Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Applications Note

Subject Section

matFR: a matlab toolbox for feature ranking

Zhicheng Zhang [1,2], Xiaokun Liang [1], Shaode Yu [3,4,5,*], and Yaoqin Xie [1,*]

[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, GD 518055, China;
[2] Department of Radiation Oncology, Stanford University, Stanford, CA 94305, United States;
[3] College of Information and Communication Engineering, Communication University of China, Beijing 100024, China;
[4] Key Laboratory of Convergent Media and Intelligent Technology (Communication University of China), Ministry of Education, Beijing 100024, China;
[5] Department of Radiation Oncology, University of Texas Southwestern Medical Center, Dallas, TX 75390, United States.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Summary: Nowadays, it becomes much easier to collect high-throughput features (variables, or attributes) for quantitative representation and precision medicine. Automatic ranking to figure out the most informative and discriminative features becomes increasingly important. This paper presents a matlab toolbox (matFR) which has already integrated 42 feature ranking (FR) methods. These methods apply mutual information, statistical analysis, structure clustering and other principles to estimate the relative importance of features in specific measure spaces. Specifically, this paper reports the toolbox organization and method summary, and an example is used to show how to apply a FR method to sort mammographic breast lesion features. The matFR toolbox is easy to use and flexible to integrate additional methods. Most importantly, it provides a tool to compare, investigate and interpret the features selected for various applications.

Availability and implementation: The toolbox is available at http://github.com/NicoYuCN/matFR. To each FR method, its demo and reference publications are provided. If interested in integrating your algorithms, please feel free to contact us.

Contact: yushaodemia@163.com (SY); yq.xie@siat.ac.cn (YX)

1 Introduction

It is feasible to collect tens of thousands of features in medical imaging and biology for quantitative representation and personalized medicine (Wang *et al.*, 2010; Gillies *et al.*, 2015) and consequently, to figure out those informative and discriminative features becomes challenging but urgently important. Several open resource toolboxes have released a few methods for feature selection, such as $sklearn$ in python (Pedregosa *et al.*, 2011), the $caret$ package in R (Kuhn *et al.*, 2008), and the $weka$ in Java (Witten *et al.*, 2002). However, few lately developed methods have been integrated in these toolboxes. In addition, some programming skills are required to apply these methods which hampers their wide application. Our motivation is to provide an easy-to-use toolbox with a variety of methods. It is worthy of note that feature selection library (FSLib) is well-organized in MATLAB (Roffo *et al.*, 2016), while there is still room for further improvement.

Feature selection is well-established and feature selection methods can generate feature rank of each feature or a subset of selected features (Cai *et al.*, 2018). This paper focuses on the feature ranking (FR) algorithms. In general, FR can be viewed as a previous step of feature selection (Saeys *et al.*, 2007; Rohart *et al.*, 2017) and a FR method aims to sort features according to their relative importance in a specific measure space.

A large number of FR methods have been developed, and little is known of their efficiency and effectiveness. This study presents a toolbox (matFR) which is implemented with MATLAB, a widely used language in many scientific areas. The contribution of this study comes from three points. First, FR methods are concerned. Forty-two methods have been embedded and a dozen of them are proposed in recent five years. Second, the tool is user-friendly. After data loaded, users can use one line of code to activate one FR methods for specific applications. And thus, systematic comparison of these methods from various perspectives becomes feasible and easy-to-implement. More interestingly, the toolbox can be continuously updated in an online-sharing fashion, in particular when algorithm developers begin to contribute to this project.

2 The matFR toolbox

2.1 Method summary

The matFR has integrated 42 methods among which 12 methods are from FSLib (Roffo *et al.*, 2016), 9 from mutual information (MI) based feature selection repository (Nguyen *et al.*, 2014), 7 are embedded in MATLAB ("rankfeatures", "relieff" and "lasso"), and others are accessible online.

FR methods can be grouped into various categories. Based on the utilized training data, these methods can be classified into supervised and unsupervised methods. The former requires data labels and feature relevance is determined by the correlation between a feature vector and its labels. On contrary, unsupervised methods require no labels and they exploit data variance and separability to measure feature relevance. The matFR toolbox contains 29 supervised and 13 unsupervised methods.

FR methods can also be categorized from theoretical perspective. In the toolbox, MI and its variants are most used to quantify the feature importance (12 methods), including max relevance, min redundancy and max dependency (Peng *et al.*, 2005; Nguyen *et al.*, 2014). The second most used is statistical analysis and it tends to assess the importance of every feature in binary classification (8 methods). Besides, a hot topic in unsupervised methods is structure learning (8 methods) and it uses graph representation to capture data structure followed by different intermediate analysis and search strategies to figure out these discriminative features (Du *et al.*, 2015). Others might use local structure preservation, eigenvector centrality, manifold learning, and other criterion.

2.2 An example

Given an input matrix X (m instances and n features per instance) and its corresponding class labels Y ($Y \in \{0, 1\}$), the procedure of using one (f) of the FR methods (F) in the toolbox can be described as $r = F(X, f, Y)$, where r stands for the output rank indexes of features in a descending order with regard to the relative importance of features. If an unsupervised method is selected, the class labels Y can be omitted.

As shown in Figure 1, a user can activate any FR methods through an interface function 'matFR_interface.m' which determines the belonging of a method f. If f is a MI based FR method, the function 'matFR_mi.m' is triggered, else the other function 'matFR_fn.m' is activated.

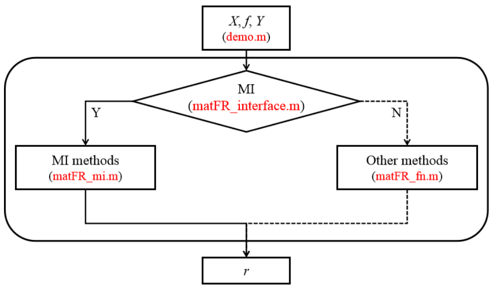

Fig. 1. The semantic diagram of how to use a FR method (f) in the toolbox (F) for a specific application. A user needs is to prepare the data (X, Y) and to define a FR method (f), and the toolbox outputs the ranked indexes of features (r). Note that FR methods are enumerated in one of the interface functions ('matFR_mi.m' and 'matFR_fn.m').

An example is demonstrated. Given the BCDR-F03 data set, it contains 406 lesions and 736 mammographic images (Arevalo *et al.*, 2016). To each annotated image, 17 features are selected to quantify the lesions from mass intensity, contour shape and lesion texture. Among the mass lesions, 230 are histologically verified benign ($y = 0$) and 176 are malignant ($y = 1$). Thus, to the input X, $m = 736$ and $n = 17$. After the data is prepared, one line code to activate the Laplacian Score algorithm (He *et al.*, 2006), an unsupervised method, is shown as below,

```
r = matFR_interface( X, 'h2_fir_laplacian_score');
```

and one line code to activate a joint MI based FR algorithm (Nguyen *et al.*, 2014), a supervised method, is shown as below,

```
r = matFR_interface( X, 'b9_mi_joint', Y).
```

Note that the short names of all FR methods are listed in the 'demo.m' file. As for further details, please refer to the publications and algorithm implementations.

2.3 Implementation

The toolbox is mainly implemented with MATLAB, while the MI based methods require a C++ compiler to compile two *cpp* files. One file aims for computing the pairwise MI matrix between feature-feature and feature-class, and the other is for the joint MI matrix. The matFR toolbox has been tested on 64-bit Windows 7/8/10 systems, MATLAB R2018a/R2019a and Microsoft Visual C++ 2012/2015/2017.

3 Future work

The future work arises from three aspects. First, to integrate available FR methods into the toolbox and to follow up newly developed methods. The most promising way is contributions from algorithm developers to this project through online collaboration. Second, to complete the details of FR methods. For instance, some advanced discretization algorithms, *e.g.*, class-attribute interdependence redundancy (CAIR) (Ching *et al.*, 1995) and class-attribute interdependence maximization (CAIM) criterion (Kurgan *et al.*, 2004), could be adopted to MI based methods for diverse options. Last but not the least, to accelerate the distribution of these FR algorithms, the toolbox could be implemented in Python and R.

Funding

This work was supported by grants of Shenzhen matching project (GJHS20170314155751703); National Key Research and Develop Program of China (2016YFC0105102); National Natural Science Foundation of China (61871374); Leading Talent of Special Support Project in Guangdong (2016TX03R139); and Fundamental Research Program of Shenzhen (JCYJ20170413162458312).

References

Arevalo, J., Gonzalez, F.A., Ramos-Pollan, R., Oliveira, J.L., Lopez, M.A.G. (2016) Representation learning for mammography mass lesion classification with convolutional neural networks, *Computer methods and programs in biomedicine*, **127**, 248-257.

Cai, J., Luo, J., Wang, S. and Yang, S. (2018) Feature selection in machine learning: A new perspective, *Neurocomputing*, **300**, 70-79.

Du, L., Shen, Y.D. (2015) Unsupervised feature selection with adaptive structure learning, *ACM SIGKDD international conference on knowledge discovery and data mining*, 209-218.

Gillies, R.J., Kinahan, P.E., Hricak, H. (2015) Radiomics: images are more than pictures, they are data, *Radiology*, **278(2)**, 563-577.

He, X., Cai, D., Niyogi, P. (2006) Laplacian score for feature selection, *Advances in neural information processing systems*, 507-514.

Kuhn, M. (2008) Building predictive models in R using the caret package, *Journal of statistical software*, **28(5)**, 1-26.

Nguyen, X.V., Chan, J., Romano, S. and Bailey, J. (2014) Effective global approaches for mutual information based feature selection, *20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 512-521.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J. (2011) Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research*, **12**, 2825-2830.

Peng, H., Long, F., Ding, C. (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **8**, 1226-1238.

Rohart, F., Gautier, B., Singh, A., Le Cao, K.A. (2017) mixOmics: An R package for 'omics feature selection and multiple data integration, *PLoS computational biology*, **13(11)**, e1005752.

Roffo, G. (2016) Feature selection library (MATLAB toolbox), *arXiv preprint*, **arXiv**, 1607.01327.

Saeys, Y., Inza, I., Larranaga, P. (2007) A review of feature selection techniques in bioinformatics, *Bioinformatics*, **23(19)**, 2507-2517.

Witten, I.H., Frank, E. (2002) Data mining: practical machine learning tools and techniques with Java implementations, *Acm Sigmod Record*, **31(1)**, 76-77.

Guo, J., Zhu, W. (2018) Dependence guided unsupervised feature selection, *Thirty-Second AAAI Conference on Artificial Intelligence*, 507-514.

Wang, D., Bodovitz, S. (2010) Single cell analysis: the new frontier in 'omics', *Trends in biotechnology*, **28(6)**, 281-290.

Ching, J.Y., Wong, A.K.C., Chan, K.C.C. (1995) Class-dependent discretization for inductive learning from continuous and mixed-mode data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17(7)**, 641-651.

Kurgan, L.A., Cios, K.J. (2004) CAIM discretization algorithm, *IEEE transactions on Knowledge and Data Engineering*, **16(2)**, 145-153.