-investing and Fast-OSFS fail on these data sets because of expensive time cost. These results validate that OFS and SAOLA can deal with extremely high dimensionality feature selection problems. When the size of features is bigger than 10,000,000, OFS is superior to SAOLA.
- From Table 7, we can see that Alpha-investing and Fast-OSFS select fewer features than OFS and SAOLA. With the increasing of the size of features, OFS roughly selects the same size of final feature subset. For

[6] http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets
[7] http://people.kyb.tuebingen.mpg.de/spider

Table 8 Running time(seconds)

Data Set	Alpha-investing	Fast-OSFS	OFS	SAOLA
MEDELON	0.1	0.1	0.1	0.1
LEUKEMIA	1	2	0.1	2
DEXTER	6	4	1	3
DOROTHEA	461	379	14	62
NEWS20	-	-	1,637	1009
WEBSPAM	-	-	19,443	1556

SAOLA, the number of selected features increases stably as the size of candidate features increases.

- From Table 8, we can find that all these algorithms can efficiently complete the selection if the size of candidate features is less than 10,000. If the size of features is bigger than 1,000,000, the running time of SAOLA increases stably, and for OFS, it increases dramatically.

In sum, Alpha-investing and Fast-OSFS select fewer features and Fast-OSFS can achieve competitive accuracy if the dimensionality of candidate features is not extremely high. OFS and SAOLA can address the challenge of extremely high dimensionality and SAOLA is the most stable algorithm among all aforementioned online feature selection approaches.

4.4.2 Comparison of Online Group Feature Selection

In this subsection, we select two state-of-the-art online group feature selection algorithms including group-SAOLA and OGFS in our experiments. The value of k for the KNN classifier is set to 1. For group-SAOLA, the parameter is the same as the SAOLA algorithm. The parameters for OGFS are set to the values used in [18]. For data sets MEDELON, LEUKEMIA, DEXTER and DOROTHEA, we randomly divide each data set into 100 feature groups without overlapping. For NEWS20 and WEBSPAM, each data set is randomly divided into 10,000 feature groups without overlapping. The results are as shown in Tables 9-10.

Table 9 Prediction accuracy

Data Set	J48		KNN	
	group-SAOLA	OGFS	group-SAOLA	OGFS
MEDELON	0.611	0.515	0.532	0.492
LEUKEMIA	0.958	0.729	0.983	0.779
DEXTER	0.843	0.556	0.795	0.549
DOROTHEA	0.936	0.903	0.918	0.869
NEWS20	0.819	0.530	0.750	0.506
WEBSPAM	0.934	0.762	0.938	0.938

Table 10 Running time and Number of selected features/groups

Data Set	running time (second)		selected features/groups	
	group-SAOLA	OGFS	group-SAOLA	OGFS
MEDELON	0.1	0.1	2/2	15/13
LEUKEMIA	3	1	16/14	66/47
DEXTER	2	4	21/19	72/49
DOROTHEA	25	24	41/27	126/67
NEWS20	2,341	1,150	140/140	192/192
WEBSPAM	3,275	22,790	17/17	401/395

From Tables 9 and 10, we can conclude as follows.

- From Table 9, group-SAOLA is better than OGFS in prediction accuracy.
- From Table 10, group-SAOLA is more stable than OGFS in running time, especially when the dimensionality is extremely high. Meanwhile, group-SAOLA selects fewer features and groups but performs better. These results validate that group-SAOLA is more efficient.

5 Challenging Issues for Online Feature Selection

In spite of the rapid development and wide application of feature selection in many fields, there are still several open issues in online feature selection with streaming features as follows.

1) Existing online feature selection algorithms mainly focus on the handling of single label classification. However, in many scenarios, the instances may have two or more labels. Thus, how to handle online feature selection for multi-label data is a challenging issue.

2) In real-world online applications, the quality of data cannot be guaranteed, such as lack of attribute values, noisy data and so on. Thus, how to select high quality features from a feature stream is another challenging issue.

3) With the rapid growth of the amount of data, centralized online feature selection algorithms will become increasingly unable to meet the requirements of computational performance, thus, distributed online feature selection algorithms will become another challenge in the future.

6 Conclusion

In this paper, we reviewed the area of online feature selection with streaming features. After a brief introduction of the traditional feature selection approaches, we focus on the latest development of online feature selection with streaming features. We summarize and analyze several representative online individual feature selection and online group feature selection algorithms. Then we summarize several benchmark data sets and evaluation measures popularly used in online feature selection algorithms, a library of online streaming feature selection (LOFS) and give the experimental results with some of these online feature selection algorithms. Finally, we address some new challenges of online feature selection in the future work.

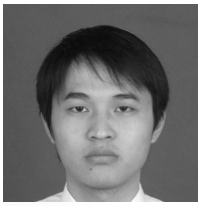
Acknowledgements This work is supported in part by the National Key Research and Development Program of China under grant 2016YFB1000901, the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education, China, under grant IRT13059, the National 973 Program of China under grant 2013CB329604, the Specialized Research Fund for the Doctoral Program of Higher Education under grant 20130111110011, the Natural Science Foundation of China under grants (61273292, 61229301, 61503112, 61673152).

References

1. Wu X, Zhu X, Wu G, Ding W. Data mining with big data. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(1): 97–107
2. Franck M. How many photos are uploaded to flickr every day and month? http:// www.flickr.com/photos/franckmichel/6855169886
3. Pollack J R, Perou C M, Alizadeh A A, Eisen M B, Pergamenschikov A, Williams C F, Jeffrey S S, Botstein D, Brown P O. Genome-wide analysis of dna copy-number changes using cdna microarrays. Nat Genet, 1999, 23(1): 41–46
4. Wang D, Irani D, Pu C. Evolutionary study of web spam: Webb spam corpus 2011 versus webb spam corpus 2006. In: Proceedings of the sixteenth annual ACM symposium on parallelism in algorithms and architectures, CollaborateCom-2012. 2012, 40–49
5. Farahat A K, Elgohary A, Ghodsi A, Kamel M S. Greedy column subset selection for large-scale data sets. Knowledge and Information Systems, 2015, 45(1): 1–34
6. Patra B K, Nandi S. Effective data summarization for hierarchical clustering in large datasets. Knowledge and Information Systems, 2015, 42(1): 1–20
7. Hoi S, Wang J, Zhao P, Jin R. Online feature selection for mining big data. In: BigMine '12: Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications. 2012
8. Guyon I, Elisseeff A. An introduction to variable and feature selection. The Journal of Machine Learning Research, 2003, 3: 1157–1182
9. Peng H, Long F, Ding C. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans on Pattern Analysis and Machine Intelligence, 2005, 27(8): 1226–1238
10. Wang M, Li H, Tao D, Lu K, Wu X. Multimodal graph-based reranking for web image search. IEEE Transactions on Image Processing, 2012, 21(11): 4649–4661
11. Ding W, Stepinski T F, Mu Y, Bandeira L, Ricardo R, Wu Y, Lu Z, Cao T, Wu X. Sub-kilometer crater discovery with boosting and transfer learning. ACM Transactions on Intelligent Systems and Technology, 2011, 2(4): 1–22
12. Wu X, Yu K, Wang H, Ding W. Online streaming feature selection. In: Proceedings of the 27th International Conference on Machine Learning. 2010, 1159–1166
13. Yu K, Wu X, Ding W, Pei J. Towards scalable and accurate online feature selection for big data. In: 2014 IEEE International Conference on Data Mining (ICDM). 2014, 660–669
14. Perkins S, Theiler J. Online feature selection using grafting. In: Proceedings of the 20th International Conference on Machine Learning. 2003, 592–599
15. Zhou J, Foster D P, Stine R A, Ungar L H. Streamwise feature selection. Journal of Machine Learning Research, 2006, 3(2): 1532–4435
16. Wu X, Yu K, Ding W, Wang H, Zhu X. Online feature selection with streaming features. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(5): 1178–1192
17. Li H, Wu X, Li Z, Ding W. Group feature selection with streaming features. In: IEEE 13th International Conference on Data Mining. 2013, 1109–1114
18. Wang J, Wang M, Li P, Liu L, Zhao Z, Hu X, Wu X. Online feature selection with group structure analysis. IEEE Transactions on Knowledge and Data Engineering, 2015, 27: 3029–3041
19. Zhang K, Zhang L, Yang M H. Real-time object tracking via online discriminative feature selection. IEEE Transactions on Image Processing, 2013, 22(12): 4664–4677
20. Collins R T, Liu Y, Leordeanu M. Online selection of discriminative tracking features. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(10): 1631–1643
21. Carvalho V R, Cohen W W. Single-pass online learning: Performance, voting schemes and online feature selection. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006
22. Jiang W, Er G, Dai Q, Gu J. Similarity-based online feature selection in content-based image retrieval. IEEE Transactions on image processing, 2006, 15(3): 702–712
23. Stefanowski J, Cuzzocrea A, Slezak D. Processing and mining complex data streams. Information Sciences, 2014, 285: 63–65
24. Xiao J, Xiao Y, Huang A, Liu D, Wang S. Feature-selection-based dynamic transfer ensemble model for customer churn prediction. Knowl-

edge and Information Systems, 2015, 43(1): 29–51
25. Zhou T C, Lyu M R T, King I, Lou J. Learning to suggest questions in social media. Knowledge and Information Systems, 2015, 43(2): 389–416
26. Liu H, Yu L. Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(4): 491–502
27. Song L, Smola A, Gretton A, Borgwardt K M, Bedo J. Supervised feature selection via dependence estimation. In: 24th International Conference on Machine Learning (ICML-2007). 2007
28. Mitra P, Murthy C, Pal S K. Unsupervised feature selection using feature similarity. IEEE Trans on Pattern Analysis and Machine Intelligence, 2002, 24(3): 301–312
29. Yu G, Zhang G, Zhang Z, Yu Z, Deng L. Semi-supervised classification based on subspace sparse representation. Knowledge and Information Systems, 2015, 43(1): 81–101
30. Zhao Z, Liu H. Semi-supervised feature selection via spectral analysis. In: Proceedings of SIAM International Conference on Data Mining. 2007, 641–647
31. Liu H, Motoda H. Computational Methods of Feature Selection. Chapman and Hall/CRC Press, 2007
32. Yu L, Liu H. Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the 20th International Conferences on Machine Learning. 2003, 601–608
33. He X, Cai D, Niyogi P. Laplacian score for feature selection. Advances in Neural Information Processing Systems, 2005, 17: 507–514
34. Gu Q, Li Z, Han J. Generalized fisher score for feature selection. Statistics, 2012
35. Zhang D, Chen S, Zhou Z. Constraint score: A new filter method for feature selection with pairwise constraints. Pattern Recognition, 2008, 41(5): 1440–1451
36. Sun D, Zhang D. Bagging constraint score for feature selection with pairwise constraints. Pattern Recognition, 2010, 43(6): 2106–2118
37. Liu M, Zhang D. Sparsity score: A novel graph preserving feature selection method. International Journal of Pattern Recognition and Artificial Intelligence, 2014, 28(4): 1450009
38. Liu M, Miao L, Zhang D. Two-stage cost-sensitive learning for software defect prediction. IEEE Transactions on Reliability, 2014, 63(2): 676–686
39. Liu M, Zhang D. Pairwise constraint-guided sparse learning for feature selection. IEEE Transactions on Cybernetics, 2015
40. Wei H, Billings S A. Feature subset selection and ranking for data dimensionality reduction. IEEE Trans on Pattern Analysis and Machine Intelligence, 2007, 29(1): 162–166
41. Yu L, Liu H. Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research, 2004, 5(1): 1205–1224
42. Kwak N, Choi C H. Input feature selection by mutual information based on parzen window. IEEE Trans on Pattern Analysis and Machine Intelligence, 2002, 24(12): 1667–1671
43. Kira K, Rendell L A. The feature selection problem: Traditional methods and a new algorithm. In: Proc of the 9th National Conf on Artificial Intelligence. 1992, 129–134
44. Robnik-Sikonja M, Kononenko I. Theoretical and empirical analysis of relieff and rrelieff. Machine Learning, 2003, 53(1-2): 23–69
45. Almuallim H, Dietterich T G. Learning with many irrelevant features. In: Proc of 9th National Conf on Artificial Intelligence. 1992, 547–552
46. Liu H, Setiono R. A probabilistic approach to feature selection–a filter solution. In: Proc of Int Conf on Machine Learning. 1996, 319–327
47. Kohavi R, Johnb G H. Wrappers for feature subset selection. Artificial Intelligence, 2013, 97(1): 273–324
48. Liu H. Feature selection for knowledge discovery and data mining. Boston: Kluwer Academic Publishers, 1998
49. Tang J, Alelyani S, Liu H. Feature selection for classification: A review. Data Classification: Algorithms and Applications, 2014, 37
50. Tibshirani R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B(Methodological), 1996, 267–288
51. Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. The Annals of Statistics, 2004, 32(2): 407–451
52. Zou H, Hastie T. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2005, 67(2): 301–320
53. Zou H. The adaptive lasso and its oracle properties. Journal of the American Statistical Association, 2006, 101: 1418–1429
54. Friedman J, Hastie T, Tibshirani R. A note on the group lasso and a sparse group lasso. Mathmatical Statistics, arXiv: 1001.0736vl, 2010
55. Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistics Society B, 2006, 68(1): 49–67
56. Wang J, Zhao P, Hoi S C, Jing R. Online feature selection and its applications. IEEE Transactions on Knowledge and Data Engineering, 2013, 26(3): 698–710
57. Yu K, Wu X, Ding W, Pei J. Scalable and accurate online feature selection for big data. arXiv: 1511.092632v2 [cs.LG], 2016
58. Yu K, Ding W, Wu X. Lofs: Library of online streaming feature selection. Knowledge Based Systems, 2016

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



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

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

Jing Wang is a researcher in the Department of Statistics, Rutgers University. She received the BE, ME and PhD degrees from the School of Computer Science and Information Engineering, Hefei University of Technology (HFUT), China, in 2009, 2011 and 2015 respectively. She is a Visiting Research Student in the Learning and Vision Research Group of National University of Singapore (NUS). Her research interests include data mining, computer vision,and machine learning.

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