Improving Multi-label Classification Performance by Label Constraints

Benhui Chen, Xuefen Hong, Lihua Duan and Jinglu Hu

Abstract—Multi-label classification is an extension of traditional classification problem in which each instance is associated with a set of labels. For some multi-label classification tasks, labels are usually overlapped and correlated, and some implicit constraint rules are existed among the labels. This paper presents an improved multi-label classification method based on label ranking strategy and label constraints. Firstly, one-against-all decomposition technique is used to divide a multi-label classification task into multiple independent binary classification sub-problems. One binary SVM classifier is trained for each label. Secondly, based on training data, label constraint rules are mined by association rule learning method. Thirdly, a correction model based on label constraints is used to correct the probabilistic outputs of SVM classifiers for label ranking. Experiment results on three well-known multi-label benchmark datasets show that the proposed method outperforms some conventional multi-label classification methods.

I. INTRODUCTION

Nowadays, multi-label classification methods are increasingly required by applications, such as protein function classification [1], text categorization [2], music categorization [3] and semantic scene classification [4]. Multi-label classification problem is an extension of traditional multi-class classification problem in which its classes are not mutually exclusive and each sample may belong to several classes simultaneously. In multi-label classification, samples in training data are associated with a set of labels and the task is to predict the label set for each unseen instances, the main challenge of this problem is that classes are usually overlapped and correlated.

The complexity of multi-label problems inevitably makes them more difficult to solve. A straightforward approach to solve a multi-label problem is to decompose it into multiple independent binary classification problems, each for one label. In the Binary Relevance (BR) approach, one binary classifier is trained for each possible label, in which all training examples for which the label is relevant are used as positives examples and all other examples as negative examples. However, BR approach does not consider the correlations between the different labels of each instance. It neglects the fact that information of one label may be helpful for the learning of another related label; especially when some labels have insufficient training examples, the label correlations may provide helpful extra information. Thus, the exploitation of label correlations has been widely accepted as a key component of current multi-label learning approaches [5].

In the multi-labeled sets, labels are often overlapped and correlated. Some implicit constraint rules are existed among the labels. For example, the probability of an image being annotated with label *Africa* would be high if we know it has labels *lion* and *grassland*; a document is unlikely to be labeled as *politics* if we know it is related to *entertainment*. Thus, effective exploitation of correlation information among different labels is crucial for the success of any multi-label learning system [6]. This paper presents an improved multi-label classification method based on label ranking strategy and label constraints. Label ranking strategy is often used in multi-label classification methods [7], [1]. For each input instance, a scoring approach is required to evaluate real-valued scores for all labels (classes), and the obtained scores are utilized to sort a label rank which labels of higher scores are more related with the input instance. Then, a threshold is decided by a certain thresholding strategy, the labels of ranking scores larger than this threshold are predicted as the label subset of the input instance.

In the proposed method, firstly, one-against-all decomposition technique is used to divide a multi-label problem into binary class sub-problems. One binary Support Vector Machine (SVM) classifier is trained for each possible label. Secondly, based on training dataset, label constraint rules are mined by association rule learning method. Thirdly, a correction model based on label constraints is used to correct the probabilistic outputs of SVM classifiers for label ranking. Experimental performance evaluation on three well-known multi-label benchmark datasets show that the proposed method outperforms some existed methods.

The rest parts of the paper are organized as follows. Section II gives a brief overview of multi-label classification and related works. Section III describes the details of the proposed multi-label classification method based on label ranking strategy and label constraints. Section IV presents the multi-label benchmark datasets and experiment results. Finally, the conclusions are discussed.

II. NOTATION AND RELATED WORKS

A. Formal Framework for Multi-label Classification

A formal statement for multi-label classification problem is described as follows. We consider $X = R^d$ as the domain of samples and $Y = \{1, 2, \ldots, k\}$ as the finite set of labels. Given a multi-label training set $T$ of size $m$, $T =$
\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} (x_i \in X, y_i \subseteq Y)\) where each sample \(x_i \in X\) is associated with a subset of relevant labels \(y_i \subseteq Y\), the object of the multi-label classification is to build a multi-label classifier \(C : X \rightarrow 2^k (k = |Y| \geq 2)\) which, for any given new instance \(x_i\), predicts all its relevant label subset \(y_i\).

For the multi-label classification method based on label ranking, the learning model is expected to produce a real-valued scoring function of the form \(f(\cdot, \cdot) : X \times Y \rightarrow \mathbb{R}\). It is supposed that, a successful learning model will tend to output larger score values for labels in \(y_i\) than those not in \(y_i\). The corresponding multi-label classifier \(C(\cdot)\) can also be derived from the real-valued scoring function:

\[
C(x_i) = \{y_i | f(x_i, y) > t, y \in Y\}. \quad (1)
\]

where \(t\) is a value estimated by a certain thresholding strategy.

In order to measure the performance of multi-label classifiers, several metrics have been proposed. A unified presentation of existing evaluation measures for multi-label classification can be found in [5], including their categorization into example-based, label-based and ranking-based measures. We consider four metrics (Hamming loss, Accuracy, Precision and Recall) in this paper. They are described as follows [8], [1], [9].

Let \(D\) be a multi-label evaluation data set, consisting of \(|D|\) multi-label examples \((x_i, y_i), i = 1 \ldots |D|, y_i \subseteq Y\). \(Y = \{1, 2, \ldots, k\}\) is the set of labels. Let \(C\) be a multi-label classifier and \(\hat{y}_i = C(x_i)\) be the set of labels predicted by \(C\) for example \(x_i\).

\textbf{Hamming loss} evaluates how many times a sample label pair is misclassified, i.e., a label not belonging to the instance is predicted or a label belonging to the instance is not predicted, which is defined as:

\[
\text{Hamming Loss}(C, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{1}{k} |y_i \oplus \hat{y}_i| \quad (2)
\]

where \(\oplus\) means for the XOR operation of two sets. The smaller this metric value is, the better the classifier performs.

\textbf{Accuracy, Precision and Recall} evaluation metrics are defined as follows.

\[
\text{Accuracy}(C, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|y_i \cap \hat{y}_i|}{|y_i \cup \hat{y}_i|} \quad (3)
\]

\[
\text{Precision}(C, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|y_i \cap \hat{y}_i|}{|\hat{y}_i|} \quad (4)
\]

\[
\text{Recall}(C, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|y_i \cap \hat{y}_i|}{|y_i|}. \quad (5)
\]

There is a common characteristic for above three measures, the larger metric value, the better the classifier performance. The performance is perfect when its value equals 1.

\section*{B. Related Approaches for Multi-label Classification}

The most employed baseline method for multi-label classification is the Binary Relevance (BR) approach. BR approach decomposes the learning problem into a set of binary classification sub-problems, one per label, where each single model is learned independently of the rest, using only the information of that particular label and ignoring all other labels. Despite its simplicity, BR approach presents several advantages: 1) any binary learning model can be used as base learner, 2) it has linear complexity with respect to the number of labels and 3) it can be easily parallelized. The main drawback of BR is that it does not take into account any label correlations and constraints and may fail to predict some label combinations if such correlations are present.

In the multi-labeled sets, labels are often overlapped and correlated. Some implicit correlations and constraints are existed among the labels. Thus, the exploitation of label correlations and constraints has been widely accepted as a key component of current multi-label learning approaches [5].

There are many methods that try to find correlations between labels. Godbole and Sarawagi [10] present an approach to overcome the label independence problem of BR. They apply the stacked generalization learning paradigm, also known simply as stacking, in the context of multi-label classification. Read et al. present a learning algorithm called Classifier Chain (CC) [11], that can model label correlations while maintaining a computational complexity of the same order as that of BR. As its name denotes, CC involves a set of binary classifiers linked along a chain, where each classifier deals with the binary relevance problem associated with one label. In the training phase, the feature space of each classifier in the chain is extended with the actual label information of all previous links. However, in the testing phase, the classifiers are applied following the chain order, using the binary outputs of the previous models as additional input information. Tsoumakas and Vlahavas [12] present a method that iteratively constructs an ensemble of Label Powerset (LP) classifiers. LP algorithm considers each unique subset of labels that exists in a multi-label training set as one of the classes of a new multi-class classification task.

In order to exploit label correlations and dependencies, some methods resort to external knowledge such as existing label correlation matrices [13] or label hierarchies [14]. Many other methods try to exploit label correlations concealed in the training data. For example, the co-occurrence of labels in training data is utilized in [15]; Ghamrawi and McCallum [16] try to model the impact of an individual feature on the co-occurrence probability of label pairs; Sun, Ji and Ye [17] utilize a hypergraph to model the correlation information contained in different labels; in Ref. [6], a Bayesian network structure is used to encode the conditional dependencies of both the labels and feature set.
III. PROPOSED MULTI-LABEL CLASSIFICATION BASED ON LABEL CONSTRAINTS

A. Mining Label Constraints Locally From Training Data Subsets

Although different multi-label classification methods have tried to exploit different types of label constraints, they usually exploit label constraints in a global way by assuming that the constraints are shared by all instances [6], [18]. However, in our practice, we find that label constraints are naturally local in many applications. A label constraint may be shared by only a subset of instances rather than all the instances. Comparing with mining label constraints in label set globally, exploiting label constraints in label subsets locally can obtain better enhancement for classifier performance. In the proposed method, Affinity Propagation (AP) [19] clustering approach is used to partition label set of training data into several subsets firstly. Then, FP-Growth [20] association rule learning method is used to mine label constraint rules based on label subsets of training data.

The AP [19] is a clustering method which takes as input a set of measures of similarity between pairs of data points and outputs a set of clusters of the points with their corresponding exemplars. It has been praised because of its ability to efficiently and quickly handle very large problems. The algorithm takes a matrix of similarity measures between each pair of points $s(i, k)$ as input. Instead of requiring that the number of clusters be predetermined, the AP takes as input a real number $s(k; k)$ for each data point $k$. These values, which are called preferences, are a measure of how likely each point is to be chosen as exemplar. The algorithm works by exchanging messages between the points until a stop condition is satisfied. There are two types of messages to be exchanged between data points. The responsibility $r(i, k)$, sent from data point $i$ to candidate exemplar point $k$, reflects the accumulated evidence for how well-suited point $k$ is to serve as the exemplar for point $i$, taking into account other potential exemplars for point $i$. The availability $a(i, k)$, sent from candidate exemplar point $k$ to point $i$, reflects the accumulated evidence for how appropriate it would be for point $i$ to choose point $k$ as its exemplar, taking into account the support from other points that point $k$ should be an exemplar. The availabilities are initialized to zero: $a(i, k) = 0$. Then, the parameters are computed and updated using the rules as follows:

$$r(i, k) \leftarrow s(i, k) - \max_{k' | s(i, k') \neq k} \{a(i, k') + s(i, k')\} \quad (6)$$

$$a(i, k) \leftarrow \min\{0, r(k, i) + \sum_{i' | s(i', k) \neq (i, k)} \max\{0, r(i', k)\}\} \quad (7)$$

$$a(k, k) \leftarrow \sum_{i' | s(i', k) \neq (i, k)} \max\{0, r(i', k)\} \quad (8)$$

In the proposed method, label constraints are presented by association rules concealed in the label subsets of training data. Based on the label subsets of training data, the purpose of the association rules is to find correlations between the labels of instances. Knowing the associations between the labels, helps to correct the label ranking list in multi-label classification. In this paper, we only consider two types of constraints: positive and negative constraints. They are defined as follows.

**Positive constraints** \(\{l_{i1}, l_{i2}, \cdots, l_{ik}\} \triangleright l_j\) denote that if label set \(\{l_{i1}, l_{i2}, \cdots, l_{ik}\}\) is relevant for a given instance \(x\) then \(l_j\) has to be also relevant. Formally

$$\{l_{i1}, l_{i2}, \cdots, l_{ik}\} \triangleright l_j := l_{i1} \in Y^+ \land l_{i2} \in Y^+ \land \cdots \land l_{ik} \in Y^+ \rightarrow l_j \in Y^+ \quad (9)$$

**Negative constraints** \(l_{i1}, l_{i2}, \cdots, l_{ik}\) \(\triangleright \sim l_j\) denote that if label set \(\{l_{i1}, l_{i2}, \cdots, l_{ik}\}\) is not relevant for a given instance \(x\) then \(l_j\) has to be also not relevant. Formally

$$\{l_{i1}, l_{i2}, \cdots, l_{ik}\} \triangleright \sim l_j := l_{i1} \in Y^- \land l_{i2} \in Y^- \land \cdots \land l_{ik} \in Y^- \rightarrow l_j \in Y^- \quad (10)$$

FP-Growth [20] association rule learning method is used to mine above two label constraints based on label subsets of training data. FP-Growth is a efficient association rules learning approach, it works in a divide and conquer way. The initial phase of FP-Growth is the construction of a memory structure called FP-tree. FP-tree is a highly compact representation of the original database, which is assumed to fit into the main memory. FP-tree contains only frequent items, each transaction has a corresponding path in the tree, and transactions having a common prefix share the common starting fragment of their paths. The procedure of creating an FP-tree requires two database scans: one to discover frequent items and their counts, and second to build the tree by adding transactions to it one by one. After an FP-tree is built, the actual FP-Growth procedure is recursively applied to it, which discovers all frequent itemsets in a depth-first manner by exploring projections (conditional FP-trees) of the tree with respect to frequent prefixes found so far. It should be noted that after the FP-tree is created, the original database is not scanned anymore, and therefore the whole mining process requires exactly two database scans. [20].

B. Correction for label ranking list based on label constraints

The label constraint rules mined from the training data are used to enhance the label ranking multi-label classifier in the proposed method. Firstly, one-against-all decomposition technique is used to divide a multi-label problem into binary class sub-problems. A label ranking list is generated by combining the probabilistic outputs of each binary SVM classifier. Then, label constraint rules are learned by FP-Growth method. And a correction model based on label constraints is used to obtain a proper label ranking list by correcting the probabilistic outputs of SVM classifiers.

Constructing a classifier to produce a posterior probability $p$ (class-input) is very useful when a classifier is making a small part of an overall decision, and the classification output must be combined for the overall decision. The standard SVM do not provide such probabilities. The Platt’s sigmoid
method is commonly used to estimate the probabilistic outputs of SVM. It maps the SVM outputs into probabilities through training the parameters of an additional sigmoid function by validation set in training data [21].

\[
p(y = 1|x) = \frac{1}{1 + \exp(Af(x) + B)}
\]  

(11)

with parameters \(A\) and \(B\). To estimate the best values of \((A, B)\), any validation subset of \(m'\) training data can be used to solve the following maximum likelihood problem,

\[
\min_{Z=(A,B)} \left\{ -t \sum_{i=1}^{t} \left( t_i \log(p_i) + (1 - t_i) \log(1 - p_i) \right) \right\}
\]

\[
p_i = \frac{1}{1 + \exp(Af_i + B)} \quad f_i = f(x_i)
\]

\[t_i = \begin{cases} \frac{N_i + 1}{N_i + 2} & \text{if } y_i = 1 \\ \frac{N_i - 1}{N_i + 2} & \text{if } y_i = -1 \end{cases} \quad i = 1, 2, \cdots, m'
\]  

(12)

where \(N_+\) means the number of positive-labeled samples, and \(N_-\) means the number of negative-labeled samples. The SVM+Sigmoid combination preserves the sparseness of the SVM while producing probabilities that are of comparable quality to the regularized likelihood kernel method.

In order to obtain a proper label ranking list by incorporating label constraints described by association rules. Based on the association rules and probabilistic outputs of SVM classifiers, an ensemble strategy is used to combine the prediction probabilistic value of basis classifier with the constraint probabilistic values that come from the other classifiers corresponding with constraint labels. More precisely, given a test instance \(x\), basis SVM classifiers estimate local probabilities \(\overline{p}_j(x)\) that the instance \(x\) belongs to class \(l_j\). Consider a constraint rule \(\{l_{i1}, l_{i2}, \cdots, l_{ik}\} \triangleright l_j\) (or \(l_{i1}, l_{i2}, \cdots, l_{ik}\) \(\triangleright l_j\)) described in previous subsection, the ensemble strategy provides an estimate of the “consensus” probability \(p_j(x)\). The set \(\varphi_j(x)\) denotes the decision labels of \(l_j\) defined by the constraint rule:

\[
\varphi_j(x) = \{l_{i1}, l_{i2}, \cdots, l_{ik}\}
\]  

(13)

As described in Eq.(14), the consensus probability \(p_j(x)\) of the ensemble depends both on the local prediction \(\overline{p}_j(x)\) and on the predictions of the classes belonging to \(\varphi_j(x)\). To balance the local prediction \(\overline{p}_j(x)\) with the predictions coming from the classifiers corresponding to classes in \(\varphi_j(x)\), it introduces a weight parameter \(w\), \(0 \leq w \leq 1\), such that if \(w = 1\) the decision at label \(l_j\) depends only on the local predictor, otherwise the prediction is shared proportionally to \(w\) and \(1 - w\) between respectively the local predictor and the set of its decision labels in constraint rule.

\[
p_j(x) = w \cdot \overline{p}_j(x) + \frac{1 - w}{|\varphi_j(x)|} \sum_{i \in \varphi_j(x)} p_i(x).
\]  

(14)

C. Framework of the Proposed Method

Following a typical label ranking multi-label classification framework, the major steps of the proposed method can be summarized as follows.

1) The \(k\) (the number of labels) binary SVMs are built by one-against-all strategy. The Platt’s sigmoid method is used to obtain the probabilistic outputs of SVM classifiers. Thus, we can get \(k\) probabilistic classifiers.
2) Affinity Propagation (AP) clustering approach is used to partition label set of training data into several label subsets. Then, FP-Growth association rule learning method is used to mine label constraint rules based on label subsets of training data.
3) In test procedure, based on the association rules and probabilistic outputs of binary SVM classifiers, an ensemble strategy described in Eq.(14) is used to combine the local predictions of basis classifiers with the constraint decisions that come from the other classifiers. And the ensemble values are used as the final prediction results. Hence, we can get a probabilistic label ranking list of \(k\) values for each instance.
4) Finally, a threshold \(t\) is decided by a certain thresholding strategy, the labels of ranking scores larger than this threshold are predicted as the label subset of the input instance.

IV. EXPERIMENTS

A. Experiment DataSets

We implement experiments on three benchmark datasets: Yeast (a gene function dataset), genebase (a motif-based protein dataset) and Scene (a scene dataset) to evaluate the proposed method, which are downloaded from [22]. The detailed information about them, such as the number of samples, attributes, classes (labels) and their average number of labels are listed in Tab. I.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Total labels</th>
<th>Total attributes</th>
<th>Avg. labels</th>
<th>Training/Testing set size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast</td>
<td>14</td>
<td>103</td>
<td>4.25</td>
<td>1500/917</td>
</tr>
<tr>
<td>Genbase</td>
<td>27</td>
<td>1186</td>
<td>1.35</td>
<td>463/199</td>
</tr>
<tr>
<td>Scene</td>
<td>6</td>
<td>294</td>
<td>1.08</td>
<td>1211/1196</td>
</tr>
</tbody>
</table>

B. Experiment Results and Analysis

In our experiments, the SVM from LSSVM library [23] is taken as a basis. SVM parameters are chosen by cross-validation procedure. In the training procedure, the Platt’s sigmoid method is used to estimate the probabilistic outputs of SVM. For each sample in the testing data, a probabilistic rank of \(k\) values for all labels is obtained by final decision procedure. The threshold \(t\) is set as 0.5 for determining the label subset and its size for each testing sample in all experiments.

About the other parameters of the proposed method, the minimum support threshold parameters of FP-Growth are set as \(\text{minsup} = 0.4\). The weight parameter \(w\) of ensemble correction strategy in Eq.(14) is set as 0.65. For the parameters of AP clustering (defined in Ref. [19]), the maximum number of iterations is set as 1000, and early terminate parameter is
set as 100 (i.e., the clustering procedure will be terminated if the estimated exemplars stay fixed for continuous 100 iterations); the damping factor, which may be needed if oscillations occur, is set as 0.9; Hamming distance is used for the similarity metric of label vectors of training samples.

We compared our method with the following popular multi-label classification methods: C4.5, Naive Bayes and Binary-SVM [8], [9]. The experimental results are presented in the Tab.II, Tab.III and Tab.IV. The winning results are marked with bold font.

### TABLE II

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Hamming loss</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5</td>
<td>0.259</td>
<td>0.423</td>
<td>0.561</td>
<td>0.595</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.301</td>
<td>0.421</td>
<td>0.610</td>
<td>0.531</td>
</tr>
<tr>
<td>Binary-SVM</td>
<td>0.2021</td>
<td>0.5302</td>
<td>0.5862</td>
<td>0.6332</td>
</tr>
<tr>
<td>Proposed method</td>
<td>0.1902</td>
<td>0.5524</td>
<td>0.6788</td>
<td>0.7156</td>
</tr>
</tbody>
</table>

### TABLE III

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Hamming loss</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5</td>
<td>0.001</td>
<td>0.987</td>
<td>0.992</td>
<td>0.995</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.035</td>
<td>0.273</td>
<td>0.273</td>
<td>0.276</td>
</tr>
<tr>
<td>Binary-SVM</td>
<td>0.0011</td>
<td>0.9891</td>
<td>0.9933</td>
<td>0.9958</td>
</tr>
<tr>
<td>Proposed method</td>
<td>0.00006</td>
<td>0.9946</td>
<td>0.9958</td>
<td>0.9956</td>
</tr>
</tbody>
</table>

### TABLE IV

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Hamming loss</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5</td>
<td>0.148</td>
<td>0.576</td>
<td>0.579</td>
<td>0.588</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.139</td>
<td>0.605</td>
<td>0.615</td>
<td>0.624</td>
</tr>
<tr>
<td>Binary-SVM</td>
<td>0.103</td>
<td>0.702</td>
<td>0.715</td>
<td>0.720</td>
</tr>
<tr>
<td>Proposed method</td>
<td>0.102</td>
<td>0.705</td>
<td>0.722</td>
<td>0.728</td>
</tr>
</tbody>
</table>

The experimental results of all evaluation metrics demonstrate that the proposed method outperforms some existed methods. It can improve the classification performance efficiently.

V. CONCLUSIONS

In this paper, we present an improved multi-label classification method based on label ranking strategy and label constraints. One-against-all decomposition technique is used to divide a multi-label problem into binary class sub-problems firstly, and one binary SVM classifier is trained for each possible label. Then, AP clustering and FP-Growth are used to mine local label constraint rules based on training dataset. Finally, a correction model based on label constraints is used to correct the probabilistic outputs of SVM classifiers for label ranking.

Three well-known multi-label benchmark are used to evaluate the proposed method. Experiment results show that the proposed method outperforms some conventional methods.