Journal of AI and Data Mining
Vol 7, No 1, 2019, 35-45
DOI: 10.22044/JADM.2018.5306.1652

Exploiting Associations between Class Labels in Multi-label Classification

Z. Mirzamomen* and Kh. Ghafooripour

Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran.

Received 08 March 2017; Revised 25 November 2017; Accepted 06 January 2018
**Corresponding author:mirzamomen@srttu.edu (Z. Mirzamomen).*

Abstract
Multi-label classification has many applications in the text categorization, biology, and medical diagnosis, in which multiple class labels can be assigned to each training instance simultaneously. As it is often the case that there are relationships between the labels, extracting the existing relationships between the labels and taking advantage of them during the training or prediction phase can bring about significant improvements. In this paper, we introduce positive, negative, and hybrid relationships between the class labels for the first time, and propose a method to extract these relations for a multi-label classification task, and to use them consequently in order to improve the predictions made by a multi-label classifier. We conduct extensive experiments to assess the effectiveness of the proposed method. The results obtained advocate the merits of the proposed method in improving the multi-label classification results.

Keywords: *Classification, Multi-label Classification, Label Relationships, Association Rule, Positive Relation, Negative Relation, Hybrid Relation.*

1. Introduction

Multi-label classification is an extension of the traditional (single-label) classification task, in which multiple class labels can be assigned to each training instance simultaneously. Most of the recent applications of data mining and machine learning including text and web categorization, image and video tagging, and medical diagnosis should be solved as a multi-label classification. It is worth mentioning that multi-label classification is different from multi-class classification, which is a single-label classification with more than two classes.

In the real multi-label classification tasks, it would be natural to assume some relationships or associations between the class labels. For example, when a photo from a natural scene is tagged with the label "boat", the probability of the labels "sea" or "river" for it would be much more than the label "desert". More examples can be found in the medical or biology applications with the common symptoms of diseases or the common genetic causes of illnesses.

Therefore, identifying the relationships between the class labels and using them in the training or predicting phases can promote the performance of multi-label classification. However, most of the existing multi-label classification methods simply ignore the existence of such relationships [1].

In the recent research works, taking advantage of the relationships between the class labels in multi-label classification has been the subject of interest [2-5]. In [3], association rules have been proposed to model such relationships and to employ them in order to correct the erroneous predictions made by the classifiers.

In this paper, we have extended the way of employing the association rules in order to be able to model more extensive relationships between the class labels. Consequently, the proposed method enables more extensive corrections to be made in the prediction phase.

The main contributions of this paper are as follow:

1. For each specific training instance, the proposed method not only contributes to its assigned labels in modeling the relationships between the labels but also contributes to the not-assigned labels.
2. This paper has extended the notation of the association rules in order to express a broader range of associations between the class labels.

3. The proposed method not only enables to correct the wrongly not-assigned labels in the prediction phase by simply adding them to the list of predicted labels but also enables to correct the wrongly assigned labels.

The rest of the paper is organized as follows: In Section 2, the related works are presented. Section 3 briefly introduces the background knowledge of the frequent pattern mining and association rule analysis. In Section 4, we introduce our proposed methods and algorithms. Section 5 is dedicated to the experiments and analysis of their results. Section 6 summarizes the main contributions of this paper, and gives the concluding remarks.

2. Related works

Multi-label classification has been the subject of interest in the recent years and many algorithms and methods proposed in the literature in this context [6-12].

There are two main approaches available to solve a multi-label classification problem: problem transformation and algorithm adaption [7]. In the former approach, the multi-label problem is transformed into some traditional single-label classification problems, each of which can be solved by employing the traditional classification methods. Consequently, to classify a new instance, the predictions made by the built classifiers should be aggregated. In the latter approach, the traditional single-label classification algorithms are extended to directly solve the multi-label problems [13, 14].

Binary Relevance (BR) [1] is the simplest method based on the first approach, in which there is a separate classifier for each class label in order to distinguish the instances of that class from the others. The rationale of the BR method is similar to the One Versus All (OVA) method [15] in the single-label classification context. The main challenge of this method is the imbalance of the training set of the base classifiers. In addition, the relationships between the class labels are not considered in BR.

Classifier Chain (CC) [16] is an extension of the BR method, in which a fixed order (e.g. a random order) between the labels is assumed and a chain of classifiers is built based on it such that the output (prediction) of the Kth classifier would be added to the feature space of the $K + 1$th classifier.

Bayesian Classifier Chain (BCC) [17] is an extension of CC, in which the order of the classifiers is determined based on a Bayesian network.

The CC and BCC methods can model the relationship of each class label with just one other label. The classifier Trellises (CT) [18] method can model the relationship of each class label with two other labels in a trellis structure, in which the mutual information is employed to determine the degree of relationship between the labels.

In [2], the correlation between the class labels of the multi-label data set is modeled by association rules. The extracted rules are used in the prediction phase to add the wrongly not-assigned labels to the final prediction made by the classifier. In [19], the association rules extracted from the label information is used to reduce the total number of labels of the problem. In [11], the k-means clustering algorithm is first employed on the datasets with numerical attributes (as a preprocessing step) to make the dataset ready for mining association rules.

There are also a broad family of methods and algorithms regarding hierarchical multi-label classification, which assume that a given hierarchical relationship exists between the class labels [4, 20, 12].

3. Background knowledge: Association analysis

Frequent pattern mining [21] is one of the most frequently used data mining techniques for finding the relationships between the data items. The extracted frequent patterns are usually represented by some association rules. Support and Confidence are the two most important parameters in mining association rules. An example association rule extracted during the market basket analysis can be:

$$(\text{laptop} \rightarrow \text{wireless mouse})$$
$$[\text{support: } 20\%, \text{ confidence: } 80\%]$$

which conveys that laptop and wireless mouse are sold in 20% of the transactions, and 80% of the time, the laptop buyer also has bought a wireless mouse. The favorite rules are those with high support and high confidence, and hence, the minimum acceptable support and confidence way should be determined at the beginning of the frequent pattern mining process. As a matter of fact, there is a trade-off between the support and confidence parameters such that maximizing one of them would result in the decrease in the other one. Thus determining the best values for these parameters would be important.

There are several methods for mining frequent patterns such as the Apriori [22] and FP-Growth[1] [23]. FP-Growth algorithm is the extension of the Apriori, in which the computational cost is reduced by eliminating the need for passing

[1] Frequent Pattern Growth

through the dataset several times. In FP-Growth algorithm, the dataset is processed only one time and a tree data structure is created based on it. This algorithm is well-described in [21].

4. Proposed method

In this section, we first introduce the symbols and notations used in the rest of the paper, and then will introduce the proposed method in great details.

4.1. Notations

Table 1 presents the notations used in the rest of the paper, along with their description. In this paper, we have introduced the positive, negative, and hybrid relationships between the class labels for the first time, and have used the last three symbols of table 1 to represent their corresponding association rules.

Table 1. Notations used in this paper.

Symbol	Description
x	The d dimensional feature vector of a training instance
q	The total number of class labels in the dataset.
y_i	The i'th label of the dataset
y	The set of class labels of the dataset: $y = \{y_1, y_2, \ldots, y_q\}$
Y(x)	The set of real labels of x such that $Y(x) \subset y$
Y'(x)	The set of labels not belonging to x such that $Y'(x) = y - Y(X)$
h(x)	The set of labels predicted for x by a multi-label classifier
R^+	The set of positive association rules
R^-	The set of negative association rules
R^c	The set of hybrid association rules

4.2. Title types of relationships: definition and modeling

In this paper, we have employed frequent pattern mining to find the associations between the class labels. Generally, this can be done by considering the labels of each training instance as the items. This way, an itemset would be created from a multi-label training dataset, and afterwards, the frequent pattern mining process can be started on it.

Our novelty in extending the previous methods is to contribute the not-assigned labels of the training instances along with their assigned labels in extracting the frequent patterns, which enable us to model more extensive relationships between the labels. Therefore, the itemset corresponding to a training instance (x,Y(x)) can be defined as (1), in which the ~ sign appears beside the labels that do not belong to the training instance.

$$\{L_1 L_2 \ldots L_q\} : \begin{cases} L_i = \sim y_i \text{ if } y_i \in Y'(x) \\ L_i = y_i \text{ if } y_i \in Y(x) \end{cases} \quad (1)$$

For example, in a problem with $y = \{y_1, y_2, \ldots, y_6\}$, if we had $Y(x) = \{y_1, y_2, y_5\}$ for a training instance x, then the corresponding itemset of x would be: $\{y_1 y_2 : y_3 : y_4 y_5 : y_6\}$.

In this paper, we have introduced the concepts of positive, negative, and hybrid relationships between the class labels as what follow.

Positive relationship: It is the relationship between the labels in the Y (x) sets of the instances in the training set, which can represent the frequent co-occurrence of some labels in the instances together. We have called the association rules representing such relationships as positive association rules. An example of the positive association rule can be as follows:

$$y_1 y_5 \rightarrow y_2$$

which can be interpreted as follows: if an instance belongs to the y_1 and y_5 classes simultaneously, then it also belongs to class y_2 with high confidence.

Negative relationship: It is the relationship between the labels in the Y'(x) sets of the instances in the training set, which can represent the frequent co-occurrence of not-existence of some labels in the instances. We have called the association rules representing such relationships as negative association rules. An example of the negative association rule can be as follows:

$$: y_6 \rightarrow : y_3$$

which can be interpreted as follows: if an instance does not belong to the y_6 class, then it also does not belong to class y_3 with high confidence.

Hybrid relationship: It is the relationship between the labels in the Y(x) U Y'(x) sets of the instances in the training set, which can represent different co-occurrence types including the frequent occurrence of existence of some labels with not-existence of some other labels, and vice versa. We have called the association rules representing such relationships as the hybrid association rules. An example of the hybrid association rule is:

$$y_3 \rightarrow : y_6$$

which has the following interpretation: if an instance belongs to the y_3 class, then it does not belong to class y_6 with high confidence. As another example, the interpretation of the below hybrid rule is as follows: if an instance belongs to the y_3 class but it does not belong to the y_2, then it belongs to the class y_1 with high confidence.

$$y_3 : y_2 \rightarrow y_1$$

4.3. Method description

The general steps of the proposed method are as follow:

1. Prepare the itemsets based on the labels of instances in the training dataset.

2. Extract the association rules of the three possible types.
3. Filter the extracted rules and keep the high quality rules (e.g. the ones with high support and confidence).
4. Apply the final rules in the prediction phase in order to correct the errors (where possible) to improve the classification results.

Algorithm 1 presents the first two steps required for extracting the association rules from a multi-label training dataset. The lines 1-12 of this algorithm are concerned with the main idea of this paper, which enables extracting positive, negative, and hybrid relationships between the labels. The initial positive/negative/hybrid itemset repository is empty, and it would be filled up using the training instances (lines 8-12 of the algorithm) according to (1).

In Algorithm 1, the FP-Growth algorithm [23] is employed for mining association rules from the three itemset repositories but this is an arbitrary choice and can be replaced with any other frequent itemset mining algorithm. (Please refer to Section 3 for more explanation about the FP-Growth algorithm.)

The extracted rules can be investigated by the domain expert, and the ones that seem to be erroneous can be discarded. The final association rules can be used to correct the two possible types of error made by any multi-label classifier. Assume a new instance x and the label y_i and a multi-label classifier H of any type. If h(x) is the set of labels predicted for x by H, then two types of errors would be possible:

1. $y_i \in Y(x)$ but $y_i \notin h(x)$
2. $y_i \notin Y(x)$ but $y_i \in h(x)$

The proposed method can help to fix both of the above errors such that:

- The positive rules can fix the first type of error.
- The negative rules can fix the second type of error.
- The hybrid rules can fix the errors of both types.

Algorithm 2 presents the way of employing the extracted rules to improve the predictions of a multi-label classifier. R^+, R^-, and R^c are the three sets of extracted association rules obtained as the output of Algorithm 1. The following section illustrates the functionality of Algorithm 2 by providing an example.

Algorithm 1. Extracting positive, negative, and hybrid association rules from a multi-label dataset by employing FP Growth algorithm.

1: **function** Extraction (Dataset D) returns R^+, R^-, R^c;
2: ► R^+ Positive association rules;
3: ► R^- Negative association rules;
4: ► R^c Complex association rules;
5: Initialize the positive itemset repository P_itemsets;
6: Initialize the negative itemset repository N_itemsets;
7: Initialize the hybrid itemset repository H_itemsets;
8: **for** each instance x in D **do**
9: Add a record to P_itemsets consisting of the labels in Y (x).
10: Add a record to N_itemsets consisting of the labels in Y'(x).
11: Add a record to H_itemsets consisting of the labels in Y (x) U Y'(x) according to Eq. (1).
12: **end for**
13: Set the minimum support and minimum confidence for FP_Growth algorithm.
14: $R^+ \leftarrow$ FP_Growth (P_itemsets);
15: $R^- \leftarrow$ FP_Growth (N_itemsets);
16: $R^c \leftarrow$ FP_Growth (H_itemsets);
17: **return** R^+, R^-, R^c;
18: **end function**

Algorithm 2. Improving the predictions of a multi-label classifier using the extracted association rules.

1: **function** Correction By Rules (h(x)) returns h(x);
2: ► R^+ Positive association rules;
3: ► R^- Negative association rules;
4: ► R^c Complex association rules;
5: Apply R^+ on the labels in h(x) and add the required labels to h(x);
6: Apply R^- on the labels in h(x) and remove the erroneous labels from h(x);
7: Apply R^c on the labels in h(x) and add/remove the required erroneous labels to/from h(x);
8: **return** h(x);
9: **end function**

4.4. Illustrative examples

Assume that the following rules are extracted from a multi-label training dataset by applying Algorithm 1:

$r_1 \in R^+ : y_p \rightarrow y_k$

$r_2 \in R^- : : y_n \rightarrow : y_m$

$r_3 \in R^c : y_p, : y_n \rightarrow : y_t, y_x$

Also assume that $h(x) = (y_m, y_p)$, which is the labels predicted for x by a multi-label classifier. According to Algorithm 2, the following corrections are possible:

- $h(x) = h(x) \cup y_k$ based on r_1.
- $h(x) = h(x) - y_m$ based on r_2.
- $h(x) = h(x) + y_x$ based on r_3.

Meantime, it is worth mentioning that the above example is just a simple case, and in the general case, there is no limitation on the number of labels in the hypothesis or the conclusion parts of the rules.

4.5. Time complexity

The time complexity of the proposed method can be analyzed in two phases: first, in the phase of extracting the association rules, and second, in the phase of applying these rules to the predictions made by a classifier (post processing). In the first phase, the time complexity of the proposed method is the same with the FP-Growth algorithm. In the second phase, the worst case time complexity would be O(r), in which r indicates the number of high-quality extracted rule, which would be a fixed constant number.

5. Experimental evaluation and analysis

We conducted extensive experiments to evaluate the effectiveness of the proposed method. We used the 10-fold cross validation methodology in all the experiments.

The significance of the observed differences in the performance metrics was tested with the Friedman test [24, 25] to compare multiple classifiers on multiple datasets based on average ranks, as suggested by Demsar [26]. When the null hypothesis was rejected, we used the posthoc Nemenyi test [26].

We extensively used the Weka [27] and Meka [28] (the multi-label extension of Weka) frameworks in order to implement the proposed method and compare it with the rival algorithms. All the experiments were done on a 1.4GH linux machine with 4 GB memory.

5.1. Rival algorithms

In this paper, we used the BR [1] method as the base multi-label classifier for evaluating our proposed method. In BR, we used the j48 [29] decision tree as the base classifier, with the default settings in Weka. We used the FP-Growth [23] with its default settings in Weka (confidence = 0.9) in order to extract the association rules.

We compared the performance of the CC [16], BCC [17], CT [18], and LP [8] algorithms with the following ones:

- BRP: The BR method improved using the positive association rules only (proposed in [3]).
- BRH: The BR method improved using both the positive and negative, the and hybrid association rules (the proposed method).
- BR: The original BR method.

5.2. Datasets

Table 2 shows the datasets used in our experiments, along with their specifications. These datasets have been commonly used in the multi-label classification research works [1, 6, 16-18, 30, 31].

Table 2. Datasets used in experiments along with their specification.

Name	Cate	No.	Labels	Cardinal	Density	Distinct
Yeast	biology	2417	14	4.237	0.303	198
Scene	image	2407	6	1.014	0.0352	133
Birds	voice	645	19	1.014	0.179	15
Music	voice	592	6	1.869	0.311	27
Flags	image	194	7	3.392	0.485	54
Emotions	voice	593	7	1.869	0.311	27
Genbass	biology	662	27	1.252	0.046	32
Cal500	voice	502	174	26.044	0.15	502
tmc2007	text	28956	22	2.158	0.098	1341

5.3. Metrics

The available metrics for evaluating multi-label classifiers can be divided into two categories: instance-based metrics and label-based metrics. We used the following live commonly used metrics in our experiments:

Subset accuracy: It is an instance-based metric, which should be maximized. According to (2), in which p is the number of test instances, this measure would be maximized if for all the test instances, the predicted labels were equal to the true labels.

$$\text{Subset Accuracy} : \frac{1}{p}\sum_{i=1}^{p}\left[h(x_i)=\mathrm{Y}(x_i)\right] \quad (2)$$

Accuracy exam: It is an instance-based metric, which should be maximized as well. According to (3), in which p is the number of test instances, this measure would be maximized again if for all the test instances, the predicted labels are equal to the

true labels. However, this is less strict than the subset accuracy because it can output a number between zero and one for each instance if the predicted labels do not exactly match, while the subset accuracy outputs zero in such cases.

$$\text{Accuracy exam}: \frac{1}{p}\sum_{i=1}^{p}\frac{\left|h(x_i)\text{I } Y(x_i)\right|}{\left|h(x_i)\text{UY}(x_i)\right|} \quad (3)$$

Average precision: This metric is also an instance-based metric that should be maximized. This metric assumes that for each test instance, the classifier outputs a ranking for each label such that the higher the ranking of a label, the higher the probability that the label belong to the instance. According to (4), in which p is the number of test instances, this measure would be maximized if for all the test instances, the ranking of all of the true labels (in the prediction of the classifier) was higher than the ranking of the other labels.

Average Precision:

$$\frac{1}{p}\sum_{i=1}^{p}\frac{1}{\left|Y_i\right|}\sum_{y\in Y_i}\frac{\left|\left\{y'|\ \text{rank}_f(x_i,y)\le \text{rank}_f(x_i,y), y'\in Y_i\right\}\right|}{\text{rank}_f(x_i,y)} \quad (4)$$

F1-micro: This metric is the extension of the well-known F-measure for multi-label classification, averaging on the prediction matrix. It can be calculated according to (5), in which p is the number of the test instances and q is the number of labels. In addition, $y^j(x_i)$ is one if the labels y^j relevant to the instance x_i, and zero otherwise. Also $h(x_i)^j$ is one if the labels y^j predicted for the instance x_i, and zero otherwise. This metric should be maximized.

$$\text{F1-micro}: \frac{2\sum_{j=1}^{q}\sum_{i=1}^{p}y^j(x_i)h(x_i)^j}{\sum_{j=1}^{q}\sum_{i=1}^{p}y^j(x_i)+\sum_{j=1}^{q}\sum_{i=1}^{p}h(x_i)^j} \quad (5)$$

Hamming loss: Equation (6) shows the hamming loss metric, in which Δ represents the symmetric difference between two sets. This metric should be minimized. In the worst case, the value of this metric for a test instance would be equal to the number of true labels of it plus the number of labels predicted for it, which occurs when the classifier fails to predict even one of the true labels of the test instance.

$$\text{Hamming loss}: \frac{1}{p}\sum_{i=1}^{p}\left|h(x_i)\Delta Y(x_i)\right| \quad (6)$$

5.4. Observations and analysis

Tables 3 to 7 present the results of the experiments based on different metrics.

In this section, we analyzed the results obtained. From the results, we not only can investigate the effect of applying the proposed method on the BR classifier but also can compare the overall performance of the resulting classifier with five other classifiers, each are proposed to somehow improve the BR method.

First, it is worth mentioning that no positive association rule (with confidence = 90%) was found for the Music, Birds, Scene, Genbase, Emotions, and Tmc datasets, and hence, the results of BR and BRP would be similar for these datasets. Meantime, in the CALL500 dataset, the set of the labels of each instance is different from the other ones. In other words, no two instances have the same set of labels. Thus no improvement is made by the proposed method.

Table 3 shows the "accuracy exam" of the rival algorithms, and table 4 shows the "subset accuracy" of them on different datasets. It can be seen that the proposed method has improved the BR results on 6 out of 9 datasets, while the BRP method (that only uses the positive relationships) has improved the BR results only on two datasets. Similar results can be seen in table 4, which shows the subset accuracy of the rival algorithms such that the proposed method has improved the BR results on 6 out of 9 datasets, while the BRP method has improved the BR results only on one datasets.

In addition, the results tabulated in table 3 show that the proposed BRH method has achieved the best results among all the rivals on five datasets, while the BR method has not been the winner at all. Moreover, the results tabulated in table 4 show that the BRH method has achieved the best results among all the rivals on three datasets, while BRP has not been the winner at all. This advocates the effectiveness of using the negative and hybrid relationships proposed in this article in improving the multi-label classification results.

However, comparing the average ranks of table 3 with the Friedman test, we obtained X_F^2= 19.31 and F_F = 4.45 with critical value 2.3 at the 0.05 critical level, and so we could reject the null hypothesis, which means that there is a significant difference among the rival algorithms. The result of the post-hoc Nemenyi test with critical distance CD = 3.00 at the 0.05 critical level is that the LP algorithm is significantly better than the BR, BRP, and CT algorithms. This means that although LP has achieved the best average rank, there is no statistically significant difference between it and the proposed BRH algorithm.

The Friedman test results for table 4 were X_F^2 = 6.12 and F_F = 1.02 with critical value 2.3 at the 0.05 critical level, and so we could not reject the null hypothesis.

Table 3. Average and standard deviation of "accuracy exam" of rival algorithms. Best results are shown boldface.

Dataset	BR	BRP	BRH	CC [16]	BCC [17]	CT [18]	LP [8]
Yeast	44.0 ± 0.02	46.3 ± 0.02	**47.0 ± 0.02**	42.9 ± 0.02	41.9 ± 0.02	42.1 ± 0.02	41.1 ± 0.02
Scene	53.5 ± 0.03	53.5 ± 0.03	57.1 ± 0.03	54.8 ± 0.03	58.2 ± 0.02	54.7 ± 0.03	**58.8 ± 0.03**
Birds	57.3 ± 0.05	57.3 ± 0.05	57.0 ± 0.05	56.3 ± 0.06	57.5 ± 0.05	57.1 ± 0.05	**57.9 ± 0.07**
Music	53.8 ± 0.05	53.8 ± 0.05	**56.6 ± 0.04**	54.3 ± 0.05	54.1 ± 0.06	54.0 ± 0.04	51.1 ± 0.05
Flags	59.1 ± 0.05	59.5 ± 0.05	**61.7 ± 0.05**	59.7 ± 0.06	59.3 ± 0.04	57.5 ± 0.05	59.2 ± 0.06
Emotions	53.9 ± 0.03	53.9 ± 0.03	**55.7 ± 0.03**	53.8 ± 0.03	54.1 ± 0.03	54.1 ± 0.04	52.3 ± 0.06
tmc2007	**61.7 ± 0.01**	61.7 ± 0.01	60.1 ± 0.01	61.5 ± 0.01	61.5 ± 0.01	61.6 ± 0.01	57.3 ± 0.01
CAL500	20.7 ± 0.01	20.7 ± 0.01	20.7 ± 0.02	**21.7 ± 0.02**	21.5 ± 0.01	21.5 ± 0.01	21.1 ± 0.01
Genbass	98.6 ± 0.17	98.6 ± 0.17	**98.7 ± 0.01**	98.6 ± 0.01	98.6 ± 0.01	98.6 ± 0.01	98.3 ± 0.01
Average rank	4.5	4.05	**2.88**	3.83	3.38	4.22	5.11

Table 4. Average and standard deviation of subset accuracy (exact match) of rival algorithms. Best results are shown boldface.

Dataset	BR	BRP	BRH	CC [16]	BCC [17]	CT [18]	LP [8]
Yeast	6.8 ± 0.02	7.6 ± 0.02	9.1 ± 0.01	**14.2 ± 0.02**	5.1 ± 0.02	7.2 ± 0.02	13.4 ± 0.02
Scene	42.7 ± 0.03	42.7 ± 0.03	47.9 ± 0.03	52.7 ± 0.03	45.8 ± 0.02	44.8 ± 0.03	**54.8 ± 0.03**
Birds	48.5 ± 0.05	48.5 ± 0.05	48.2 ± 0.06	47.9 ± 0.06	49.0 ± 0.06	48.4 ± 0.06	**49.0 ± 0.07**
Music	23.8 ± 0.06	23.8 ± 0.06	**32.1 ± 0.04**	24.9 ± 0.05	24.2 ± 0.07	24.2 ± 0.05	27.4 ± 0.06
Flags	13.9 ± 0.09	13.9 ± 0.09	17.1 ± 0.08	16.0 ± 0.1	15.0 ± 0.07	12.9 ± 0.08	**26.9 ± 0.09**
Emotions	24.0 ± 0.06	24.0 ± 0.06	**31.0 ± 0.05**	24.6 ± 0.06	24.6 ± 0.055	24.5 ± 0.06	28.8 ± 0.09
tmc2007	35.3 ± 0.01	35.5 ± 0.01	33.7 ± 0.01	37.2 ± 0.01	35.3 ± 0.01	35.7 ± 0.01	**38.4 ± 0.01**
CAL500	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Genbass	97.1 ± 0.36	97.1 ± 0.36	**97.3 ± 0.02**	97.1 ± 0.02	97.1 ± 0.02	97.1 ± 0.02	97.3 ± 0.02
Average rank	5.44	5.05	3.16	3.38	4.33	4.83	**1.77**

Table 5 shows the average precision of the rival algorithms on different datasets. It can be seen that the proposed method has achieved the best average rank among its rivals. Comparing the average ranks of table 5 with the Friedman test, we obtained X_F^2 = 17.39 and F_F = 3.8 with critical value 2.3 at the 0.05 critical level, and so we could reject the null hypothesis, which means that there is a significant difference among the rival algorithms. The result of the post-hoc Nemenyi test with critical distance CD = 3.00 at the 0.05 critical level is that the proposed BRH algorithm is significantly better than the BCC algorithm.

Table 6 shows the F1-micro average of the rival algorithms on different datasets. It can be seen that the proposed method has achieved the second best average rank among its rivals. Comparing the average ranks of table 6 with the Friedman test, we obtained X_F^2 = 16.36 and F_F = 3.48 with critical value 2.3 at the 0.05 critical level, and so we could reject the null hypothesis, which means that there is a significant difference among the rival algorithms. The result of the post-hoc Nemenyi test with critical distance CD = 3.00 at the 0.05 critical level is that the proposed BRH algorithm along with the BRP and BCC algorithms are significantly better than the LP algorithm.

Table 7 shows the hamming loss of the rival algorithms on different datasets. It can be seen that the proposed method has achieved the best average rank among its rivals. Comparing the average ranks of table 7 with the Friedman test, we obtained X_F^2 = 24.69 and FF = 6.74 with critical value 2.3 at the 0.05 critical level, and so we could reject the null hypothesis.

The result of the post-hoc Nemenyi test with critical distance CD = 3.00 at the 0.05 critical level is that the proposed BRH algorithm along with the BR and BRP algorithms are significantly better than the LP algorithm.

Table 5. Average and standard deviation of average precision of rival algorithms. Best results are shown boldface.

Dataset	BR	BRP	BRH	CC [16]	BCC [17]	CT [18]	LP [8]
Yeast	40.2 ± 0.01	38.7 ± 0.01	38.2 ± 0.01	40.0 ± 0.01	38.5 ± 0.01	39.0 ± 0.01	40.3 ± 0.01
Scene	38.3 ± 0.03	38.3 ± 0.03	**42.1 ± 0.03**	35.4 ± 0.02	34.4 ± 0.01	34.4 ± 0.02	35.2 ± 0.01
Birds	57.8 ± 0.05	57.8 ± 0.05	**58.5 ± 0.05**	56.5 ± 0.05	56.6 ± 0.05	56.7 ± 0.05	56.2 ± 0.05
Music	49.0 ± 0.04	49.0 ± 0.04	**50.1 ± 0.03**	40.0 ± 0.03	39.9 ± 0.02	40.5 ± 0.03	43.1 ± 0.03
Flags	59.5 ± 0.05	60.7 ± 0.02	**61.4 ± 0.04**	54.5 ± 0.05	54.2 ± 0.03	54.6 ± 0.05	53.8 ± 0.05
Emotions	49.7 ± 0.04	49.7 ± 0.04	**49.8 ± 0.03**	40.7 ± 0.03	40.0 ± 0.02	40.0 ± .003	42.7 ± 0.03
tmc2007	**20.6 ± 0.02**	20.6 ± 0.02	19.5 ± 0.02	17.9 ± 0.00	15.0 ± 0.00	15.0 ± 0.00	15.1 ± 0.00
CAL500	16.5 ± 0.01	16.5 ± 0.01	16.2 ± 0.01	18.9 ± 0.01	**19.3 ± 0.01**	19.3 ± 0.01	18.7 ± 0.01
Genbass	9.3 ± 0.01	9.3 ± 0.01	**10.1 ± 0.02**	7.9 ± 0.01	7.9 ± 0.01	7.9 ± 0.01	7.9 ± 0.01
Average rank	2.72	2.94	**2.55**	4.61	5.61	4.83	4.72

Table 6. Average and standard deviation of F1-micro of rival algorithms. Best results are shown boldface.

Dataset	BR	BRP	BRH	CC [16]	BCC [17]	CT [18]	LP [8]
Yeast	58.6 ± 0.02	60.7 ± 0.02	**61.1 ± 0.05**	55.6 ± 0.02	56.7 ± 0.02	56.7 ± 0.02	53.7 ± 0.02
Scene	61.9 ± 0.02	**61.9 ± 0.02**	60.0 ± 0.03	59.7 ± 0.03	61.2 ± 0.02	61.9 ± 0.02	59.7 ± 0.03
Birds	43.9 ± 0.06	43.9 ± 0.06	42.8 ± 0.06	43.4 ± 0.07	**44.9 ± 0.07**	43.4 ± 0.06	43.4 ± 0.07
Music	66.7 ± 0.04	66.7 ± 0.04	**67.8 ± 0.04**	66.9 ± 0.04	66.9 ± 0.05	66.8 ± 0.04	61.3 ± 0.04
Flags	73.6 ± 0.04	73.9 ± 0.04	**75.0 ± 0.04**	73.5 ± 0.05	73.7 ± 0.03	72.9 ± 0.04	71.9 ± 0.05
Emotions	67.3 ± 0.03	67.3 ± 0.03	**67.4 ± 0.03**	66.8 ± 0.02	67.2 ± 0.03	66.8 ± 0.03	62.1 ± 0.05
tmc2007	**71.6 ± 0.01**	71.6 ± 0.01	70.4 ± 0.01	70.8 ± 0.01	71.4 ± 0.01	71.4 ± 0.01	63.8 ± 0.00
CAL500	34.0 ± 0.01	34.0 ± 0.01	34.0 ± 0.02	**35.1 ± 0.02**	35.1 ± 0.02	35.1 ± 0.01	34.2 ± 0.02
Genbass	98.8 ± 0.12	98.8 ± 0.12	**98.9 ± 0.01**	98.8 ± 0.01	98.8 ± 0.01	98.8 ± 0.01	98.0 ± 0.02
Average rank	3.44	**3.11**	3.22	4.61	3.16	4.05	6.39

Figure 1 depicts the effect of the proposed method on the BR performance regarding the hamming loss measure. It can be seen that the proposed method has always improved the performance of the BR method except for the Scene dataset.

We analyzed the Scene dataset and its extracted rules, and figured out that the BR method was unable to assign any labels to 23% of the instances, and hence, no label was predicted for 23% of the instances, which is known as the "empty prediction" issue [32] in the multi-label classification context. Formula 8 shows one of the frequent patterns found in this 6-class dataset, which causes addition of class label 4 to all of the instances with empty prediction. As the empty prediction issue had occurred mostly on the instances that did not have class label 4, the hamming loss for BRH method was reduced in comparison with BR.

$$: y_1 : y_2 : y_3 : y_5 : y_6 \rightarrow y_4 \quad (8)$$

Figure 2 depicts the effect of our proposed method on the performance of the BR algorithm regarding the subset accuracy, accuracy, average precision, and F1-micro. It can be seen that although the proposed method has improved the BR and BRP performance most of the time, on the Birds and Tmc datasets, it sometimes has no effect or even it has worsen the results. It is worth mentioning that the proposed method can improve the classification results if useful frequent patterns are found between the labels.

Table 7. Average and standard deviation of Hamming loss of rival algorithms. Best results are shown boldface.

Dataset	BR	BRP	BRH	CC [16]	BCC [17]	CT [18]	LP [8]
Yeast	24.5 ± 0.01	23.9 ± 0.01	**22.7 ± 0.01**	26.6 ± 0.01	26.0 ± 0.01	25.8 ± 0.01	27.9 ± 0.01
Scene	**13.7 ± 0.01**	13.7 ± 0.01	15.2 ± 0.01	14.6 ± 0.01	13.8 ± 0.01	14.6 ± 0.01	14.3 ± 0.01
Birds	**4.9 ± 0.01**	**4.9 ± 0.01**	**4.9 ± 0.01**	**4.9 ± 0.01**	**4.9 ± 0.01**	5.0 ± 0.01	5.6 ± 0.01
Music	23.1 ± 0.03	23.1 ± 0.03	**20.9 ± 0.02**	23.1 ± 0.02	23.1 ± 0.01	23.0 ± 0.02	24.4 ± 0.03
Flags	25.4 ± 0.04	25.2 ± 0.04	**24.6 ± 0.04**	26.0 ± 0.04	25.6 ± 0.03	25.9 ± 0.04	27.1 ± 0.05
Emotions	22.7 ± 0.03	22.7 ± 0.03	**21.1 ± 0.02**	22.9 ± 0.03	22.8 ± 0.02	22.9 ± 0.03	23.9 ± 0.02
tmc2007	**5.5 ± 0.00**	**5.5 ± 0.00**	**5.5 ± 0.00**	5.6 ± 0.00	**5.5 ± 0.00**	**5.5 ± 0.00**	7.1 ± 0.00
CAL500	**16.1 ± 0.00**	**16.1 ± 0.00**	**16.1 ± 0.00**	17.4 ± 0.01	16.5 ± 0.01	16.5 ± 0.00	19.8 ± 0.01
Genbass	0.1 ± 0.01	0.1 ± 0.01	0.1 ± 0.00	**0.0 ± 0.00**	0.1 ± 0.00	0.1 ± 0.00	0.2 ± 0.00
Average rank	2.94	2.72	**2.55**	4.83	3.89	4.39	6.67

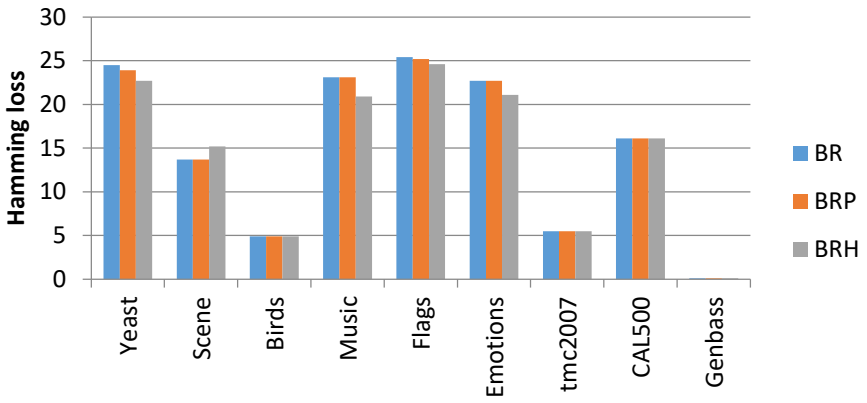

Figure 1. Effect of using positive and hybrid association rules on BR performance measured by hamming loss. Smaller values represent a better performance.

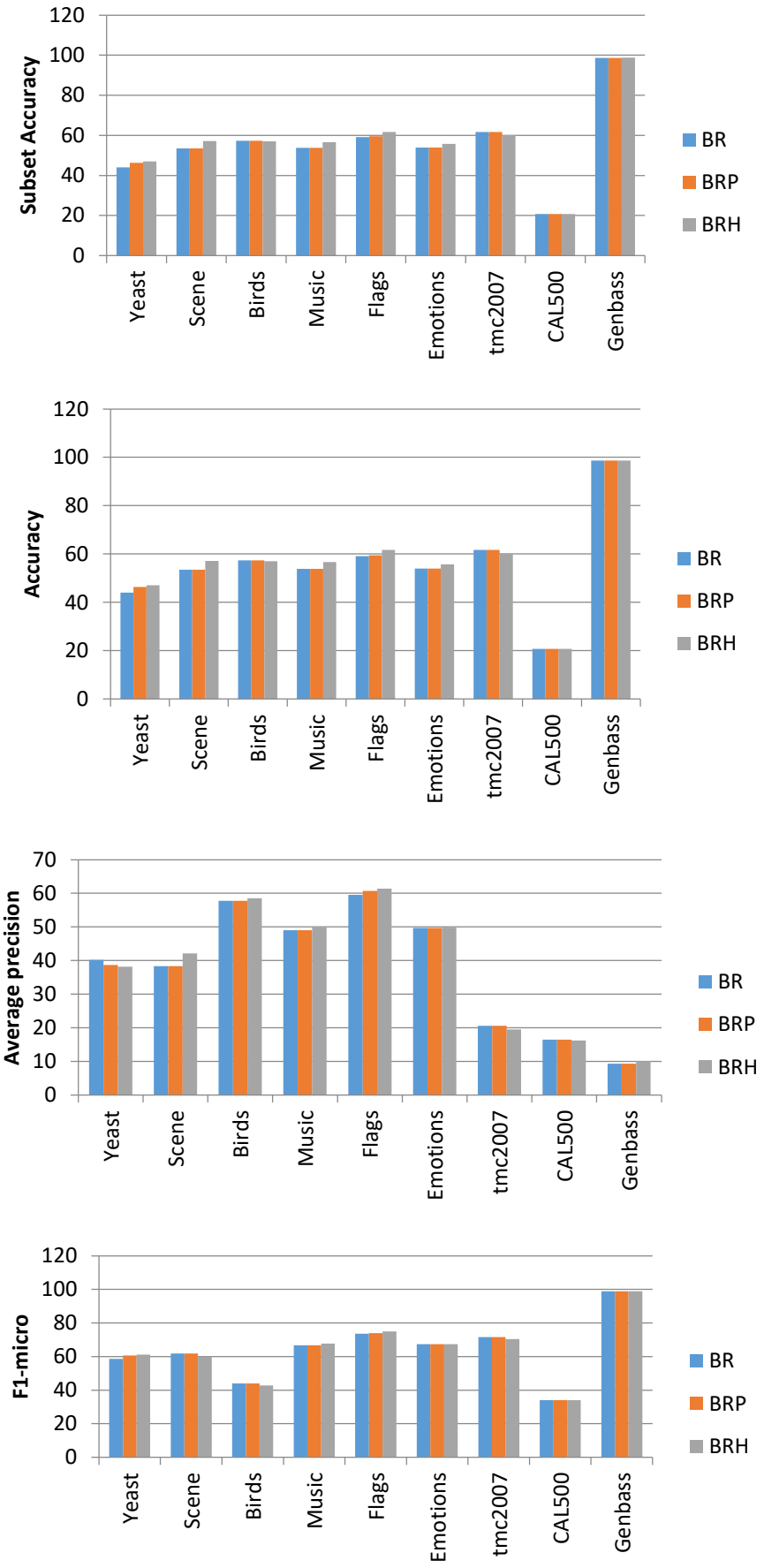

Figure 2. Effect of using positive and hybrid association rules on BR performance measured by subset accuracy, accuracy, average precision, and F1-micro. Bigger values represent a better performance.

For example, if the labels with high classification accuracy appear on the left side of the rule and the labels with low classification accuracy appear on the right hand side of it, then the performance will improve significantly. Overall, the significance of the improvement in the classification results by employing the proposed method highly depends upon the dataset, the correlations between the class labels, and the quality of the extracted frequent patterns.

7. Conclusion

In this work, we focused on using the correlations between the class labels in multi-label classification problems in order to improve the classification results. We presented the novel idea of contributing the labels that were not assigned to the instances along with the assigned labels in the process of extraction of the relationships between the class labels. We defined positive, negative, and hybrid relationships between the class labels in the multi-label classification context for the first time, and proposed a method for extracting such relationships for the multi-label classification problems. In addition, we proposed a post-processing method to revise and correct the predictions made by a multi-label classifier by employing the extracted relationships between the labels. We measured the performance of the proposed method by several metrics on several datasets, and compared it with several well-known multi-label classification algorithms. Our experimental results show that the proposed method has a strong ability in improving the multi-label classification results.

References

[1] Read, J., Pfahringer, B., Holmes, G. & Frank, E. (2011). Classifier chains for multi-label classification. Springer, Mach Learn vol. 85, pp. 333-359.

[2] Alazaidah, R., Thabtah, F. & Al-Raaidehi, Q. (2015). A Multi-Label Classification Approach Based on Correlations Among Labels. International Journal of Advanced Computer Science and Applications, vol 6, no 2, pp. 52-59.

[3] Patel, K., Kapadia, N. & Parikh, M. (2014). Discover Multi-label Classification using Association Rule Mining. International journal of Advance Engineering and Research Development, vol. 1, Issue. 1, ISSN: 2348-4470.

[4] Levati, J., Kocev, D. & Deroski, S. (2015). The importance of the label hierarchy in hierarchical multi-label classification. Journal of Intelligent Information Systems, vol. 45, no. 2, pp. 247-271.

[5] Wang, S., Wang, J., Wang, Z. & Ji, Q. (2014). Enhancing multi-label classification by modeling dependencies among labels. Elsevier, Pattern Recognition vol. 47, pp. 3405-3413.

[6] Alvares, C. E, Carolina, M. M. & Metz, J. (2011). Multi-label Problem Transformation Methods: a Case Study. CLEI ELECTRONIC JOURNAL, vol. 14, pp. 4-14.

[7] Zhang, M. & Zhou, Z. (2013). A Review on Multi-Label Learning Algorithms. IEEE Transactions on Knowledge and Data Engineering, vol. 26, issue. 8.

[8] Menca, E.L. & Furnkranz, J. (2008). Pairwise Learning of Multilabel Classifications with Perceptrons. IEEE 978-1-4244-1821-3/08.

[9] Read, J., Pfahringer, B. & Holmes, G. (2008). Multi-label Classification using Ensembles of Pruned Sets. 8th IEEE International Conference on Data Mining, 2008.

[10] Read, J. & Perez-Cruz, F. (2014). Deep Learning for Multi-label Classification. arXiv: 1502.05988v1 [cs.LG].

[11] Haripriya, H. Prathibhamol, Cp., Yashwant, R. M. S., Sandeep, A. M., Sankar, S. N. (2016). Multi Label Prediction Using Association Rule Generation and Simple k-Means. International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), 2016.

[12] Ramírez-Corona, M., Sucar, L.E., & Morales, E.F. (2016). Hierarchical multilabel classification based on path evaluation. International Journal of Approximate Reasoning, vol. 68, pp. 179-193.

[13] Zhang, M. & Zhou, Z. (2013). ML-KNN:Alazy learning approach to multi-label learning. ELSEVIER, Pattern Recognition vol. 40, pp. 2038 2048.

[14] Vens, C., Struyf, J., Schietgat, L., Deroski, S. & Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. Springer, Mach Learn vol. 73, pp. 185214.

[15] Hashemi, S., Yang, Y., Mirzamomen, Z. & Kangavari, M. (2009). Adapted one-versus-all decision trees for data stream classification. IEEE Trans Knowledge and Data Engineering, vol. 21, no. 5, pp. 624-637.

[16] Read, J., Pfahringer, B., Holmes, G. & Frank, E. (2009). Classifier Chains for Multi-label Classification. Springer-Verlag Berlin Heidelberg, ECML PKDD, Part II, LNAI 5782, pp. 254-269.

[17] Sucar, E.L., Bielza, C., Morales, E.F., Hernandez-Leal, P., Zaragoza, J.H. & Larraaga, P. (2014). Mult-label classification with Bayesian network-based chain Classifiers. ELSEVIER, Pattern Recognition Letters vol. 41, pp. 1422.

[18] Read J., Martino L., Olmos P. M. & Luengo David. (2015). Scalable Multi-output Label Prediction: from Classifier Chains to Classifier Trellises, arXiv: 1501.04870v1 [stat.ML].

[19] Charte, F., Rivera, A., Jesus, M.J. & Herrera, F. (2012). Improving Multi-label classifiers via Label Reduction with Association Rules. Springer-Verlag Berlin Heidelberg, HAIS part II, LNCS 7209, pp. 188-199.

[20] Cerri, R., Barros, R.C. & de Carvalho, A. (2014). Hierarchical multi-label classification using local neural networks. Journal of Computer and System Sciences, vol. 80, no. 1, pp. 39-56.

[21] Han, J., Pei, J. & Kamber, M. (2012). Data Mining concepts and techniques. Elsevier. Third Edition Book.

[22] Agrawal, R. & Srikant, R. (1998). Fast Algorithms for mining Association Rules in Large Databases. 20th International Conference on Very Large Data Bases, pp. 478-499.

[23] Han, J., Pei, J. & Yin, Y. (2000). Mining frequent patterns without candidate generation. Proceeding of the 2000 ACM-SIGMID International Conference on Management of Data, pp. 1-12.

[24] Friedman, M. (1937). The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. J Am Stat Assoc, vol. 32, pp. 675-701.

[25] Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics, vol. 11, no. 1, pp. 86-92.

[26] Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res, vol. 7, pp. 1-30.

[27] Waikato Environment for Knowledge Analysis Version 3.6.10 (1999-2013), Available: www.cs.waikato.ac.nz/ ml/weka.

[28] A Multi-label Extension to WEKA Version 1.7.7 (2012-2015), Available: www.meka.sourceforge.net.

[29] Quinlan, R.J. (1993), C4.5: Programs for Machine Learning. vol. 1, Morgan Kaufmann, San Mateo.

[30] Kajdanowicz, T. & Kazienko, P. (2013). Heuristic Classifier Chains for Multi-label Classification. Springer-Verlag Berlin Heidelberg. FQAS, LNAI 8132, pp. 555-566.

[31] Dembczynski, K., Cheng, W. & Hullermeier, E. (2010). Bayes Optimal Multi-label Classification via Probabilistic Classifier Chains, 27-th International Conference on Machine Learning, Haifa, 2010.

[32] Liu, S. & Chen, J. (2015). An empirical study of empty prediction of multi-label classification. Expert Systems with Applications, vol. 42, no. 13, pp. 5567-5579.

نشریه هوش مصنوعی و داده کاوی

بهره‌برداری از ارتباطات بین برچسب‌ها در رده‌بندی چندبرچسبی

زهرا میرزامومن* و خلیل غفوری‌پور

دانشکده مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران.

ارسال ۲۰۱۷/۰۳/۰۸؛ بازنگری ۲۰۱۷/۱۱/۲۵؛ پذیرش ۲۰۱۸/۰۱/۰۶

چکیده:

رده‌بندی چندبرچسبی، که در آن می‌توان به‌صورت همزمان چند برچسب کلاس به یک نمونه نسبت داد، کاربردهای زیادی در دسته‌بندی متون، بیولوژی و تشخیص پزشکی دارد. از آنجا که در اغلب مسائل، روابطی بین برچسب‌ها وجود دارد، استخراج روابط موجود بین برچسب‌ها و بهره‌برداری از آن‌ها در طول فاز آموزش یا در مرحله انجام پیش‌بینی، می‌تواند بهبود عمده‌ای در رده‌بندی به‌همراه داشته باشد. در این مقاله، ما مفهوم ارتباط مثبت، ارتباط منفی و ارتباط ترکیبی بین برچسب‌ها را برای اولین بار معرفی کرده‌ایم و روشی برای استخراج این‌گونه روابط در مسائل رده‌بندی چندبرچسبی پیشنهاد کرده‌ایم. همچنین، روشی برای استفاده از ارتباطات استخراج‌شده برای بهبود پیش‌بینی‌های انجام‌شده توسط رده‌بند چندبرچسبی ارائه کرده‌ایم. در این مقاله، آزمایش‌های گسترده‌ای برای ارزیابی کارایی روش پیشنهادی انجام شده‌است. نتایج بدست‌آمده، توانایی روش پیشنهادی در بهبود نتایج رده‌بندی چندبرچسبی را نشان می‌دهد.

کلمات کلیدی: رده‌بندی، رده‌بندی چندبرچسبی، ارتباطات بین برچسب‌ها، قانون انجمنی، ارتباط مثبت، ارتباط منفی، ارتباط ترکیبی.