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.
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 $k$th 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:

(laptop → wireless mouse)

[support: 20%, 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\footnote{Frequent Pattern Growth} [23]. FP-Growth algorithm is the extension of the Apriori, in which the computational cost is reduced by eliminating the need for passing
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.

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 \( \sim \) sign appears beside the labels that do not belong to the training instance.

\[
\begin{align*}
L_1L_2...L_q = & \begin{cases} 
L_i = \sim y_i & \text{if } y_i \notin Y(x) \\
L_i = y_i & \text{if } y_i \in Y(x)
\end{cases}
\end{align*}
\]

(1)

For example, in a problem with \( y = \{ y_1, y_2, ..., y_6 \} \), if we had \( Y(x) = \{ y_1, y_2, y_3 \} \) 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:

\[
y_1y_2 \rightarrow y_3
\]

which can be interpreted as follows: if an instance belongs to the \( y_1 \) and \( y_3 \) 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 \( Y(x) \cup Y'(x) \) 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:

\[
y_3 \rightarrow \sim y_6
\]

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 \( y_2 \) class, 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.

```plaintext
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 \( Y(x) \cup Y'(x) \) according to Eq. (1);
12:     end for
13:     Set the minimum support and minimum confidence for FP_Growth algorithm.
14:     \( R^+ \leftarrow \text{FP}_\text{Growth} (P_{itemsets}) \);
15:     \( R^- \leftarrow \text{FP}_\text{Growth} (N_{itemsets}) \);
16:     \( R^c \leftarrow \text{FP}_\text{Growth} (H_{itemsets}) \);
17:     return \( R^+ \), \( R^- \), \( R^c \);
18: end function
```

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

```plaintext
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:

\[ r_i \in R^+: y_p \rightarrow y_k \]

\[ r_s \in R^+ : y_n \rightarrow y_m \]

\[ r_j \in R^c : y_p^s \rightarrow y_r, y_s \]

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:

- \( h(x) = h(x) \cup y_s \) based on \( r_i \).
- \( h(x) = h(x) \setminus y_m \) based on \( r_s \).
- \( h(x) = h(x) + y_s \) based on \( r_j \).

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:

- **BR**: 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>
<thead>
<tr>
<th>Name</th>
<th>Cate</th>
<th>No.</th>
<th>Labels</th>
<th>Cardinal</th>
<th>Density</th>
<th>Distinct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast</td>
<td>biology</td>
<td>2417</td>
<td>14</td>
<td>4.237</td>
<td>0.303</td>
<td>198</td>
</tr>
<tr>
<td>Scene</td>
<td>image</td>
<td>2406</td>
<td>7</td>
<td>1.014</td>
<td>0.0352</td>
<td>133</td>
</tr>
<tr>
<td>Birds</td>
<td>voice</td>
<td>645</td>
<td>19</td>
<td>1.014</td>
<td>0.179</td>
<td>15</td>
</tr>
<tr>
<td>Music</td>
<td>voice</td>
<td>592</td>
<td>6</td>
<td>1.869</td>
<td>0.311</td>
<td>27</td>
</tr>
<tr>
<td>Flags</td>
<td>image</td>
<td>194</td>
<td>7</td>
<td>3.392</td>
<td>0.485</td>
<td>54</td>
</tr>
<tr>
<td>Emotions</td>
<td>voice</td>
<td>593</td>
<td>7</td>
<td>1.869</td>
<td>0.311</td>
<td>27</td>
</tr>
<tr>
<td>Genbass</td>
<td>biology</td>
<td>662</td>
<td>27</td>
<td>1.252</td>
<td>0.046</td>
<td>32</td>
</tr>
<tr>
<td>Cal500</td>
<td>voice</td>
<td>502</td>
<td>174</td>
<td>26.044</td>
<td>0.15</td>
<td>502</td>
</tr>
<tr>
<td>tmc2007</td>
<td>text</td>
<td>38956</td>
<td>22</td>
<td>2.158</td>
<td>0.098</td>
<td>1341</td>
</tr>
</tbody>
</table>

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.\[
Subset Accuracy = \frac{1}{p} \sum_{i=1}^{p} \left[h(x_i) = Y(x_i)\right]
\]

- **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.

\[ \text{Accuracy exam: } \frac{1}{p} \sum_{j=1}^{p} \left[ \frac{h_{ij} \cap Y_{ij} \cap Y_{ij}^c}{h_{ij} \cup Y_{ij} \cup Y_{ij}^c} \right] \]  

(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.

\[ \text{Average Precision: } \frac{1}{p} \sum_{j=1}^{p} \left[ \frac{1}{2} \sum_{i=1}^{q} \frac{y^i | \text{rank}_i(x_j, y^i) \leq \text{rank}_i(x_j, y^j) | y^j \notin Y_j}{\text{rank}_i(x_j, y^i)} \right] \]  

(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 \( h(x_i) \) is one if the labels \( y^j \) predicted for the instance \( x_i \), and zero otherwise. This metric should be maximized.

\[ \text{F1-micro: } \frac{2 \sum_{j=1}^{q} \sum_{i=1}^{p} y^j | \text{rank}_i(x_j, y^j) \leq \text{rank}_i(x_j, y^i) | y^i \notin Y_j}{\sum_{j=1}^{q} \sum_{i=1}^{p} y^j | \text{rank}_i(x_j, y^j) + \sum_{i=1}^{p} h(x_i) } \]  

(5)

Hamming loss: Equation (6) shows the hamming loss metric, in which \( \Delta \) 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.

\[ \text{Hamming loss: } \frac{1}{p} \sum_{i=0}^{p} \sum_{j=1}^{p} |h(x_i) \cap Y | \]  

(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 \( X^2 = 19.31 \) and \( F = 4.45 \) 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 \( X^2 = 6.12 \) and \( F = 1.02 \) 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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast</td>
<td>44.0 ± 0.02</td>
<td>46.3 ± 0.02</td>
<td>47.0 ± 0.02</td>
<td>42.9 ± 0.02</td>
<td>41.9 ± 0.02</td>
<td>42.1 ± 0.02</td>
<td>41.1 ± 0.02</td>
</tr>
<tr>
<td>Scene</td>
<td>53.5 ± 0.03</td>
<td>53.5 ± 0.03</td>
<td>57.1 ± 0.03</td>
<td>54.8 ± 0.03</td>
<td>58.2 ± 0.02</td>
<td>54.7 ± 0.03</td>
<td>58.8 ± 0.03</td>
</tr>
<tr>
<td>Birds</td>
<td>57.3 ± 0.05</td>
<td>57.3 ± 0.05</td>
<td>57.0 ± 0.05</td>
<td>56.3 ± 0.06</td>
<td>57.5 ± 0.05</td>
<td>57.1 ± 0.05</td>
<td>57.9 ± 0.07</td>
</tr>
<tr>
<td>Music</td>
<td>53.8 ± 0.05</td>
<td>53.8 ± 0.05</td>
<td>56.6 ± 0.04</td>
<td>54.3 ± 0.05</td>
<td>54.1 ± 0.06</td>
<td>54.0 ± 0.04</td>
<td>51.1 ± 0.05</td>
</tr>
<tr>
<td>Flags</td>
<td>59.1 ± 0.05</td>
<td>59.5 ± 0.05</td>
<td>61.7 ± 0.05</td>
<td>59.7 ± 0.06</td>
<td>59.3 ± 0.04</td>
<td>57.5 ± 0.05</td>
<td>59.2 ± 0.06</td>
</tr>
<tr>
<td>Emotions</td>
<td>53.9 ± 0.03</td>
<td>53.9 ± 0.03</td>
<td>55.7 ± 0.03</td>
<td>53.8 ± 0.03</td>
<td>54.1 ± 0.03</td>
<td>54.1 ± 0.04</td>
<td>52.3 ± 0.06</td>
</tr>
<tr>
<td>tmc2007</td>
<td>61.7 ± 0.01</td>
<td>61.7 ± 0.01</td>
<td>60.1 ± 0.01</td>
<td>61.5 ± 0.01</td>
<td>61.5 ± 0.01</td>
<td>61.6 ± 0.01</td>
<td>57.3 ± 0.01</td>
</tr>
<tr>
<td>CAL500</td>
<td>20.7 ± 0.01</td>
<td>20.7 ± 0.01</td>
<td>20.7 ± 0.02</td>
<td>21.7 ± 0.02</td>
<td>21.5 ± 0.01</td>
<td>21.5 ± 0.01</td>
<td>21.1 ± 0.01</td>
</tr>
<tr>
<td>Genbass</td>
<td>98.6 ± 0.17</td>
<td>98.6 ± 0.17</td>
<td>98.7 ± 0.01</td>
<td>98.6 ± 0.01</td>
<td>98.6 ± 0.01</td>
<td>98.6 ± 0.01</td>
<td>98.6 ± 0.01</td>
</tr>
<tr>
<td>Average rank</td>
<td>4.5</td>
<td>4.05</td>
<td>2.88</td>
<td>3.83</td>
<td>3.38</td>
<td>4.22</td>
<td>5.11</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast</td>
<td>6.8 ± 0.02</td>
<td>7.6 ± 0.02</td>
<td>9.1 ± 0.01</td>
<td>14.2 ± 0.02</td>
<td>5.1 ± 0.02</td>
<td>7.2 ± 0.02</td>
<td>13.4 ± 0.02</td>
</tr>
<tr>
<td>Scene</td>
<td>42.7 ± 0.03</td>
<td>42.7 ± 0.03</td>
<td>47.9 ± 0.03</td>
<td>52.7 ± 0.03</td>
<td>45.8 ± 0.02</td>
<td>44.8 ± 0.03</td>
<td>54.8 ± 0.03</td>
</tr>
<tr>
<td>Birds</td>
<td>48.5 ± 0.05</td>
<td>48.5 ± 0.05</td>
<td>48.2 ± 0.06</td>
<td>47.9 ± 0.06</td>
<td>49.0 ± 0.06</td>
<td>48.4 ± 0.06</td>
<td>49.0 ± 0.07</td>
</tr>
<tr>
<td>Music</td>
<td>23.8 ± 0.06</td>
<td>23.8 ± 0.06</td>
<td>32.1 ± 0.04</td>
<td>24.9 ± 0.05</td>
<td>24.2 ± 0.07</td>
<td>24.2 ± 0.05</td>
<td>27.4 ± 0.06</td>
</tr>
<tr>
<td>Emotions</td>
<td>13.9 ± 0.09</td>
<td>13.9 ± 0.09</td>
<td>17.1 ± 0.08</td>
<td>16.0 ± 0.10</td>
<td>15.0 ± 0.07</td>
<td>12.9 ± 0.08</td>
<td>26.9 ± 0.09</td>
</tr>
<tr>
<td>tmc2007</td>
<td>34.0 ± 0.06</td>
<td>24.0 ± 0.06</td>
<td>31.0 ± 0.05</td>
<td>24.6 ± 0.06</td>
<td>24.6 ± 0.05</td>
<td>24.5 ± 0.06</td>
<td>28.8 ± 0.09</td>
</tr>
<tr>
<td>CAL500</td>
<td>0.0 ± 0.00</td>
<td>0.0 ± 0.00</td>
<td>0.0 ± 0.00</td>
<td>0.0 ± 0.00</td>
<td>0.0 ± 0.00</td>
<td>0.0 ± 0.00</td>
<td>0.0 ± 0.00</td>
</tr>
<tr>
<td>Genbass</td>
<td>97.1 ± 0.36</td>
<td>97.1 ± 0.36</td>
<td>97.3 ± 0.02</td>
<td>97.1 ± 0.02</td>
<td>97.1 ± 0.02</td>
<td>97.1 ± 0.02</td>
<td>97.3 ± 0.02</td>
</tr>
<tr>
<td>Average rank</td>
<td>5.44</td>
<td>5.05</td>
<td>3.16</td>
<td>3.38</td>
<td>4.33</td>
<td>4.83</td>
<td>1.77</td>
</tr>
</tbody>
</table>

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 $X^2 = 17.39$ and $F = 3.8$ 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 $X^2 = 16.36$ and $F = 3.48$ 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 $X^2 = 24.69$ 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.
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, which is known as the "empty prediction" issue [32] in the multi-instance 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.

\[ Y_1 : Y_2 : Y_3 : Y_5 : Y_6 \rightarrow Y_4 \]  

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.

![Figure 1. Effect of using positive and hybrid association rules on BR performance measured by hamming loss. Smaller values represent a better performance.](image)

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BR</th>
<th>BCC</th>
<th>BRH</th>
<th>CT</th>
<th>LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast</td>
<td>58.6 ± 0.02</td>
<td>60.7 ± 0.02</td>
<td><strong>61.1 ± 0.05</strong></td>
<td>55.6 ± 0.02</td>
<td>56.7 ± 0.02</td>
</tr>
<tr>
<td>Scene</td>
<td>61.9 ± 0.02</td>
<td>61.9 ± 0.02</td>
<td>60.0 ± 0.03</td>
<td>59.7 ± 0.03</td>
<td>61.2 ± 0.02</td>
</tr>
<tr>
<td>Birds</td>
<td>43.9 ± 0.06</td>
<td>43.9 ± 0.06</td>
<td>42.8 ± 0.06</td>
<td>43.4 ± 0.07</td>
<td><strong>44.9 ± 0.07</strong></td>
</tr>
<tr>
<td>Music</td>
<td>66.7 ± 0.04</td>
<td>66.7 ± 0.04</td>
<td><strong>67.8 ± 0.04</strong></td>
<td>66.9 ± 0.04</td>
<td>66.9 ± 0.05</td>
</tr>
<tr>
<td>Flags</td>
<td>73.6 ± 0.04</td>
<td>73.9 ± 0.04</td>
<td><strong>75.0 ± 0.04</strong></td>
<td>73.5 ± 0.05</td>
<td>73.7 ± 0.03</td>
</tr>
<tr>
<td>Emotions</td>
<td>67.3 ± 0.03</td>
<td>67.3 ± 0.03</td>
<td><strong>67.4 ± 0.03</strong></td>
<td>66.8 ± 0.02</td>
<td>67.2 ± 0.03</td>
</tr>
<tr>
<td>tmc2007</td>
<td><strong>71.6 ± 0.01</strong></td>
<td>71.6 ± 0.01</td>
<td>70.4 ± 0.01</td>
<td>70.8 ± 0.01</td>
<td>71.4 ± 0.01</td>
</tr>
<tr>
<td>CAL500</td>
<td>34.0 ± 0.01</td>
<td>34.0 ± 0.01</td>
<td>34.0 ± 0.02</td>
<td>35.4 ± 0.02</td>
<td><strong>35.4 ± 0.02</strong></td>
</tr>
<tr>
<td>Genbass</td>
<td>98.8 ± 0.12</td>
<td>98.8 ± 0.12</td>
<td><strong>98.9 ± 0.01</strong></td>
<td>98.8 ± 0.01</td>
<td>98.8 ± 0.01</td>
</tr>
<tr>
<td>Average rank</td>
<td>3.44</td>
<td>3.11</td>
<td>3.22</td>
<td>4.61</td>
<td>3.16</td>
</tr>
</tbody>
</table>

![Table 7. Average and standard deviation of Hamming loss of rival algorithms. Best results are shown boldface.](image)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BR</th>
<th>BRP</th>
<th>BCC</th>
<th>CT</th>
<th>LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast</td>
<td><strong>24.5 ± 0.01</strong></td>
<td>23.9 ± 0.01</td>
<td>22.7 ± 0.01</td>
<td>26.6 ± 0.01</td>
<td>26.0 ± 0.01</td>
</tr>
<tr>
<td>Scene</td>
<td><strong>13.7 ± 0.01</strong></td>
<td>13.7 ± 0.01</td>
<td>15.2 ± 0.01</td>
<td>14.6 ± 0.01</td>
<td>13.8 ± 0.01</td>
</tr>
<tr>
<td>Birds</td>
<td><strong>4.9 ± 0.01</strong></td>
<td>4.9 ± 0.01</td>
<td>4.9 ± 0.01</td>
<td>4.9 ± 0.01</td>
<td><strong>4.9 ± 0.01</strong></td>
</tr>
<tr>
<td>Music</td>
<td>23.1 ± 0.03</td>
<td>23.1 ± 0.03</td>
<td><strong>20.9 ± 0.02</strong></td>
<td>23.1 ± 0.02</td>
<td>23.1 ± 0.01</td>
</tr>
<tr>
<td>Flags</td>
<td>25.4 ± 0.04</td>
<td>25.2 ± 0.04</td>
<td><strong>24.6 ± 0.04</strong></td>
<td>26.0 ± 0.04</td>
<td>25.6 ± 0.03</td>
</tr>
<tr>
<td>Emotions</td>
<td>22.7 ± 0.03</td>
<td>22.7 ± 0.03</td>
<td><strong>21.1 ± 0.02</strong></td>
<td>22.9 ± 0.03</td>
<td>22.8 ± 0.02</td>
</tr>
<tr>
<td>tmc2007</td>
<td><strong>5.5 ± 0.00</strong></td>
<td>5.5 ± 0.00</td>
<td>5.5 ± 0.00</td>
<td>5.6 ± 0.00</td>
<td><strong>5.5 ± 0.00</strong></td>
</tr>
<tr>
<td>CAL500</td>
<td>16.1 ± 0.00</td>
<td>16.1 ± 0.00</td>
<td><strong>16.1 ± 0.00</strong></td>
<td>17.4 ± 0.01</td>
<td>16.5 ± 0.01</td>
</tr>
<tr>
<td>Genbass</td>
<td>0.1 ± 0.01</td>
<td>0.1 ± 0.01</td>
<td>0.1 ± 0.00</td>
<td><strong>0.0 ± 0.00</strong></td>
<td>0.1 ± 0.00</td>
</tr>
<tr>
<td>Average rank</td>
<td>2.94</td>
<td>2.72</td>
<td>2.55</td>
<td>4.83</td>
<td>3.89</td>
</tr>
</tbody>
</table>
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


بهره‌برداری از ارتباطات بین برچسب‌ها در رده‌بندی چندبرچسبی

زهراء میرزامومن* و خلیل غفوری پور
دانشکده مهندسی کامپیوتر، دانشگاه تربیت مدرس شهید رجایی، تهران، ایران.

ارسال 80/80/80/7802؛ بازنگری 72/00/7802؛ پذیرش 80/80/80/7800:

چکیده:
رده‌بندی چندبرچسبی، که در آن می‌توان به‌صورت همزمان چند برچسب کلاس به یک نمونه نسبت داد، کاربردهای زیادی در درست‌بندی مشورت، بیولوژی و تشخیص بیش رضا دارد. از آنجا که در اغلب مسائل، روابط بین برچسب‌ها وجود دارد، استخراج روابط موجود بین برچسب‌ها و بهره‌برداری از آنها در طول فاز آموزش یا در مرحله انجام پیش‌بینی، می‌تواند به‌عنوان ابزاری در رده‌بندی به‌عنوان داشته باشد. در این مقاله، مفهوم ارتباط منفی، ارتباط منفی و ارتباطات بین برچسب‌ها را برابر اولین بار معرفی کرده‌ایم و روشی برای استخراج این گونه روابط در مسائل رده‌بندی چندبرچسبی پیشنهاد کرده‌ایم. همچنین، روشی برای استفاده از ارتباطات استخراج‌شده بین برچسب‌ها به‌عنوان ابزاری در رده‌بندی چندبرچسبی پیشنهاد کرده‌ایم. در این مقاله، قانون گسترده‌ای برای ارزیابی کارایی روش پیش‌بینی رده‌بندی انجام شده‌است. نتایج بدست‌آمده تولید روش پیش‌بینی در بهبود نتایج رده‌بندی چندبرچسبی را نشان می‌دهد.

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