*Journal of **AI** and **D**ata **M**ining*
Vol 7, No 3, 2019, 355-365

DOI: 10.22044/JADM.2018.5780.1688

MLIFT: Enhancing Multi-label Classifier with Ensemble Feature Selection

Sh. Kashef* and H. Nezamabadi-pour

Intelligent Data Processing Laboratory (IDPL), Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran.

Received 02 June 2017; Revised 04 November 2017; Accepted 09 June 2018

*Corresponding author: *shkashef.1988@yahoo.com (Sh. Kashef).*

Abstract

Multi-label classification has gained significant attention during recent years, due to the increasing number of modern applications associated with multi-label data. Despite its short life, different approaches have been presented to solve the task of multi-label classification. LIFT is a multi-label classifier which utilizes a new strategy to multi-label learning by leveraging label-specific features. Label-specific features means that each class label is supposed to have its own characteristics and is determined by some specific features that are the most discriminative features for that label. LIFT employs clustering methods to discover the properties of data. More precisely, LIFT divides the training instances into positive and negative clusters for each label which respectively consist of the training examples with and without that label. It then selects representative centroids in the positive and negative instances of each label by k-means clustering and replaces the original features of a sample by the distances to these representatives. Constructing new features, the dimensionality of the new space reduces significantly. However, to construct these new features, the original features are needed. Therefore, the complexity of the process of multi-label classification does not diminish, in practice. In this paper, we make a modification on LIFT to reduce the computational burden of the classifier and improve or at least preserve the performance of it, as well. The experimental results show that the proposed algorithm has obtained these goals, simultaneously.

Keywords: *Multi-label Data, LIFT Classification, Ensemble Feature Selection.*

1. Introduction

In traditional supervised learning problems, each instance in the dataset belongs to only one label Y_i from a set of labels *L*. However, in some real-world problems, each instance may belongs to a set of labels, $Y_i \subset L$, simultaneously. For example, an image might be annotated as ‘Sunset’ and ‘Beach’. Such prediction tasks are usually denoted as multi-label classification problems, and are used in an increasing number of modern applications such as, semantic image [1] and video [2] annotation, classification of protein functions [3] and genes [4], categorization of text [5] and emotions evoked by music [6], etc.

Multi-label classification methods are mainly grouped into two categories: problem transformation and algorithm adaptation [7]. The first category of methods map the problem of multi-label classification into one or more single-label classification problems where any state-of-the-art single-label learning algorithm can be employed. Representative algorithms of this category are Label Powerset and Binary Relevance [8]. The second category of methods extend some popular learning algorithms to handle multi-label data, directly. Multi-label Naïve Bayes [9], multi-label lazy learning algorithm, ML-kNN [10] and adapting decision tree techniques [11], are some popular approaches of this group.

One of the challenges of multi-label classification is the high number of features of multi-label datasets. This is especially true for images and texts, due to their rich semantics. The high dimensionality of data represents challenges such as poor performance, over-fitting and computational burden to classification analysis [12]. Many of these features are redundant and/or irrelevant and do not play an important role in improving the discriminative ability of the classifier between classes [13]. On the other hand, sometimes, finding the values of a specific feature is so costly that it would be preferred to eliminate it in return for accepting more classification error. Therefore, the main objective of feature selection (FS) is to simplify a dataset by reducing its dimensionality and identifying relevant underlying features without degrading predictive accuracy [14]. Indeed, it is usually observed that the performance of the classifier also increases after feature selection.

The FS approaches can generally be divided into three groups: filter, wrapper, and embedded approaches. The filter approaches operate independently of any learning algorithm. These methods rank the features by some criteria and remove those features that do not achieve a sufficient score. Filter methods have a relatively high speed and are suitable for high-dimensional datasets. These methods are mainly divided into two groups: univariate and multivariate. Methods of the first group, evaluate the quality of features, individually and do not consider possible association with other features. Information gain (IG) [15] and F-score [16] are categorized in this group. Multivariate approaches consider the dependencies between features, but they are computationally more expensive. Mutual information (MI), Relief [17] and fast correlation-based filter [18] are some examples of this category [19].

On the other hand, wrapper methods select those features with high prediction performance estimated by a determined learning algorithm. The accuracy of wrapper methods is larger than filter methods, but the degree of its computational complexity is higher [13]. In the embedded model, feature selection is integrated into the process of training for a given learning algorithm. Therefore, embedded approaches can employ extra information of the cost function. These methods are much faster than wrapper; however, the performance also depends on the classifier.

Similar to single-label feature selection, multi-label approaches are divided into three groups with the same definitions: filter methods [12, 20-22], wrapper methods [9, 23] and embedded ones [24, 25]. In most multi-label feature selection techniques, the multi-label dataset is first transformed to a single-label dataset, using a problem transformation technique such as Label Powerset (LP) or Binary Relevance (BR). Then, any single-label feature selection algorithm can be used to select salient features. Spolaor et al. [15] used LP and BR to transform the problem, and then employed IG and ReliefF for feature selection. Finally, the performance of these four multi-label feature selection methods were compared. Chen et al. [26] present a transformation strategy called Entropy-based Label Assignment (ELA). The authors firstly transform the multi-label data into single-label data, and then apply three single-label filter methods including IG, CHI and OCFS [27]. In Ref. [28] authors employed a pruned problem transformation (PPT) method introduced in [29] to transform multi-label dataset into single-label one. Then, a greedy feature selection procedure based on multidimensional mutual information is executed. A similar method is proposed in [22] which converts the multi-label problem to a single-label problem using PPT, and then utilizes the ReliefF algorithm for giving a weight to each feature.

On the other hand, some multi-label feature selection algorithms which directly work with multi-label datasets are presented in some papers. Lee et al. [20] presented a multi-label feature selection method based on multivariate mutual information, which selects a prominent feature subset by maximizing the multivariate mutual information between the selected features and the labels. Different multi-label feature selection approaches are proposed in [22, 30, 31] by extending the single-label feature selection ReliefF algorithm. An extension of the well-known FCBF method which is a filter method in single-label feature selection is proposed in [32]. This method uses a graphical scheme to indicate the relationship between features and labels. Lee et al. [25] proposed a memetic multi-label feature selection approach, which utilizes memetic procedures to redefine the feature subsets found through a genetic search. Ref. [33] employs evolutionary algorithms for feature selection. The authors firstly employ a filter method to eliminate irrelevant features. Then, an evolutionary algorithm like GSA is used to select the most salient features among the remained features. An incremental multi-label feature selection method based on max-dependency and min-redundancy criterion inspired by the well-known single-label filter method, mRMR, is proposed in [12]. Lin et al. [34] proposed a multi-label feature selection approach that selects salient features based on multi-label neighborhood mutual information. At first, all instances are granulated under different labels using the margin of instance, and three different neighborhood mutual information for multi-label learning are defined. Then, they introduced an optimization objective function to measure the quality of candidate features. A comprehensive review on multi-label feature selection methods can be found in [35].

In this paper, we make a modification on LIFT algorithm which is a multi-label classification method, proposed by Zhang [36]. In this method, some features are constructed for each training instance based on its labels. In the performance phase, first, these features are constructed for the unseen sample. Then, the original features of that sample are ignored and the constructed features substitute them. Finally, classification task is done using these new features that are usually much fewer than the original ones. The problem of LIFT appears in the performance phase. Although, the number of constructed features is less than the original ones, the original features are necessary to obtain the new features. As a result, the mentioned problems associated with redundant, irrelevant and costly features still remain. In this paper, we propose to perform feature selection before employing LIFT algorithm, to make sure that only useful features are fed to LIFT. The results show that in addition to significant reduction of features, the performance of the algorithm remains relatively stable.

The rest of this paper is organized as follows: Section 2 presents LIFT algorithm and a brief introduction to the used filter feature selection approaches. Our method for solving the problem of LIFT algorithm is presented in Section 3. Section 4 reports the results of comparative studies. Finally, our conclusions are given in Section 5.

2. Fundamental concepts

Suppose D be a dataset with N instances $E_i = ((x_i, Y_i),\ i = 1,..., N)$. Each instance is associated with a feature vector $x_i = (x_{i1}, x_{i2},..., x_{iM})$ characterized by M features $X_j,\ j = 1,..., M$, and a label set $L = \{y_1, y_2,..., y_q\}$ denotes the label space with q possible class labels. The task of multi-label learning is to predict the label subset of an unseen instance $E = (x, ?)$, with an acceptable accuracy. Figure 1 shows this representation.

In the following, three filter-based feature selection methods to be used in this paper and the LIFT algorithm are explained.

2.1. Relief

Relief [17] algorithm is a random search technique based on filter methods. It is a classical approach for feature estimation in single-label data, and is designed for binary class problems without missing values [22]. For m random samples from the training set, Relief acts as follows.

	D				L			
	X_1	X_1	⋯	X_1	y_1	y_2	⋯	y_q
E_1	x_{11}	x_{12}	⋯	x_{1M}	0	1	1	0
E_2	x_{21}	x_{22}	⋯	x_{2M}	1	1	0	1
	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋮
E_N	x_{N1}	x_{N1}	⋯	x_{NM}	1	0	1	0

Figure 1.Multi-label data.

First, it searches for the ‘nearest hit’ and ‘nearest miss’ of the selected sample i, which are respectively, the closest same-class instance and the closest different-class instance based on Euclidean distance. Then, it updates feature weights which were initialized by zero for all features. In weighting procedure, the quality of features are estimated according to how well a feature distinguishes two samples from the same classes and from different classes. A higher weight of a feature means that it has a better ability to identify the instances of a class from other classes [19].

2.2. Fast correlation-based filter (FCBF)

FCBF is a multivariate filter approach presented in [18], which is especially designed for high-dimensional data. It considers feature-class correlations as well as feature-feature correlations to find a subset of features which are highly correlated to the class but not highly correlated to the other features. It introduces a measure called Symmetrical Uncertainty (SU) as the ratio between the information gain (IG) and the entropy of two variables. First, it calculates the SU value for each feature and selects those features associated with *SU* values higher than a user-defined threshold. Then, redundant features are removed from this subset, and a subset of relevant informative features remain.

2.3. Information gain (IG)

Information gain is a univariate filter method based on the concept of entropy in information theory. It measures the dependency between each feature of dataset D and the class label, as defined by (1). It ranks features base on their amount of information, such higher values of IG for feature X_i indicates stronger relationship between that feature and the class label [37].

$$IG(D, X_i) = entropy(D) - \sum_{v \in X_i} \frac{|D_v|}{|D|} entropy(D_v) \quad (1)$$

here, feature X_i, $i = 1...M$, can take distinct values, and each subset $D_v \subseteq D$ consists of the set of examples where X_i has the value v.

2.4. LIFT algorithm

The LIFT algorithm works based on two main steps, i.e. construction of label-specific features and induction of classification models. First, LIFT aims to construct discriminative features which take the specific characteristics of each label to simplify its discrimination process. To this end, LIFT utilizes clustering techniques to get insights into the properties of data. It forms a set of positive training examples as well as the set of negative training examples with respect to each class label. In other words, for label l_i , $i=1,...,q$ the *i*th positive and negative sets consist of the training examples with and without label l_i respectively. The well-known k-means algorithm is used to partition these sets into disjoint clusters. Here, both of positive and negative sets of each label are partitioned to the same number of clusters. The cluster centers which specify the main structure of the training examples in regard to l_i, can be used as prototypes for the construction of label-specific features. For each training example, LIFT calculates the Euclidean distance between that sample and the cluster centers of the positive and negative sets. For *q* labels of the multi-label dataset, this process repeats *q* times. Thus, at the end of this stage there are *q* training sets with *2m* features where *m* is the number of cluster centers of positive (equal to negative) sets. At this step, *q* binary classifiers are trained with the produced label-specific features, one for each label. In the testing phase, new features are first constructed for each unseen instance based on its distances to the cluster centers of positive and negative sets, for each label. The labels of this instance are then predicted by the learned system.

3. Proposed method

As it was discussed before, the drawback of LIFT algorithm is the need to have all features for constructing the new features. As a result, the mentioned problems associated with redundant, irrelevant and costly features still remain. A simple idea can make the LIFT algorithm to be computationally efficient. The idea is to remove irrelevant and redundant features and feed salient features to LIFT. Undoubtedly, this idea will reduce the computational burden, and even if the performance deteriorates, slightly, removing costly features is preferred. However, the experimental results show the superiority of our method in most cases, compared to the original LIFT.

Figure 2 shows the diagram of the proposed method. At first, the multi-label dataset is transformed to single-label dataset using the Binary Relevance (BR) approach. In the next step, an ensemble on *n* filter feature selection approaches suggest the best features, individually. The most salient features are then selected among these features in the aggregation phase. Finally, the selected features are fed to the LIFT algorithm.

3.1. BR Transformation

Binary Relevance approach is the most common transformation strategy which transforms the multi-label learning problem into *q* binary classification problems, where *q* is the number of possible labels. More precisely, for the $_j$th label y_j, BR first constructs a corresponding binary training set by considering the relevance of each training instance to y_j. Then, a binary learning algorithm is employed to induce a binary classifier. Each binary classifier is responsible for predicting the association of instances to the corresponding label. For predicting the label set of a test instance, every binary classifier is asked to predict whether or not the test instance belongs to the corresponding label, and then the relevant labels are combined [38].

3.2. Ensemble

When BR transforms the multi-label dataset into *q* binary single-label datasets, the ensemble on *n* filter feature selection methods is performed on these datasets. Three filter methods including FCBF, IG and ReliefF which are three well-known single-label filter feature selectors are selected for this phase. Applying FCBF on each dataset, a binary vector of size *M* is created, where 1 implies selecting and 0 implies deselecting the corresponding feature. At the end of this process, there are *q* binary vectors, and the final selected features are obtained by the OR operator, i.e. it returns the feature *x* as the output, if it is selected in at least one of the vectors.

Similar process is done for Relief and IG methods. However, ReliefF and IG return a weight for each feature and do not specify selected and deselected features. To be fair, the number of features to be selected by these methods are considered equal to the number of selected features, *c*, obtained by FCBF. Therefore, after summing the weights of *q* vectors for each feature, the first *c* features with the highest weights are selected. To determine the final feature set, the ensemble of Relief, IG and FCBF is utilized. 'Ensemble feature selection' is a new strategy which combines the outputs of several feature selectors to achieve better results. Generally, this strategy consists of two steps: in the first step, a number of feature selectors are considered, and the outputs of these base feature selectors are combined in step 2. Bolón et al. propose two models for ensemble feature selections which are shown in figures 3 and 4 [39]. In this paper, the second model is selected. When the base feature selectors return their output features, the aggregation operation would start to determine the final feature subset. One of the fundamental challenges in ensemble strategy is how to combine the outputs of the base methods. In the literature, a number of combination approaches exist which are reviewed in [40]. Here, the simple 'OR' operation is utilized for aggregation, and the obtained features at this stage are fed to LIFT. Another crucial issue in ensemble strategy is to choose complementary base feature selectors. For example, if the base feature selectors use the same strategies for selecting features, the output of the ensemble method is similar to the outputs of the base parts.

As in real-world life, that the opinions of several experts usually outperform the individual decisions, the proposed ensemble system is considered to act better. To verify this claim, several experiments are tested on different permutations of the three filter methods.

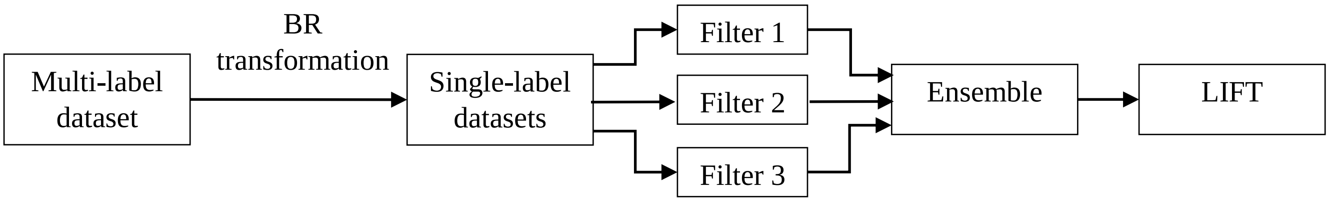

Figure 2. The diagram of the proposed system

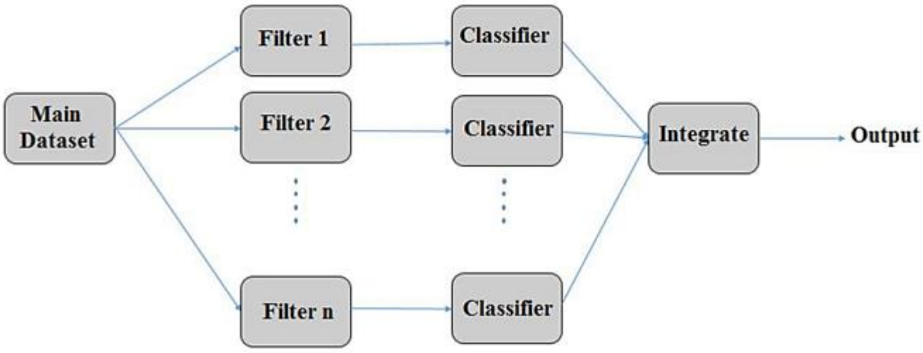

Figure 3. Model 1 of ensemble feature selection strategy [39].

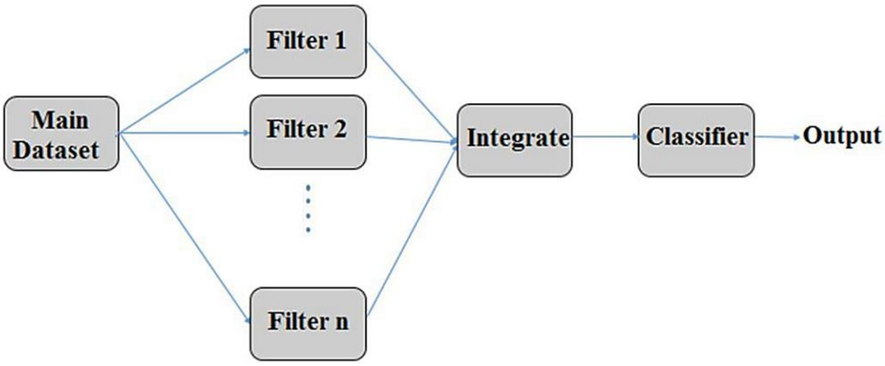

Figure 4. Model 2 of ensemble feature selection strategy [39].

3.3. Pseudo code of the proposed method

For more explanation, the pseudo code of the proposed method is given in Algorithm 1.

Algorithm 1: the proposed method

```
Input: Multi-label dataset $D$ with $M$ features, $N$ samples and $q$ labels
Output: selected features $F$
Transform $D$ to $q$ single-label datasets using BR
for $i = 1:q$
    $Filter1\ (i,:)$ = a binary vector of size $M$ which assigns 1 to selected and 0 to deselected features defined by FCBF method.
    $Filter2\ (i,:)$ = a vector of size $M$ which assigns a weight to each feature using ReliefF method
    $Filter3\ (i,:)$ = a vector of size $M$ which assigns a weight to each feature using IG method
end
$v_1$ = sum the matrix $Filter1$ columns to form a vector of size $M$
$FCBF$ = select features corresponding to non-zero values of $v_1$
$c$ = length of $FCBF$.
$v_2$ = sum the matrix $Filter2$ columns to form a vector of size $M$ and sort it in descending order
$RF$ = select the first $c$ features of $v_2$
$v_3$ = sum the matrix $Filter3$ columns to form a vector of size $M$ and sort it in descending order
$IG$ = select the first $c$ features of $v_3$
$F = (FCBF)\ or\ (RF)\ or\ (IG)$
```

Table 1. Discerption of the datasets used in the experiments

Dataset	N	M	q	Type	LC	LD	Domain
emotions	593	72	6	numeric	1.869	0.311	music
genbase	662	1185	27	nominal	1.252	0.046	biology
medical	978	1449	45	nominal	1.245	0.028	text
enron	1702	1001	53	nominal	3.378	0.064	text
image	2000	294	5	numeric	1.236	0.247	images
scene	2407	294	6	numeric	1.074	0.179	images

4. Experimental studies

This section evaluates the performance of the proposed approach on 6 multi-label datasets from different applications. The results are then compared to the results of the original LIFT algorithm, ML-kNN, and some multi-label feature selection methods.

4.1. Datasets

In the experiments, 6 real multi-label datasets from different applications obtained from the Mulan repository[1] were used. Table 1 summarizes the characteristics of these datasets including dataset name (Dataset); dataset domain (Domain); number of instances (N); number of features (M); number of labels (q=|L|); feature type (Type); label cardinality (LC), which is the average number of labels associated with each instance defined by (2) and label density (LD), which is the cardinality normalized by |L| defined by (3).

$$LC(D) = \frac{1}{|D|}\sum_{i=1}^{|D|}|Y_i| \tag{2}$$

$$LD(D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i|}{|L|} \tag{3}$$

4.2. Performance evaluation criteria

To evaluate the improvement of the proposed approaches compared to the original LIFT algorithm, we employ several evaluation measures popularly use in multi-label tasks, including hamming loss, one-error, coverage, and ranking loss. In summary, these criteria evaluate the learning system’s performance on each test example and then return the mean value across the test set. Let $T = \{(x_i, Y_i), i = 1, \ldots, p\}$ be a given test set where $Y_i \subseteq L$ is a correct label subset, and $Z_i \subseteq L$ be a predicted label set corresponding to t_i. Also, let $f(x, y)$ denotes the score assigned to

[1] - http://mulan.sourceforge.net/datasets.html

label y for sample x. These methods are defined in the following [38]:

Hamming loss

Hamming loss calculates the percentage of labels which are misclassified, i.e. the instance associated to a wrong label or a label belonging to the true sample which is not predicted [41].

$$\text{Hamming Loss}(h, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \Delta Z_i|}{|L|} \qquad (4)$$

where, Δ is the symmetric difference between two sets. Hamming loss computes the percentage of labels whose relevance is not predicted correctly.

One error

This measure counts the number of times that the top-ranked label is not relevant:

$$one-error(f) = \frac{1}{|D|}\sum_{i=1}^{|D|}[[\arg\max_{y\in Y} f(x_i, y)] \notin Y_i] \qquad (5)$$

Coverage

It evaluates the average number of steps to move down in the list of ranked labels to cover all the relevant labels of a sample.

$$\text{coverage}(f) = \frac{1}{|D|}\sum_{i=1}^{|D|}\max_{y\in Y} rank_f(x_i, y) - 1 \qquad (6)$$

where, $rank_f(x_i, y)$ denotes the rank of y in Y based on the descending order induced by *f*.

Ranking loss

Ranking loss counts the average fraction of reversely ordered pairs; i.e. an irrelevant label is ranked higher than a relevant label.

$$rloss(f) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{1}{|Y_i||\bar{Y}_i|}|\{(y', y'') | f(x_i, y') \le f(x_i, y''), (y', y'') \in Y_i \times \bar{Y}_i\}| \qquad (7)$$

Average feature reduction

Another parameter which is used for comparison is the average feature reduction, F_r, to investigate the rate of feature reduction [13].

$$F_r = \frac{M - r}{M} \qquad (8)$$

where, *M* is the total number of features and *r* is the number of selected features by the FS algorithm. The more it is close to 1, the more features are eliminated, which leads to lower classifier's complexity.

Smaller values show better performance for all criteria except average feature reduction. Also, all measures are normalized between 0 and 1 except for coverage.

4.3. Justification

A series of experiments were conducted in order to find the most effective combination of the three filter methods. Table 2 shows the comparison of these methods over several datasets in terms of hamming loss criterion. Numbers written in brackets are the ranks obtained by each algorithm among the others. According to this table, it is observed that the ensemble of the three methods, output the best results. Similar experiments were performed for other evaluation criteria, and the results proved the superiority of the last method, i.e. LIFT_RF_FCBF_IG over the other ones, in average. Therefore, this method is chosen for the feature selection. Among different aggregation strategies discussed in [40], two simple aggregation methods including the AND and OR operators were tested for combining the results of the three filter methods. The experiments on several evaluation criteria showed better results for the OR operator.

The proposed system which is presented in figure 2 with the three filter approaches including IG, FCBF and ReliefF methods in the ensemble phase and the OR operator as the aggregation strategy is called MLIFT, hereafter.

4.4. Results and discussion

During each experiment, 60% of samples were chosen randomly for training. Remaining 40% of samples were used for testing. Results are averaged over 20 independent runs in each dataset and by every algorithm. For implementing FCBF, IG and ReliefF, fspackage [42] is used, which is a package based on Weka [43] and is available to the community at http://featureselection.asu.edu/. LIFT [36][2] is employed with its default parameters, and for ML-kNN the number of nearest neighbours is set to 10.

Table 3, illustrates the results of comparing algorithms including proposed MLIFT (LIFT- RF -FCBF- IG), LIFT, ML-kNN, and four multi-label feature selection methods including LP-RF, LP-IG, BR-RF, and BR-IG presented in [15] over 6 various-sized datasets. The best result among the comparing methods is highlighted in boldface. According to this table, the MLIFT and LIFT algorithms have the best results in all criteria except for *Feature reduction*. Of course, it should be noted that LIFT and ML-kNN are multi-label classifiers which are not expected to reduce the dimensionality of the datasets. Comparing the LIFT and MLIFT algorithms, this table shows that

[2] - http://cse.seu.edu.cn/people/zhangml/Resources.htm#data

MLIFT obtains better results using a smaller feature set. For example, more than 96% of features are eliminated for genbase dataset, and the results remain relatively unchanged compare to the original LIFT algorithm. As mentioned before, even if the results deteriorate slightly for removal of a large number of features, feature selection is still justified. Table 4 shows the average ranks of the comparing algorithms through Friedman 1*N statistical test for each evaluation measure. The last column presents the sum of the ranks for each algorithm and the number written in brackets in the last column shows the total rank of each method. Lower sum of ranks for an algorithm indicates better average results against the others.

The obtained p-values for each measure is also written in this table that shows significant results, as all of the p-values are less than 0.05. According to this table, MLIFT gets the first rank, LIFT is ranked second, ML-kNN is placed in the third position, LP-RF gets rank number 4, both of LP-IG and BR-IG get the fifth rank, and BR-RF is ranked last. Moreover, Zhang [36] proved the superiority of LIFT algorithm over four well-established multi-label learning algorithms, including Bsvm [2], ML_kNN [10], BP_MLL [4] and ECC [44]. Thus, the superiority of the proposed methods over these approaches can also be concluded.

Table 2. The comparison of different ensembles of the three feature selection methods in terms of hamming loss

	LIFT_RF	**LIFT_FCBF**	**LIFT_IG**	**LIFT_RF_FCBF**	**LIFT_IG_FCBF**	**LIFT_IG_RF**	**LIFT_RF_FCBF_IG**
emotions	0.2341[4]	0.2421[6]	0.2394[5]	0.2451[7]	0.2311[2]	**0.2303[1]**	0.2535[3]
genbase	0.0035[6]	0.0029[2]	0.0034[5]	0.0030[3]	0.0033[4]	**0.0027[1]**	0.0033[4]
medical	0.0124[6]	**0.0116[1]**	0.0119[2]	0.0120[3]	0.0122[4]	0.0123[5]	0.0119[2]
image	0.1997[6]	0.1799[4]	0.2152[7]	0.1746[2]	0.1753[3]	0.1913[5]	**0.1616[1]**
scene	0.1136[5]	0.0919[4]	0.1293[7]	0.0867[3]	0.0866[2]	0.1109[6]	**0.0813[1]**

Table 3. Comparison of performance of the algorithms on 6 datasets.

		emotions	**genbase**	**medical**	**enron**	**image**	**scene**
Hamming loss	MLIFT	**0.2535**	**0.0033**	**0.0119**	**0.0467**	0.1616	**0.0813**
	LIFT	0.2622	**0.0033**	0.0132	**0.0467**	**0.1603**	0.0815
	ML-kNN	0.2687	0.0054	0.0163	0.0533	0.8874	0.0905
	BR-RF	0.2655	0.0056	0.0149	0.0530	0.8974	0.0933
	BR-IG	0.2667	0.0057	0.0149	0.0590	0.8941	0.0915
	LP-RF	0.2654	0.0058	0.0185	0.0534	0.8917	0.0929
	LP-IG	0.2627	0.0051	0.0157	0.0626	0.8917	0.0927
One error	MLIFT	**0.3609**	0.0007	**0.1626**	0.2529	**0.2836**	**0.2030**
	LIFT	0.3738	**0.0003**	0.1820	**0.2444**	0.2839	0.2073
	ML-kNN	0.3867	0.0124	0.2758	0.3245	0.3324	0.2380
	BR-RF	0.3907	0.0100	0.2682	0.3190	0.3523	0.2462
	BR-IG	0.3890	0.0091	0.2313	0.3951	0.3496	0.2405
	LP-RF	0.3905	0.0113	0.4285	0.3214	0.3369	0.2462
	LP-IG	0.3992	0.0119	0.2583	0.4647	0.3392	0.2424
Coverage	MLIFT	**2.1736**	**0.5284**	**2.0416**	12.6290	0.8813	**0.4094**
	LIFT	2.2179	**0.5284**	2.2205	**12.4910**	**0.8736**	0.4209
	ML-kNN	2.3042	0.5775	3.0092	13.6460	0.9920	0.4953
	BR-RF	2.2861	0.6704	3.2309	13.5775	1.0519	0.5166
	BR-IG	2.2802	0.7221	4.8145	14.5877	1.0445	0.5039
	LP-RF	2.2688	0.6492	3.1313	13.4831	1.0070	0.5058
	LP-IG	2.1865	0.7211	3.2198	15.4910	1.0268	0.5090
Ranking loss	MLIFT	**0.2374**	**0.0053**	**0.0292**	0.0830	0.1516	**0.0654**
	LIFT	0.2453	0.0056	0.0326	**0.0815**	**0.1514**	0.0672
	ML-kNN	0.2632	0.0069	0.0481	0.0963	0.1817	0.082
	BR-RF	0.2632	0.0084	0.0523	0.0958	0.1932	0.0856
	BR-IG	0.2602	0.0098	0.0857	0.1091	0.1946	0.0839
	LP-RF	0.2590	0.0082	0.0502	0.0955	0.1842	0.0841
	LP-IG	0.2450	0.0104	0.0514	0.1190	0.1891	0.0840
Feature reduction	MLIFT	0.1319	0.9629	0.7696	0.7115	0.057	0.0215
	LIFT	0	0	0	0	0	0
	ML-kNN	0	0	0	0	0	0
	BR-RF	**0.4575**	0.9601	0.8669	0.0298	0.5694	0.2274
	BR-IG	0.2027	**0.9789**	**0.9964**	**0.9977**	0.1262	0.0264
	LP-RF	0.1545	0.9681	0.9365	0.0034	**0.4844**	**0.2277**
	LP-IG	0.1201	0.9763	0.9870	0.7255	0.2571	0.0219

Table 1. Average rankings of the algorithms obtained by each evaluation measure by performing Friedman test.

Statistical *Test*	**method**	**Hamming loss** p-value = 0.000607	**One error** p-value = 0.000481	**Coverage** p-value = 0.000278	**Ranking loss** p-value = 0.000309	**Feature reduction** p-value = 0.000209	**Sum of ranks**
Friedman 1*N	**MLIFT**	**1.3333[1]**	**1.3333[1]**	**1.4167[1]**	**1.3333[1]**	4.3333[4]	**8[1]**
	LIFT	1.6667[2]	1.6667[2]	1.7500[2]	1.8333[2]	6.5000[5]	13[2]
	ML-kNN	4.5000[3]	4.5000[4]	4.0000[3]	3.9167[3]	6.5000[5]	18[3]
	BR-RF	5.0833[5]	5.2500[5]	5.8333[5]	5.7500[6]	2.8333[2]	23[6]
	BR-IG	5.2500[6]	4.3333[3]	5.8333[5]	5.8333[7]	**2.0000[1]**	22[5]
	LP-RF	5.5833[7]	5.2500[5]	4.0000[3]	4.1667[4]	2.8333[2]	21[4]
	LP-IG	4.5833[4]	5.6667[6]	5.1667[4]	5.1667[5]	3.000[3]	22[5]

The obtained p-values for each measure is also written in this table that shows significant results, as all of the p-values are less than 0.05. According to this table, MLIFT gets the first rank, LIFT is ranked second, ML-kNN is placed in the third position, LP-RF gets rank number 4, both of LP-IG and BR-IG get the fifth rank, and BR-RF is ranked last. Moreover, Zhang [36] proved the superiority of LIFT algorithm over four well-established multi-label learning algorithms, including Bsvm [2], ML_kNN [10], BP_MLL [4] and ECC [44]. Thus, the superiority of the proposed methods over these approaches can also be concluded.

5. Conclusion

This paper proposes a modification to LIFT [36] algorithm which is a multi-label learning strategy via label-specific features. More precisely, LIFT reduces the dimension of samples using the information of their labels. However, to construct the new features, the original features of each sample are needed. Therefore, the problems related to costly, irrelevant and redundant features still remain. To overcome this challenge, we suggest to remove irrelevant and redundant features before the LIFT algorithm. To do so, the ensemble strategy which is one of the promising techniques in single-label feature selection is employed to select the most salient features in multi-label data. Firstly, the multi-label data is transform into single-label data using the BR method. Then, the ensemble of three well-known single-label filter approaches, including IG, ReliefF and FCBF are employed and the results are aggregated using the OR operator. The experimental results show that in spite of eliminating a significant number of features, the proposed method has better performance compared to the LIFT algorithm and other comparing methods.

References

[1] Yang, J., Jiang, Y.-G., Hauptmann, A. G. & Ngo, C.-W. (2007). Evaluating bag-of-visual-words representations in scene classification, In Proceedings of the international workshop on Workshop on multimedia information retrieval, pp. 197-206.

[2] Boutell, M. R., Luo, J., Shen, X. & Brown, C. M. (2004). Learning multi-label scene classification, Pattern recognition, vol. 37, pp. 1757-1771.

[3] Diplaris, S., Tsoumakas, G., Mitkas, P. A., & Vlahavas, I. (2005). Protein classification with multiple algorithms, In Panhellenic Conference on Informatics, 2005, pp. 448-456.

[4] Zhang, M.-L., & Zhou, Z.-H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. IEEE transactions on Knowledge and Data Engineering, vol. 18, pp. 1338-1351.

[5] Luo, Q., Chen, E., & Xiong, H. (2011). A semantic term weighting scheme for text categorization. Expert Systems with Applications, vol. 38, pp. 12708-12716.

[6] Trohidis, K., Tsoumakas, G., Kalliris, G., & Vlahavas, I. P. (2008). Multi-Label Classification of Music into Emotions. In ISMIR, pp. 325-330.

[7] Spolaôr, N., Cherman, E. A., Monard, M. C., & Lee, H. D. (2012). Filter approach feature selection methods to support multi-label learning based on relieff and information gain. In Advances in Artificial Intelligence-SBIA 2012, ed: Springer, 2012, pp. 72-81.

[8] Cherman, E. A., Monard, M. C., & Metz, J. (2011). Multi-label problem transformation methods: a case study. CLEI Electronic Journal, vol. 14, pp. 4-4.

[9] Zhang, M.-L., Peña, J. M., & Robles, V. (2009). Feature selection for multi-label naive Bayes classification. Information Sciences, vol. 179, pp. 3218-3229.

[10] Zhang, M.-L., & Zhou, Z.-H. (2007). ML-KNN: A lazy learning approach to multi-label learning. Pattern recognition, vol. 40, pp. 2038-2048.

[11] De Comité, F., Gilleron, R., & Tommasi, M. (2003). Learning multi-label alternating decision trees from texts and data. In International Workshop on Machine Learning and Data Mining in Pattern Recognition, 2003, pp. 35-49.

[12] Lin, Y., Hu, Q., Liu, J., & Duan, J. (2015). Multi-label feature selection based on max-dependency and min-redundancy. Neurocomputing, vol. 168, pp. 92-103.

[13] Kashef, S., & Nezamabadi-pour, H. (2015). An advanced ACO algorithm for feature subset selection. Neurocomputing, vol. 147, pp. 271-279.

[14] Kashef, S., & Nezamabadi-pour, H. (2013). A new feature selection algorithm based on binary ant colony optimization. In Information and Knowledge Technology (IKT), 2013 5th Conference on, 2013, pp. 50-54.

[15] SpolaôR, N., Cherman, 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, vol. 292, pp. 135-151.

[16] Ding, S. (2009). Feature selection based F-score and ACO algorithm in support vector machine. In Knowledge Acquisition and Modeling, 2009. KAM'09. Second International Symposium on, 2009, pp. 19-23.

[17] Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. In Proceedings of the ninth international workshop on Machine learning, 1992, pp. 249-256.

[18] Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In Proceedings of the 20th international conference on machine learning (ICML-03), 2003, pp. 856-863.

[19] Rouhi, A., & Nezamabadi-pour, H. (2016). A hybrid method for dimensionality reduction in microarray data based on advanced binary ant colony algorithm. In 2016 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC), 2016, pp. 70-75.

[20] Lee, J., & Kim, D.-W. (2013). Feature selection for multi-label classification using multivariate mutual information. Pattern Recognition Letters, vol. 34, pp. 349-357.

[21] Lee, J., & Kim, D.-W. (2015). Mutual information-based multi-label feature selection using interaction information. Expert Systems with Applications, vol. 42, pp. 2013-2025.

[22] Reyes, O., Morell, C., & Ventura, S. (2015). Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context. Neurocomputing, vol. 161, pp. 168-182.

[23] Gharroudi, O., Elghazel, H., & Aussem, A. (2014). A comparison of multi-label feature selection methods using the random forest paradigm. In Canadian Conference on Artificial Intelligence, 2014, pp. 95-106.

[24] Cheng, H., Deng, W., Fu, C., Wang, Y., & Qin, Z. (2011). Graph-based semi-supervised feature selection with application to automatic spam image identification. Computer Science for Environmental Engineering and EcoInformatics, pp. 259-264.

[25] Lee, J., & Kim, D.-W. (2015). Memetic feature selection algorithm for multi-label classification. Information Sciences, vol. 293, pp. 80-96.

[26] Chen, W., Yan, J., Zhang, B., Chen, Z., & Yang, Q. (2007). Document transformation for multi-label feature selection in text categorization. In Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on, 2007, pp. 451-456.

[27] Yan, J., Liu, N., Zhang, B., Yan, S., Chen, Z. & Cheng, Q. et al. (2005). OCFS: optimal orthogonal centroid feature selection for text categorization. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, 2005, pp. 122-129.

[28] Doquire, G., & Verleysen, M. (2011). Feature selection for multi-label classification problems. In International Work-Conference on Artificial Neural Networks, 2011, pp. 9-16.

[29] Read, J., Pfahringer, B., & Holmes, G. (2008). Multi-label classification using ensembles of pruned sets. In 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 995-1000.

[30] Reyes, O., Morell, C., & Ventura, S. (2013). ReliefF-ML: an extension of reliefF algorithm to multi-label learning. In Iberoamerican Congress on Pattern Recognition, 2013, pp. 528-535.

[31] SpolaôR, N., Cherman, E. A., Monard, M. C., & Lee, H. D. (2013). Relief for multi-label feature selection. IEEE Brazilian Conference on Intelligent Systems (BRACIS), pp. 6-11, 2013.

[32] Lastra, G., Luaces, O., Quevedo, J. R., & Bahamonde, A. (2011). Graphical feature selection for multilabel classification tasks. In International Symposium on Intelligent Data Analysis, 2011, pp. 246-257.

[33] Kashef, S., & Nezamabadi-pour, H. (2017). An effective method of multi-label feature selection employing evolutionary algorithms. In Swarm Intelligence and Evolutionary Computation (CSIEC), 2017 2nd Conference on, 2017, pp. 21-25.

[34] Lin, Y., Hu, Q., Liu, J., Chen, J., & Duan, J. (2016). Multi-label feature selection based on neighborhood mutual information. Applied Soft Computing, vol. 38, pp. 244-256.

[35] Kashef, S., Nezamabadi-pour, H., & Nikpour, B. "Multi-label feature selection: a comprehensive review and guiding experiments. Accepted for Publication in WIREs Data Mining and Knowledge Discovery, 28 Nov. 2017.

[36] Zhang, M.-L., & Wu, L. (2015). LIFT: Multi-label learning with label-specific features. IEEE transactions on pattern analysis and machine intelligence, vol. 37, pp. 107-120.

[37] Spolaôr, N., Monard, M. C., Tsoumakas, G., & Lee, H. D. (2016). A systematic review of multi-label

feature selection and a new method based on label construction. Neurocomputing, vol. 180, pp. 3-15.

[38] Zhang, L., Hu, Q., Duan, J., & Wang, X. (2014). Multi-label feature selection with fuzzy rough sets. In International Conference on Rough Sets and Knowledge Technology, 2014, pp. 121-128.

[39] Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2012). An ensemble of filters and classifiers for microarray data classification. Pattern Recognition, vol. 45, pp. 531-539.

[40] Mousavi, R., & Eftekhari, M. (2015). A new ensemble learning methodology based on hybridization of classifier ensemble selection approaches. Applied Soft Computing, vol. 37, pp. 652-666.

[41] Cherman, E. A., Spolaôr, N., Valverde-Rebaza, J., & Monard, M. C. (2015). Lazy multi-label learning algorithms based on mutuality strategies. Journal of Intelligent & Robotic Systems, vol. 80, pp. 261-276.

[42] Liu, H. (2010). Feature Selection at Arizona State University, Data Mining and Machine Learning Laboratory, Last access: October, 2010.

[43] Hall, M., Witten, I., & Frank, E. (2011). Data mining: Practical machine learning tools and techniques, Kaufmann, Burlington, 2011.

[44] Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011). Classifier chains for multi-label classification. Machine learning, vol. 85, p. 333.