



Paper Type: Research Paper



PyIT-MLFS: A Python-Based Information Theoretical Multi-Label Feature Selection Library

Sadegh Eskandari*

Department of Computer Science, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran; eskandari@guilan.ac.ir.

Citation:

Eskandari, S. (2022). PyIT-MLFS: a Python-based information theoretical multi-label feature selection library. *International journal of research in industrial engineering*, 11 (1), 9-15.

Received: 04/10/2021

Reviewed: 02/11/2021

Revised: 01/01/2022

Accepted: 30/01/2022

Abstract

Multi-label learning is an emerging research direction that deals with data in which an instance may belong to multiple class labels simultaneously. As many multi-label data contain very large feature space with hundreds of irrelevant and redundant features, multi-label feature selection is a fundamental pre-processing tool for selecting a subset of most representative and discriminative features. This paper introduces a Python-based open-source library that provides the state-of-the-art information theoretical filter-based multi-label feature selection algorithms. The library, called PyIT-MLFS, is designed to facilitate the development of new algorithms. It is the first comprehensive open-source library for implementing algorithms of multilabel feature selection. Moreover, it provides a high-level interface that enables the end-users to test and compare different already implemented algorithms. PyIT-MLFS is available from <https://github.com/Sadegh28/PyIT-MLFS>.

Keywords: Feature selection, Multi-label learning library, Data mining.

1 | Introduction

Recent years, multi-label learning has received increased attention in applications such as text sentiment analysis [13], music emotion recognition [23], movie content analysis [16] and image analysis [7], [6], [18] and [1]. Let $X^F = \{x_1, x_2, \dots, x_m\}$ denote a set of n -dimensional instances constructed from a set of features F , such that $|F| = n$. Then, in a multi-label data each instance $x_i \in X^F$ is assigned to a label subset $y_i \subseteq y$, where $y = \{y_1, y_2, \dots, y_{|L|}\}$ is a finite set of labels.

Many practical multi-label data contain a very large feature space [17], [14] and [5]. Generally, most of the features in a large feature space are irrelevant or redundant. Such features are directly translated into a large number of learning parameters and hence decrease generalization performance of classifiers. Multi-label feature selection tries to address this issue by selecting a small subset of most representative and discriminative features.



International Journal of Research in Industrial Engineering. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).



Corresponding Author: eskandari@guilan.ac.ir

<http://dx.doi.org/10.22105/riej.2022.308916.1252>

Many multi-label feature selection algorithms are introduced in the literature [19]. Among them, information theoretical filter-based methods have generated much interest, because they are simple, fast and not biased to any special learner. Moreover, it is shown that information theoretical-based criteria, such as mutual information, are able to measure any kind of dependency between random variables [2].

Given a multi-label data (X^F, y) with original feature set F and label set y , the goal of information theoretical filter-based multi-label feature selection is to identify the smallest feature subset $S \subset F$ such that $I(X^S, y) = I(X^F, y)$. Finding such a subset is computationally infeasible because all possible subsets of F should be evaluated. Therefore, the existing multi-label feature selection methods try to generate informative subsets using criteria called evaluation functions. An evaluation function takes a feature $f \in F$ as input and returns a goodness value $J(f)$. Different methods can be specified based on the evaluation function they use. Generally, these algorithms start with an empty selected subset and gradually add the features that have the best evaluation values.

As an emerging research direction, information theoretical filter-based multi-label feature selection is a well-studied problem, with several algorithms and evaluation functions introduced in the literature [3], [11], [9], [8], [7], [25], [10] and [24]. However, to the best of our knowledge, there is no comprehensive open-source packages existing for this problem. Therefore, the open-source library called Python-Based Information Theoretical Multi-Label Feature Selection (PyIT-MLFS) is developed. The author hopes that it will facilitate the efforts on this novel research direction.

The unique contributions of PyIT-MLFS are: 1) it is the first comprehensive opensource library for implementing algorithms of information theoretical filter-based multilabel feature selection, 2) It provides the state-of-the-art algorithms in the field, including one problem transformation method that converts multi-label data into single-label data and eight algorithm adaptation methods that try to select informative features from multi-label data directly, 3) It is written in Python, a dominant language in machine learning and data mining tasks, so that it can be used as a pre-processing tool alongside other well-known Python-based libraries, such as scikit-learn [15] and scikit-multilearn [20].

2 | Information Theory

Information theory provides effective tools to measure different types of correlations (linear and non-linear) between features. Thereupon this theory is widely used in feature selection [22]. In this section, the theory is defined, focusing on entropy and mutual information concepts. In this section, the theory is defined, focusing on entropy and mutual information concepts.

For a discrete random variable X , the entropy is defined as.

$$H(X) = - \sum_{x \in X} p(x) \log p(x). \tag{1}$$

$H(X)$ quantifies the uncertainty present in the distribution of X . If $H(X) = 1$, then there is no certainty about the outcome of X . In contrast, if $H(X) = 0$ then the outcome of X can be predicted perfectly. The conditional entropy of X given Y , is defined as:

$$H(X|Y) = - \sum_{y \in Y} \sum_{x \in X} p(x, y) \log p(x|y). \tag{2}$$

$H(X|Y)$ quantifies the amount of uncertainty remaining in X given the random variable Y is known. The Mutual information is defined as the reduction in the entropy of one random variable due to the knowledge of the other. Mutual information of two random variables X and Y , is defined as:

$$I(X; Y) = H(X) - H(X|Y). \tag{3}$$

Table 1 lists the algorithms implemented by PyIT-MLFS. For each algorithm, this table reports its type (PT = Problem Transformation and AAD = Algorithm Adaptation), evaluation function and computational complexity. A problem transformation method converts the multi-label data into a single-label one and consequently use traditional single-label feature selection algorithms to select the best subset. On the other hand, an algorithm adaptation method selects informative features from multi-label data directly. Its worth noting that all the implemented algorithms are filter-based methods. Each candidate feature is evaluated using a model-independent criteria that describes general characteristics of the training data. Therefore, the subsets selected by PyIT-MLFS algorithms are not biased to any special classifier.

Table 1. List of the multi-label feature selection approaches provided with the PyIT-MLFS.

No	Acronym	Type	Evaluation Function	Computational Complexity
1	PPT-MI [3]	PT	$J(f) = I(X^{F-S}, PPT(y)) - I(X^{F-S-\{f\}}, PPT(y))$	$O(F ^2)$
2	IGMF [11]	AAD	$J(f) = \frac{2 \times (H(f) + H(y) - H(f, y))}{H(f) + H(y)}$	$O(F L)$
3	D2F [9]	AAD	$J(f) = \sum_{f_i \in Y} I(f, y_i) - \sum_{f_i \in S} \sum_{y_j \in Y} I(f, f_i, y_j)$	$O(L F ^2 + L F ^3)$
4	PMU [8]	AAD	$J(f) = \sum_{y_i \in Y} I(f, y_i) - \sum_{f_i \in S} \sum_{y_j \in Y} I(f, f_i, y_j) - \sum_{y_i \in Y} \sum_{y_j \in Y} I(f, y_i, y_j)$	$O(L F ^2 + L F ^3 + L ^2 F ^2)$
5	MDMR [12]	AAD	$J(f) = \sum_{y_i \in Y} I(f, y_i) - \sum_{f_i \in S} \sum_{y_j \in Y} I(f, f_i, y_j) - \sum_{y_i \in Y} \sum_{y_j \in Y} I(f, y_i, y_j)$	$O(L F ^2 + L F ^3 + L ^2 F ^2)$
6	SCLS [10]	AAD	$J(f) = \sum_{f_i \in Y} I(f, y_i) - \sum_{f_j \in S} \frac{I(f, f_j)}{H(f)} \sum_{y_i \in Y} I(f, y_i)$	$O(F ^3 + L F ^2)$
7	LRFS [24]	AAD	$J(f) = \sum_{y_i \in Y} \sum_{y_j \in Y, y_i \neq y_j} I(f, y_i y_j) - \frac{1}{ S } \sum_{f_j \in S} I(f, f_j)$	$O(F ^3 + L ^2 F ^2)$
8	LSMFS [25]	AAD	$J(f) = \sum_{y_i \in Y} \left\{ I(f, y_i) + \sum_{y_j \in Y, y_i \neq y_j} \max\{0, I(f, y_i, y_j)\} \right\} - \sum_{f_j \in S} I(f, f_j)$	$O(F ^3 + L F ^2 + L ^2 F ^2)$
9	MLSMFS [25]	AAD	$J(f) = \sum_{y_i \in Y} \left\{ I(f, y_i) + \max_{y_j \in Y, y_i \neq y_j} \{0, I(f, y_i y_j)\} \right\} - \sum_{f_j \in S} I(f, f_j)$	$O(F ^3 + L F ^2 + L ^2 F ^2)$

Table 1 reports their type, evaluation function and computational complexity. As for the evaluation function, $PPT(\cdot)$ is the label transformation function, S is the current selected subset, $H(\cdot)$ and $H(\cdot, \cdot)$ represent the entropy and joint entropy, respectively, and $I(\cdot, \cdot)$, $I(\cdot, \cdot, \cdot)$ and $I(\cdot, \cdot | \cdot)$ represent mutual information, multi-variate mutual information, and conditional mutual information, respectively.

4 | Architecture of PyIT-MLFS

The major structure of PyIT-MLFS (Fig. 1) is written on top of three Python libraries, NumPy [4], Pyitlib¹, and scikit-multilearn [20]. NumPy is the fundamental package for scientific computing in Python that provides efficient algorithms on multidimensional arrays, such as sorting, selecting, reshaping, and I/O. Pyitlib is a library of information-theoretic methods for data analysis. Scikit-multilearn is a scikit-learn compatible [15] multi-label classification library. It provides implementations of the most popular algorithms on multi-label data, including the well-known MLKNN classifier as well as popular approaches from scikit-learn as base classifiers.

PyIT-MLFS consists of four main parts: 1) the IM (Information Measures) module, 2) the data module, 3) the MLFS (Multi-Label Feature Selection) package, and 4) the high-level PyIT-MLFS module. The IM module provides different informationtheoretical measures for calculating relevance and redundancy values. The data module provides different data-related functionalities, abstracted in read data method. This method is able to read train/test data from local csv files. It is also able to download data from Mulan Library [21]. The MLFS package contains the implementations of different multi-label feature selection algorithms listed in Table 1. Each algorithm in this package provides two operations, rank and select: in the former, the importance of each individual feature in feature space is evaluated; in the latter, the final subset of features to be selected is provided. The PyIT-MLFS module is a high-level enduser interface for running different algorithms on different datasets and evaluating their speed and accuracy.

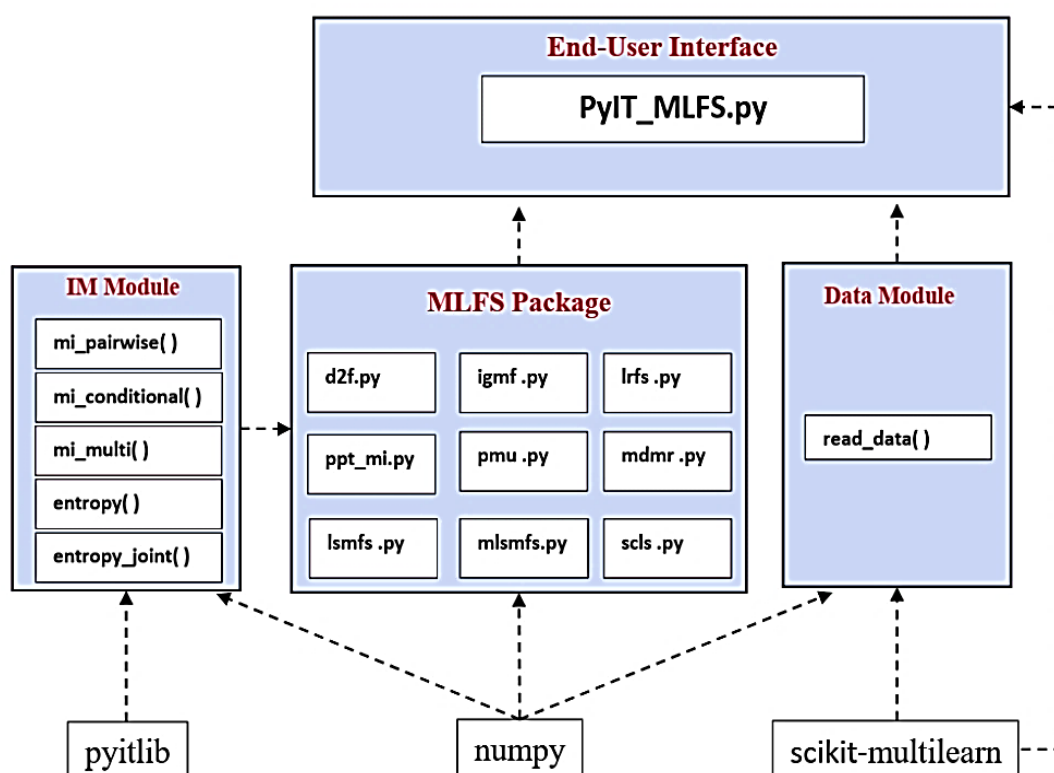


Fig. 1. Architecture of PyIT-MLFS.

¹ <https://pafoster.github.io/pyitlib/>

Although PyIT-MLFS can be used to develop new algorithms, the most appealing characteristic of PyIT-MLFS is that it enables the end-users to test and compare different already implemented information theoretical filter-based multi-label feature selection algorithms. Such a comparison can be made through the following general command:

```
python PyIT-MLFS.py --datasets d1, d2, ..., dn
                    --fs-methods a1, a2, ..., am
                    --classifiers c1, c2, ..., ck
                    --metrics m1, m2, ..., mt
```

where each d_i is a dataset, each a_i is an algorithm from those listed in *Table 1*, each c_i is a multi-label classifier, and each m_i is a metric for evaluating the quality of the classifiers on feature subsets of different sizes. For example the following command ranks the features of 'emotions' dataset (from Mulan library)

```
python PyIT-MLFS.py --datasets 'emotions', 'medical'
                    --fs-methods 'LRFS', 'PPT_MI'
                    --classifiers 'MLKNN'
                    --metrics 'hamming loss', 'label ranking loss'
```

using 'LRFS' and 'PPT MI' methods, then classifies the datasets using 'MLKNN' classifier and finally evaluates the classification results using two metrics namely 'hamming loss' and 'label ranking loss'. The results are depicted in *Fig. 2* and *Fig. 3*.

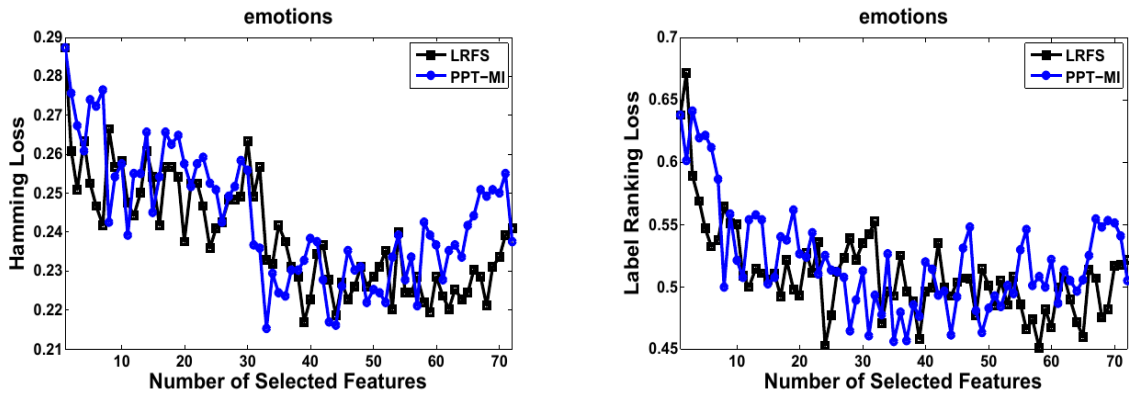


Fig. 2. The results on emotions dataset (left) hamming loss, (right) label ranking loss. The running times to rank all the features using LRFS and PPT-MI are 14 and 4 seconds, respectively.

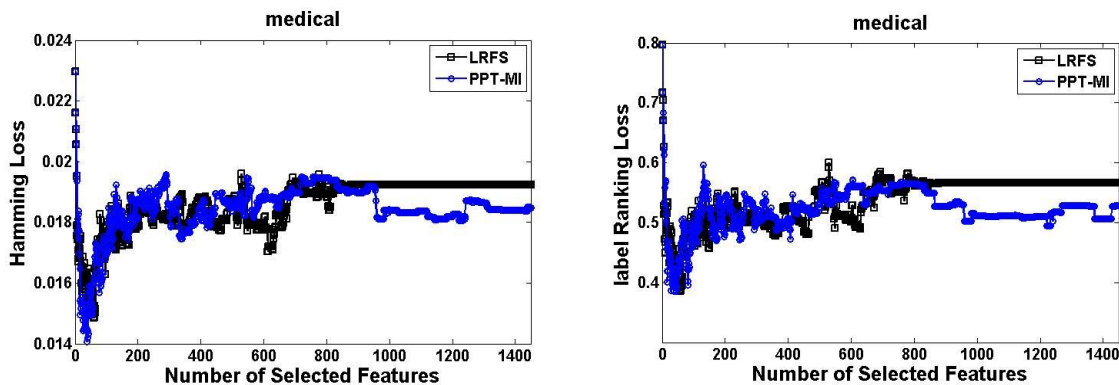


Fig. 3. The results on medical dataset (left) hamming loss, (right) label ranking loss. The running times to rank all the features using LRFS and PPT-MI are 10778 and 92 seconds, respectively.

This paper presents PyIT-MLFS, a Python-based information theoretical filter-based multi-label feature selection library that provides nine state-of-the-art algorithms from the literature as well as easy-to-use tools to develop new algorithms. Future versions of PyIT-MLFS will integrate new developed methods. Moreover, it will add the ability to read data from various formats (in addition to csv files).

References

- [1] Deshpande, H., Singh, A., & Herunde, H. (2020). Comparative analysis on YOLO object detection with OpenCV. *International journal of research in industrial engineering*, 9(1), 46-64.
- [2] Dionisio, A., Menezes, R., & Mendes, D. A. (2004). Mutual information: a measure of dependency for nonlinear time series. *Physica A: statistical mechanics and its applications*, 344(1-2), 326-329.
- [3] Doquire, G., & Verleysen, M. (2011, June). Feature selection for multi-label classification problems. *International work-conference on artificial neural networks* (pp. 9-16). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21501-8_2
- [4] Harris, Ch. R., Millman, K. J., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585, 357–362. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2).
- [5] Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-Pour, H. (2020). MFS-MCDM: Multi-label feature selection using multi-criteria decision making. *Knowledge-based systems*, 206, 106365. <https://doi.org/10.1016/j.knosys.2020.106365>
- [6] Herunde, H., Singh, A., Deshpande, H., & Shetty, P. (2020). Detection of pedestrian and different types of vehicles using image processing. *International journal of research in industrial engineering*, 9(2), 99-113.
- [7] Hong, R., Wang, M., Gao, Y., Tao, D., Li, X., & Wu, X. (2013). Image annotation by multiple-instance learning with discriminative feature mapping and selection. *IEEE transactions on cybernetics*, 44(5), 669-680.
- [8] Lee, J., & Kim, D. W. (2013). Feature selection for multi-label classification using multivariate mutual information. *Pattern recognition letters*, 34(3), 349-357.
- [9] Lee, J., & Kim, D. W. (2015). Mutual information-based multi-label feature selection using interaction information. *Expert systems with applications*, 42(4), 2013-2025.
- [10] Lee, J., & Kim, D. W. (2017). SCLS: multi-label feature selection based on scalable criterion for large label set. *Pattern recognition*, 66, 342-352.
- [11] Li, L., Liu, H., Ma, Z., Mo, Y., Duan, Z., Zhou, J., & Zhao, J. (2014, December). Multi-label feature selection via information gain. *International conference on advanced data mining and applications* (pp. 345-355). Springer, Cham.
- [12] Lin, Y., Hu, Q., Liu, J., & Duan, J. (2015). Multi-label feature selection based on max-dependency and min-redundancy. *Neurocomputing*, 168, 92-103. <https://doi.org/10.1016/j.neucom.2015.06.010>
- [13] Liu, S. M., & Chen, J. H. (2015). A multi-label classification based approach for sentiment classification. *Expert systems with applications*, 42(3), 1083-1093.
- [14] Lv, S., Shi, S., Wang, H., & Li, F. (2021). Semi-supervised multi-label feature selection with adaptive structure learning and manifold learning. *Knowledge-based systems*, 214, 106757. <https://doi.org/10.1016/j.knosys.2021.106757>
- [15] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: machine learning in Python. *The journal of machine learning research*, 12, 2825-2830.
- [16] Pradeep, N., Rao Mangalore, K. K., Rajpal, B., Prasad, N., & Shastri, R. (2020). Content based movie recommendation system. *International journal of research in industrial engineering*, 9(4), 337-348.
- [17] Pereira, R. B., Plastino, A., Zadrozny, B., & Merschmann, L. H. (2018). Categorizing feature selection methods for multi-label classification. *Artificial intelligence review*, 49(1), 57-78.
- [18] Singh, A., Herunde, H., & Furtado, F. (2020). Modified Haar-cascade model for face detection issues. *International journal of research in industrial engineering*, 9(2), 143-171.
- [19] Spolaôr, N., Cherhan, E. A., Monard, M. C., & Lee, H. D. (2013). A comparison of multi-label feature selection methods using the problem transformation approach. *Electronic notes in theoretical computer science*, 292, 135-151.

- [20] Szymanski, P., & Kajdanowicz, T. (2019). Scikit-multilearn: a scikit-based Python environment for performing multi-label classification. *The journal of machine learning research*, 20(1), 209-230.
- [21] Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., & Vlahavas, I. (2011). Mulan: a java library for multi-label learning. *The journal of machine learning research*, 12, 2411-2414.
- [22] Vergara, J. R., & Estévez, P. A. (2014). A review of feature selection methods based on mutual information. *Neural computing and applications*, 24(1), 175-186.
- [23] Yang, Y. H., & Chen, H. H. (2012). Machine recognition of music emotion: a review. *ACM transactions on intelligent systems and technology (TIST)*, 3(3), 1-30.
- [24] Zhang, P., Liu, G., & Gao, W. (2019). Distinguishing two types of labels for multi-label feature selection. *Pattern recognition*, 95, 72-82.
- [25] Zhang, P., Liu, G., Gao, W., & Song, J. (2021). Multi-label feature selection considering label supplementation. *Pattern recognition*, 120, 108137. <https://doi.org/10.1016/j.patcog.2021.108137>