An Improved Multi-label Classification Based on Label Ranking and Delicate Boundary SVM

Benhui Chen, Weifeng Gu and Jinglu Hu (Member, IEEE)

Abstract—In this paper, an improved multi-label classification is proposed based on label ranking and delicate decision boundary SVM. Firstly, an improved probabilistic SVM with delicate decision boundary is used as the scoring method to obtain a proper label rank. It can improve the probabilistic label rank by introducing the information of overlapped training samples into learning procedure. Secondly, a threshold selection related with input instance and label rank is proposed to decide the classification results. It can estimate an appropriate threshold for each testing instance according to the characteristics of instance and label rank. Experimental results on four multi-label benchmark datasets show that the proposed method improves the performance of classification efficiently, compared with binary SVM method and some existing well-known methods.

I. INTRODUCTION

Multi-label classification problem is an extension of traditional multi-class classification problem in which each sample may belong to several classes simultaneously. In multi-class classification, every instance can be assigned to only one class even though the number of classes is more than two. The essential difference between two problems is that classes in multi-class classification are assumed to be mutually exclusive while classes in multi-label classification are often correlated. Multi-label classification methods are increasingly required by modern applications, such as protein function classification [1], text categorization [2], [3], music categorization [4] and semantic scene classification [5]. In those applications, samples in the training set are associated with a set of labels and the task is to predict the label set for the unseen instances. Generally, traditional multi-class learning algorithms cannot work with multi-label problem effectively.

Label ranking strategy is often used in multi-label classification methods [1], [6]. This strategy requires a real-valued score for each class (label) to order a label rank, the labels of higher score values are more related with the new instance, then classifies a new instance into the classes that ranking score above a threshold. In a ranking based classification, both the scoring method and the threshold selection can influence the classification results significantly. An ideal ranking based classifier can be imagined as follows, the scoring method can provide a proper label rank (i.e., the labels of higher score values are more related with the instance). The threshold selection strategy can estimate an appropriate threshold according to the characteristics of instance and label ranking strategy.

The scoring methods are the major focus of research in the multi-label classification [7], [8]. However, the threshold selection is often processed as a unimportant post-processing step and so far few studies have been done on this subject. For the document classification applications, in Ref. [9], threshold selection techniques are successfully applied to obtain better performance. A per-class threshold selection strategy has been used to improve the performance of the one-versus-all binary Support Vector Machine (SVM) multi-label classification in Ref. [10]. It estimates an appropriate threshold value for each class by optimizing macro-average F-measure.

In this paper, we propose a multi-label classification based on label ranking and delicate boundary SVM. Firstly, an improved probabilistic SVM with delicate decision boundary is used as the scoring method. A correction model with delicate decision boundary is used to improve the probabilistic label ranking by introducing the information of overlapped training samples into learning procedure. Secondly, a threshold selection related with input instance and label rank is proposed to decide the classification results. The validation approach is utilized to estimate the ideal threshold values for all training samples by optimizing a certain evaluation measure. Then according to this ideal threshold set of training samples, in test procedure, a K-nearest neighbors (KNN) strategy is used to estimate an appropriate threshold for each testing instance. In experiments, four benchmark datasets from different applications are used to evaluate the proposed multi-label classification.

The rest parts of the paper are organized as follows. Section II gives a brief overview of ranking based multi-label classification, SVM classification and threshold selection strategies. Section III describes the details of the proposed ranking based multi-label classification method. It includes the scoring method of the probabilistic SVM with delicate decision boundary and the threshold selection related with input instance and label rank. Section IV presents the four benchmark datasets and the results of experimental performance evaluation. Finally, the conclusions are discussed.

II. RANKING BASED MULTI-LABEL CLASSIFICATION AND SUPPORT VECTOR MACHINE

A. Ranking Based Multi-label Classification

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

For the ranking based multi-label classification, the learning system is expected to produce a real-valued scoring function of the form \( f(\cdot, \cdot) : X \times Y \rightarrow IR \). It is supposed that, a successful learning system 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:

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

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

**B. Support Vector Machine**

SVM is a supervised learning method introduced by Vapnik based on his Statistical Learning Theory and Structural Minimization Principle [11]. The basic idea of using SVM for classification is to find the optimal separating hyperplane between the positive and negative samples. The optimal hyperplane is defined as the one giving the maximum margin between the training samples that are closest to it. Support vectors are the samples that lie closest to the separating hyperplane. Once this hyperplane is found, new samples can be classified simply by determining on which side of the hyperplane they fall.

Given training data, \( T = \{(x_1, y_1), \ldots, (x_m, y_m)\} \), where \( x_i \in \mathbb{R}^n \) is the input pattern, and \( y_i \in \{-1, 1\} \) is the class label for a two-class problem, SVM for classification attempts to find a classifier \( f(X) \), which minimizes the expected misclassification rate. A linear classifier \( f(X) \) in SVM is equivalent to solving a convex quadratic optimization problem bellow:

\[
\min \frac{1}{2} ||W||^2 + C \sum_{i=1}^{n} \xi_i \tag{2}
\]

subject to \( y_i (\langle W, x_i \rangle + b) \geq 1 - \xi_i \) and \( \xi_i \geq 0 \), where \( C \) is called the regularization parameter, and is used to balance the complexity of classifier and the classification accuracy on the training set \( T \). This quadratic problem is generally solved through its dual formulation [12]. Simply replacing the involved vector inner-product with a non-linear kernel function converts linear SVM into a more flexible non-linear SVM, which is the essence of the famous kernel trick.

**C. Threshold Selection Strategies**

There are three commonly used threshold selection strategies for classification [9], including the rank-based method (RCut), the proportion-based assignments (PCut) and the score-based local optimization (SCut). Let \( k \) denotes the number of classes (labels) in the problem, \( d \) denotes the

number of instances in the testing (or validation) set, and assume one score is produced by the classifier for each instance-class pair. The threshold selection algorithms are defined as follows.

**RCut**, it is a per-instance strategy, for each instance, sort classes by score and assign positive to each of the \( t \) top-ranking classes. The parameter \( t \) is a value (an integer between 1 and \( k \) ) can be either specified by the user or automatically tuned using a validation set. That is, the value of \( t \) optimizing the global performance of the classifier on the validation set is fixed when applying the classifier on new instance in the testing set. RCut with \( t = 1 \) is also commonly used for single-label classification.

**PCut**, it is a per-class strategy, for each class \( c_j \), sort the testing instance by score and assign positive to each of the \( k_j \) top-ranking instances, where \( k_j = P(c_j) \times x \times d \) is the number of instance assigned to class \( c_j \), and \( P(c_j) \) is the prior probability (estimated using a training set) for an arbitrary instance to be an member of class \( c_j \). The parameter \( x \) is automatically tuned in the same fashion as tuning \( t \) for RCut, by varying the value of \( x \) until the global performance of the classifier is optimized on the validation set.

**SCut**, it is also a per-class strategy, score a validation set of documents for each class and tune the threshold over the local pool of score values until the optimal performance of the classifier is obtained for that class. The per-class thresholds is fixed when applying the classifier to new instance in the testing set.

The three strategies have different properties of information used, suitability, flexibility and overfitting risk. The choice of threshold selection strategy depends on the characteristics of classifier and application [9].

**III. THE PROPOSED RANKING BASED MULTI-LABEL CLASSIFICATION**

According to the previous analysis, a ranking based multi-label classification consists of two important parts: scoring method and threshold selection. Scoring method should provide a proper label rank, i.e., the labels of high score values are more related with the instance. Threshold selection strategy should choose appropriate thresholds for testing instances.

In the proposed ranking based multi-label classification, instead of estimating a static threshold for all testing instances in conventional methods, threshold selection is considered as a procedure that relies on both input instance and label rank. It is expected to choose appropriate thresholds according to the characteristics of instances and label ranking strategy.

As showed in Fig.1, efforts on both parts are made to improve the classification performance. For the scoring part, an improved probabilistic SVM with delicate decision boundary is used to obtain a proper label ranking results. For the threshold selection part, based on the ideal threshold set of training data estimated by validation, a KNN learning model is used to realize the threshold selection procedure related with both input instance and label rank.
A. Multi-label Classification Based on Probabilistic SVM with Delicate Decision Boundary

The probabilistic SVM with deliberate decision boundary proposed in Ref [13] is used as scoring method to obtain label rank for instance. In this method, for each overlapped label pair, a correction model is designed to improve the probabilistic SVM classification results for those overlapping (double-label) samples. The principle of correction model for overlapped label pair is described as follows.

The basic Binary-SVM method cannot work very well for those overlapping samples in practice because of the complexity of multi-label problem. But there is an important common characteristic of double-label samples of overlapped label pair (two labels have same overlapped samples). It is that the majority of double-label samples are distributed near to two binary SVM separating surfaces simultaneously. The correction model is built according to this common characteristic of overlapped label pair. As showed in Fig.2, training samples in this overlapping sample space are utilized to fit two delicate boundaries for overlapped samples. Then, two delicate boundaries are used to obtain better classification results of testing instances in this overlapping space.

1) SVM+Sigmoid Probabilistic Classifier: The SVM+Sigmoid method is used to estimate the classification probabilities. 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 [14].

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

(3)

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,

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

\[
p_i = \frac{1}{1 + \exp(AF_1 + B)}, \quad f_i = f(x_i)
\]

\[
t_i = \begin{cases} 
N_{+1} & \text{if } y_i = 1 \\
N_{+2} & \text{if } y_i = -1 
\end{cases} \quad i = 1, 2, \ldots m' \quad (4)
\]

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

2) Correction Model Based on Delicate Decision Boundaries: Considering the basic problem of classifying the essentially overlapping classes of label \( p \) and \( q \). As shown in Fig.2, two probabilistic separating surfaces are constructed by SVM+Sigmoid classifiers. They divide the sample space into four areas: “only \( p \) area 1”, “only \( q \) area 2”, “\( p \) and \( q \) area 3” and “no one’s area 4”. It must be noted that in some cases (e.g. for linearly separable classes), one or two of these areas may be empty. The two SVM separating surfaces are constructed in different high dimensional space, we only want to summarize the probabilistic relativity of two surfaces, so they are analyzed in same dimensional space for convenience.

The correction model for the certain overlapped label pair of \( p \) and \( q \) is realized as follows. The distributing range of overlapping sample space (specified by the four borders \( \epsilon_{p+}, \epsilon_{p-}, \epsilon_{q+} \) and \( \epsilon_{q-} \) ) are estimated by double-label training samples and their relative information with two separating surfaces firstly. Then, training samples in this overlapping sample space are utilized to fit two delicate boundaries for two label respectively (bold curves shown in Fig.2). Finally, the delicate boundaries are used to obtain better classification results of SVMs. The outline is listed as follows and the details can be found in [13].
Step 1, the $k$ (the number of label) binary SVMs are built by one-versus-rest strategy. The Platt’s sigmoid method is used to obtain the probabilistic outputs of SVMs. Thus, we can get $k$ probabilistic classifiers.

Step 2, according to the one-versus-one strategy and probabilistic outputs of $k$ binary SVMs, the correction models are built for corresponding overlapped label pairs (those pairs have same overlapped samples, the maximum number of correction models is $k \times (k - 1)/2$).

1) The double-label training samples near to two separating surfaces simultaneously are selected, and their SVM distance values to hyperplane of two labels are used to estimate the four range borders of two separating surfaces (as showed in Fig.2). The four borders can be used to decide the range of overlapping sample space.

2) According to the four borders, training samples in the range of overlapping sample space are selected. According to those overlapped training samples, a statistical method and the relative information of two separating surfaces are used to fit two curve functions as delicate classification boundaries respectively.

3) In the testing procedure, the four range borders and fitted delicate boundaries of two separating surfaces are utilized to correct the classification results. For every instance in the testing dataset, if it belong to the overlapping sample space, its final probabilistic outputs are corrected by two fitted curve functions of two labels.

Step 3, in integration and ranking procedure, for each sample-label pair, the mean value of corresponding correction models outputs is used as the final probability for this label. Hence, we can get a probabilistic label rank of $k$ values for each instance.

B. The Proposed Threshold Selection

Different from the conventional methods to estimate a static threshold for all testing instances, the proposed threshold selection is considered as a procedure that relies on both input instance and label rank. It is expected to choose appropriate thresholds according to the characteristics of instance and label ranking strategy. Based on the ideal threshold set of training data estimated by validation strategy, a KNN learning model is used to realize the threshold selection related with input instance and label rank.

As showed in Fig.1, firstly, a $d$-fold validation strategy is utilized to estimate ideal threshold values for training samples. Training data is split into several training/validation subsets. Then, training subsets are used to train the classifiers, and estimate an ideal threshold value for each sample in validation subset by optimizing a certain evaluate measure. Secondly, according to the ideal per-sample threshold values of training samples, a KNN method is utilized to obtain appropriate threshold values for testing instances. The outline of the proposed threshold selection method is described as follows.

Step 1, split training data into $d$ folds. According to the fold validation method, for fold $i = 1, \ldots, d$.

1) Train the SVM+Sigmoid multi-label classifier (mentioned in previous subsection) on the $d-1$ folds (except the validation fold $i$).

2) Based to the trained multi-label classifier, consider the fold $i$ as the testing data, obtain the probabilistic test outputs (score values for each classes) for samples in the fold $i$.

3) For each sample $I_p$ in the validation fold $i$, sort the score values of classes (labels) as a descending rank $(l^{(1)}, l^{(2)}, \ldots, l^{(k)})$, where $l^{(1)}$ is the label of maximum score and $k$ is the number of classes. Then, respectively take subset $\{l^{(1)}\}, \{l^{(1)}l^{(2)}\}, \{l^{(1)}l^{(2)}l^{(3)}\}, \cdots$ (from the subset contains the top label, add a new label one by one sequentially) as the candidate result label set for the sample $I_p$ and evaluate it by a certain evaluate measure. Select the best candidate label set which optimizing the evaluation measure $\{l^{(1)}l^{(2)}\cdots l^{(j)}\}, 1 < j < k$. Then, take the mean of score values of $l^{(j)}$ and $l^{(j+1)}$ as the ideal threshold value of the sample $I_p$.

Step 2, the ideal threshold values for all training samples are obtained by validation of Step 1. This ideal threshold set of training data is utilized in test procedure to estimate the thresholds of testing instances by KNN strategy.

Step 3, in test procedure, for each instance $I_q$ in the testing data, the similarities with all training samples are calculated by a certain distance metric (Euclidean distance, Hamming distance or other measures). The mean value of $kn$ nearest samples’ thresholds is assigned as the threshold of the testing instance $I_q$, where $kn$ is the parameter of KNN method.

The proposed threshold selection is a per-instance strategy similar with the RCut method. But different from RCut method, instead of using a static parameter $t$ as the threshold for all testing instances, the proposed method estimate threshold for different testing instance by a lazy learning method of KNN according to the ideal threshold values of training data. For the $d$-fold validation strategy, the parameter...
$d$ should be chosen by considering the trade-off between the performance and computational consume.

IV. EXPERIMENTS AND RESULTS

A. Benchmark Datasets

Four benchmark multi-label datasets are used to evaluate the proposed improved multi-label classification. There are Yeast (a gene function dataset), Genbase (a motif-based protein dataset), Scene (a scene dataset) and Emotions (a music dataset). Four datasets are downloaded from website of [6]. The detailed information about them, such as the number of samples, attributes, classes and their average number of labels or classes are listed in Tab. I.

B. Evaluation Measures

Multi-label classification requires different metrics than those used in traditional single-label classification. 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 $z_i = C(x_i)$ be the set of labels predicted by $C$ for example $x_i$. The following evaluation metrics for label ranking used in [1], [3], [6], [15] are used in this paper:

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:

$$HammingLoss(C, D) = \frac{1}{|D|} \sum_{i=1}^{D} \frac{1}{\|y_i\Delta z_i\|} \tag{5}$$

where $\Delta$ means for the XOR operation of two sets. The smaller this measure is, the better our method performs.

Accuracy, Precision and Recall evaluation criteria are defined as follows.

$$Accuracy(C, D) = \frac{1}{|D|} \sum_{i=1}^{D} \frac{|y_i \cap z_i|}{|y_i \cup z_i|} \tag{6}$$

$$Precision(C, D) = \frac{1}{|D|} \sum_{i=1}^{D} \frac{|y_i \cap z_i|}{|z_i|} \tag{7}$$

$$Recall(C, D) = \frac{1}{|D|} \sum_{i=1}^{D} \frac{|y_i \cap z_i|}{|y_i|}. \tag{8}$$

There is a common characteristic for above three measures, the larger measure value, the better the classification performance.

C. Experimental Setting and Results

In our experiments, the LSSVM [16] is taken as a basis. SVM parameters are chosen by cross-validation procedure. In the training procedure, the methods mentioned in Section III are utilized to train SVM classifiers, calculate the parameters of sigmoid functions and correction models and estimate the ideal threshold values for all training samples. In the testing procedure, for each test instance, a probabilistic rank of $k$ values for all labels is obtained by final decision procedure. Then, according to the ideal threshold values of training samples, a proper threshold estimated based on KNN method. This proper threshold is used to determine the label subset finally.

We select the Accuracy evaluation metric as the optimizing measure for estimating ideal threshold set of training data. In addition, there are two important parameters for the proposed threshold selection. The first is parameter $d$ of $d$-fold validation, considering trade-off between performance and computational consume, we set this parameter according to the number of samples. The second is parameter $k$ of KNN method, we select this parameter by experimentalism. Two parameters for four datasets are listed in Tab. II, and the Fig.3 shows relation of parameter $k$ and performance of classification for the Yeast dataset.

### Table I

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Labels</th>
<th>Features</th>
<th>Density</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast</td>
<td>14</td>
<td>103 (Numeric)</td>
<td>4.25</td>
<td>1500917</td>
</tr>
<tr>
<td>Genbase</td>
<td>27</td>
<td>1186 (Discrete)</td>
<td>1.35</td>
<td>463/199</td>
</tr>
<tr>
<td>Scene</td>
<td>6</td>
<td>294 (Numeric)</td>
<td>1.08</td>
<td>1211/1196</td>
</tr>
<tr>
<td>Emotions</td>
<td>6</td>
<td>72 (Numeric)</td>
<td>1.87</td>
<td>391/202</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$d$ of fold validation</th>
<th>$k$ of KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yeast</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>Genbase</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Scene</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>Emotions</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

![Fig. 3. Relation between parameter $k$ of KNN in threshold selection and performance of classification for the Yeast dataset](image)
TABLE IV
EXPERIMENTAL RESULTS FOR THE GENBASE DATASET.

<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><strong>0.00006</strong></td>
<td><strong>0.9912</strong></td>
<td><strong>0.9947</strong></td>
<td><strong>0.9950</strong></td>
</tr>
</tbody>
</table>

TABLE V
EXPERIMENTAL RESULTS FOR THE SCENE DATASET.

<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>0.139</td>
<td>0.513</td>
<td>0.611</td>
<td>0.534</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.247</td>
<td>0.435</td>
<td><strong>0.816</strong></td>
<td>0.443</td>
</tr>
<tr>
<td>Binary-SVM</td>
<td>0.1032</td>
<td>0.7021</td>
<td>0.7134</td>
<td>0.7372</td>
</tr>
<tr>
<td>Proposed method</td>
<td><strong>0.0917</strong></td>
<td><strong>0.7263</strong></td>
<td>0.7752</td>
<td><strong>0.7933</strong></td>
</tr>
</tbody>
</table>

We implemented experiments of the proposed method and Binary-SVM method on four datasets. The results on the datasets of Yeast, Genbase and Scene are compared with two popular multi-label classification methods: C4.5 and Naive Bayes [3], [6]. The results of the two methods are cited from [6]. All experimental results for four datasets are presented in the Tab.III, Tab.IV, Tab.V and Tab.VI. The winning results are marked with bold font. The experimental results of all evaluation metrics demonstrate that the proposed method improves the performance of classification efficiently.

V. CONCLUSIONS

This paper proposes an improved multi-label classification based on label ranking and delicate decision boundary SVM. The improved probabilistic SVM with correction model is used as the scoring method. It can improve the probabilistic label ranking by introducing the information of multi-label training samples in overlapping sample space into learning procedure. A per-instance threshold selection is proposed to decide the classification results. In threshold selection, validation approach is utilized to estimate the ideal threshold value for each sample in training set by optimizing a certain evaluation measure. Then according to those ideal thresholds of training samples, a KNN method is used to estimate the threshold for each instance. The experimental results on four benchmark datasets show that the proposed method improves the performance of classification efficiently, compared with binary SVM method and some existing well-known methods.

ACKNOWLEDGMENT

This research was supported by Waseda University Global COE Program ‘International Research and Education Center for Ambient SoC’ sponsored by MEXT, Japan.

REFERENCES