Knowledge-Based Systems 113 (2016) 1–3

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Original software publication

# LOFS: A library of online streaming feature selection

Kui Yu[a,*], Wei Ding[b], Xindong Wu[c]

[a] University of South Australia, Adelaide, Australia
[b] University of Massachusetts Boston, Boston, USA
[c] University of Louisiana, Lafayette, USA

ARTICLE INFO

*Article history:*
Received 21 June 2016
Revised 26 August 2016
Accepted 28 August 2016
Available online 1 September 2016

*Keywords:*
Streaming feature selection
Online group feature selection

ABSTRACT

As an emerging research direction, online streaming feature selection deals with sequentially added dimensions in a feature space while the number of data instances is fixed. Online streaming feature selection provides a new, complementary algorithmic methodology to enrich online feature selection, especially targets to high dimensionality in big data analytics. This paper introduces the first comprehensive open-source library, called LOFS, for use in MATLAB and OCTAVE that implements the state-of-the-art algorithms of online streaming feature selection. The library is designed to facilitate the development of new algorithms in this research direction and make comparisons between the new methods and existing ones available. LOFS is available from https://github.com/kuiy/LOFS.



## 1. Introduction

Feature selection is to select a parsimonious feature subset to improve model interpretability and efficiency without degrading model accuracy [1]. Traditionally, online feature selection deals with the observations sequentially added while the total dimensionality is fixed [2]. However, in many real world applications, it is either impossible to acquire an entire feature set or impractical to wait for the complete set before feature selection starts. For instance, in Twitter, trending topics keep changing over time, and thus the dimensionality of data is changed dynamically. When a new top topic appears, it may come with a set of new keywords, which usually serve as key features to identify new hot topics. In bioinformatics, it is expensive for feature selection to acquire an entire set of features for each data observation due to the high cost in conducting wet lab experiments [2]. Many big data applications call for online streaming feature selection to consume sequentially added dimensions over time.

As an emerging research direction, online streaming feature selection provides a new, complementary algorithmic methodology to enrich online feature selection, especially addresses high dimensionality in big data analytics. But to the best of our knowledge, there is no comprehensive open-source packages existing for this problem. To facilitate research efforts on this research direction, we develop the open-source library called LOFS (**L**ibrary of **O**nline streaming **F**eature **S**election).

The main contribution of the LOFS library lies on three aspects. (1) It is the first comprehensive open-source library for implementing algorithms of online streaming feature selection. (2) It provides the state-of-the-art algorithms of online streaming feature selection mainly developed by our research group. (3) It is written in MATLAB and OCTAVE respectively, easy to use, and completely open source. We hope it will facilitate the development of new online algorithms for tackling the grand challenges of high dimensionality in big data analytics, and encourage researchers to extend LOFS and share their algorithms through the LOFS framework.

## 2. Problems and background

In general, assuming $S$ is the feature set containing all features available till time $t_{i-1}$, and $C$ is the class attribute, then a training data set $D$ is defined by $D = \{S, C\}$, which is a sequence of features that is presented over time. As we process one dimension at a time, the research problem is that at any time $t_i$, how to online maintain a minimum size of feature subset $S^{\star}_{t_i}$ of maximizing its predictive performance for classification. If $F_i$ is a new coming feature at time $t_i$, $S^{\star}_{t_{i-1}} \subset S$ is the selected feature set till time $t_{i-1}$ and $P(C|\zeta)$ denotes the posterior probability of $C$ conditioned on a subset $\zeta$, the problem of online streaming feature selection is formulated as Eq. (1).

* Corresponding author.
*E-mail addresses:* Kui.Yu@unisa.edu.au, ykui713@gmail.com (K. Yu), ding@cs.umb.edu (W. Ding), xwu@louisiana.edu (X. Wu).

http://dx.doi.org/10.1016/j.knosys.2016.08.026
0950-7051/

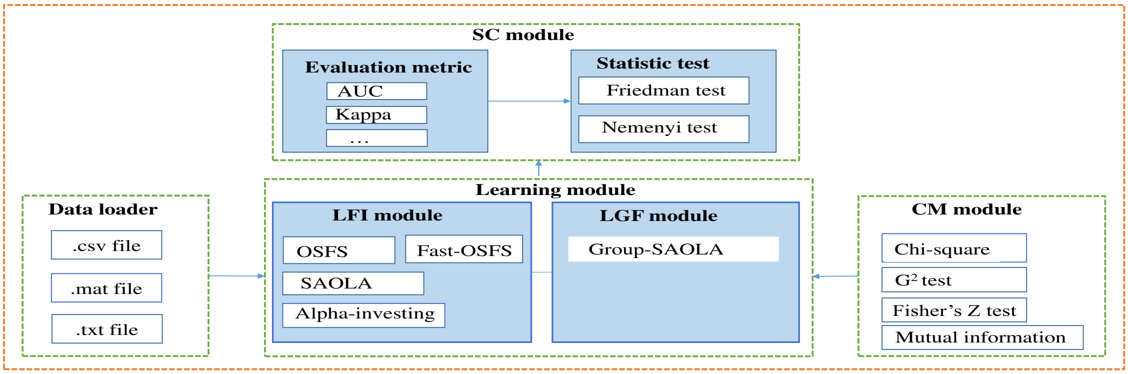


**Fig. 1.** Architecture and functionalities of LOFS.

```
%load data
load('spect.mat');
[n,p]=size(spect);
%set the index of the class attribute
class_index=p;
%set training and testing data
traindata=spect(1:500,:);
testdata=spect(501:end,:);
%set the significant level to 0.01
alpha=0.01;
%using the G² test
test='g2';
%learning module, using example of Fast-OSFS for discrete data
[selectedFeatures,time]=fast_osfs_d(traindata,class_index,alpha,test);
%evaluation module,using KNN classifier (k=3)
%test_class denotes the predicted class labels
test_class =
knnclassify(testdata(:,selectedFeatures),traindata(:,selectedFeatures),traindata(:,class_index),3);
%calculate AUC
[X,Y,T,AUC] = perfcurve(testdata(:,class_index),test_class,1);
%calculate prediction accuracy
accuracy=length(find(testdata(:,class_index) == test_class))/length(test_class);
%calculate kappa statistic
kappa = cal_kappa(class_label(test_indices),test_class,'class');
```

**Fig. 2.** An illustrative example of implementing LOFS.

$$S^{\star}_{t_i} = \arg\min_{S'}\{|S'| : S' = \arg\max_{\zeta \subseteq \{S^{\star}_{t_{i-1}} \cup F_i\}} P(C|\zeta)\}. \quad (1)$$

To solve Eq. (1), currently the state-of-the-art algorithms include Grafting [3], Alpha-investing [4] , OSFS [5], Fast-OSFS [6], and SAOLA [7]. All of those algorithms only deal with one dimension at a time upon its arrival.

Group information sometimes is embedded in a feature space. For instance, in image analysis, features are generated in groups which represent color, texture and other visual information. If $G_{t_{i-1}}$ is the set of all feature groups available till time $t_{i-1}$, then at time $t_{i-1}$, $D$ is denoted by $D = \{G_{t_{i-1}}, C\}$, which is a sequence of feature groups that is added sequentially. To consume grouped features sequentially added over time, online selection of dynamic groups is formulated as Eq. (2).

$$\begin{aligned} & G^{\star}_{t_i} = \arg\max_{G_\zeta \subseteq \{G^{\star}_{t_{i-1}} \cup G_{t_i}\}} P(C|G_\zeta) \\ & s.t. \\ & (a) \forall F_k \in G_j, G_j \subset G^{\star}_{t_i}, P(C|\{G_j - \{F_k\}, F_k\}) \neq P(C|\{G_j - \{F_k\}\}) \\ & (b) \forall G_j \subset G^{\star}_{t_i}, P(C|\{G^{\star}_{t_i} - G_j, G_j\}) \neq P(C|\{G^{\star}_{t_i} - G_j\}). \end{aligned} \quad (2)$$

In Eq. (2), $G_{t_i}$ is a new coming group at time $t_i$, and $G^{\star}_{t_{i-1}} \subseteq G_{t_{i-1}}$ is the set of selected groups till time $t_{i-1}$. Eq. (2) attempts to yield a set of groups at time $t_i$ that is as parsimonious as possible at the levels of both intra-groups (constraint (a)) and inter-groups (constraint (b)) simultaneously for maximizing its predictive performance for classification. To online utilize grouped features, the group-SAOLA algorithm was proposed [8].

## 3. Framework of LOFS

The LOFS architecture is based on a separation of three modules, that is, CM (**C**orrelation **M**easure), Learning, and SC (**S**tatistical **C**omparison), as shown in Fig. 1. The three modules in the LOFS architecture are designed independently, and all codes follow the MATALB standards. The learning module consists of two submodules, LFI (**L**earning **F**eatures added **I**ndividually) and LGF (**L**earning **G**rouped **F**eatures added sequentially).

In the CM module, LOFS provides four measures to calculate correlations between features, Chi-square test, $G^2$ test, Fisher's Z test, and mutual information, where Chi-square test, $G^2$ test, and mutual information for dealing with discrete data while Fisher's Z test for handling continuous data.

With the measures above, the LFI module includes Alpha-investing [4], OSFS [5], Fast-OSFS [6], and SAOLA [7] to learn features added individually over time, while the LGF module provides the group-SAOLA algorithm [8] to online mine grouped features added sequentially.

Based on the learning module, the SC module provides a series of performance evaluation metrics (i.e., prediction accuracy, AUC, kappa statistic, and compactness, etc.). To conduct statistical comparisons of algorithms, the SC model further provides the Friedman test and the Nemenyi test [9].

## 4. Implementation of LOFS and empirical results

The LOFS library comes with detailed documentation. The documentation is available from https://github.com/kuiy/LOFS. This documentation describes the setup and usage of LOFS. All functions and related data structures are explained in detail. Fig. 2 gives an example to show how to implement the functions of the Fast-OSFS algorithm in LOFS, such as loading data, setting parameters, running algorithms, and evaluating performance.

In addition to the documentation, the extensive performance comparisons between online streaming feature selection and traditional online feature selection, and the empirical comparisons between online streaming feature selection and traditional feature selection can be found in [8].

## 5. Conclusion

This paper presents LOSF, an open-source package for online streaming feature selection to facilitate research efforts in machine learning and data mining. Through the LOFS framework, we hope that it will facilitate researchers to develop new online learning algorithms for big data analytics and share their algorithms.

## Appendix. Required metadata

### Current executable software version (Tables 1 and 2)

**Table 1**
Software metadata (optional).

| Nr. | (executable) Software metadata description | Please fill in this column |
|---|---|---|
| S1 | Current software version | V1.0. |
| S2 | Permanent link to executables of this version | https://github.com/kuiy/LOFS/releases/tag/v1.0 |
| S3 | Legal Software License | the GNU General Public License version 3 |
| S4 | Computing platform/Operating System | Microsoft Windows 7/10, and Linux |
| S5 | Installation requirements & dependencies | MATLAB12a, OCTAVE4.0.2, C++ compiler for Windows/Linux, and MIToolbox (https://github.com/Craigacp/FEAST) |
| S6 | If available, link to user manual - if formally published include a reference to the publication in the reference list | https://github.com/kuiy/LOFS/tree/master/LOFS_Matlab/manual https://github.com/kuiy/LOFS/tree/master/LOFS_Octave/manual |
| S7 | Support email for questions | ykui713@gmail.com |

### Current code version

**Table 2**
Code metadata (mandatory).

| Nr. | Code metadata description | Please fill in this column |
|---|---|---|
| C1 | Current code version | V1.0 |
| C2 | Permanent link to code/repository used of this code version | https://github.com/kuiy/LOFS/releases/tag/v1.0 |
| C3 | Legal Code License | the GNU General Public License version 3 |
| C4 | Code versioning system used | Git |
| C5 | Software languages, tools, and services used | MATLAB12a and OCTAVE4.0.2 |
| C6 | Compilation requirements, operating environments & dependencies | C++ compiler for Windows/Linux, and the library of MIToolbox (https://github.com/Craigacp/FEAST) |
| C7 | If available Link to developer documentation/manual | https://github.com/kuiy/LOFS/tree/master/LOFS_Matlab/manual https://github.com/kuiy/LOFS/tree/master/LOFS_Octave/manual |
| C8 | Support email for questions | ykui713@gmail.com |

## References


[1] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[2] J. Wang, P. Zhao, S.C. Hoi, R. Jin, Online feature selection and its applications, knowledge and data engineering, IEEE Trans. 26 (3) (2014) 698–710.
[3] S. Perkins, J. Theiler, Online feature selection using grafting, in: ICML-2003, 2003, pp. 592–599.
[4] J. Zhou, D.P. Foster, R.A. Stine, L.H. Ungar, Streamwise feature selection, J. Mach. Learn. Res. 7 (2006) 1861–1885.
[5] X. Wu, K. Yu, H. Wang, W. Ding, Online streaming feature selection, in: ICML-2010, 2010, pp. 1159–1166.
[6] X. Wu, K. Yu, W. Ding, H. Wang, X. Zhu, Online feature selection with streaming features, Pattern Anal. Mach. Intell. IEEE Trans. 35 (5) (2013) 1178–1192.
[7] K. Yu, X. Wu, W. Ding, J. Pei, Towards scalable and accurate online feature selection for big data, in: IEEE ICDM-2014, 2014, pp. 660–669.
[8] K. Yu, X. Wu, W. Ding, J. Pei, Scalable and accurate online feature selection for big data, ACM Transactions on Knowledge Discovery from Data, 2016. In press
[9] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.