International Journal of Information Technologies and Systems Approach
Volume 16 • Issue 3

# Multilabel Classifier Chains Algorithm Based on Maximum Spanning Tree and Directed Acyclic Graph

Wenbiao Zhao, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, China

Runxin Li, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, China*

https://orcid.org/0000-0003-4272-3437

Zhenhong Shang, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, China

## ABSTRACT

The classifier chains algorithm is aimed at solving the multilabel classification problem by composing the labels into a randomized label order. The classification effect of this algorithm depends heavily on whether the label order is optimal. To obtain a better label ordering, the authors propose a multilabel classifier chains algorithm based on a maximum spanning tree and a directed acyclic graph. The algorithm first uses Pearson's correlation coefficient to calculate the correlation between labels and constructs the maximum spanning tree of labels, then calculates the mutual decision difficulty between labels to transform the maximum spanning tree into a directed acyclic graph, and it uses topological ranking to output the optimized label ordering. Finally, the authors use the classifier chains algorithm to train and predict against this label ordering. Experimental comparisons were conducted between the proposed algorithm and other related algorithms on seven datasets, and the proposed algorithm ranked first and second in six evaluation metrics, accounting for 76.2% and 16.7%, respectively. The experimental results demonstrated the effectiveness of the proposed algorithm and affirmed its contribution in exploring and utilizing label-related information.

## KEYWORDS

classifier chains, directed acyclic graph, maximum spanning tree, multilabel classification, Pearson correlation coefficient

## INTRODUCTION

Unlike the traditional single-label classification problem, the multilabel classification (MLC) problem allows a sample to simultaneously have multiple label categories. (For example, a news article can belong to the topics of both technology and culture.) This ability means multilabel classification problems can reflect many real-world problems. Examples include text classification (Liu et al., 2021; Minaee et al., 2021; Nam et al., 2014), video annotation (Markatopoulou et al., 2018), image annotation (Lanchantin et al., 2021; Zhu et al., 2017), music classification (Tiple et al., 2022), and protein function prediction (Guan et al., 2018). In practical production applications, labeling samples by hand is difficult and expensive. Thus, solving the multilabel classification problem is valuable.

DOI: 10.4018/IJITSA.324066 *Corresponding Author

A straightforward solution to MLC is the binary relevance (BR) algorithm (Boutell et al., 2004). It transforms the original multilabel problem into a series of single-label problems. This algorithm, however, although simple and efficient, does not utilize the information brought between the labels and therefore does not obtain better classification results. It is practicable to improve the multilabel classification accuracy by using the information hidden between the labels. Typical approaches include stacked binary relevance (2BR) (Godbole & Sarawagi, 2004), classifier chains (CC) (Read et al., 2011), multilabel k-nearest neighbor (ML-kNN) (Zhang & Zhou, 2007), rank support vector machine (RankSVM) (Elisseeff & Weston, 2001), among others.

The CC algorithm uses labels as additional features to exploit the correlation information between the labels. The specific practice is to select a label ordering, and all the labels are ranked before the target labels are used as additional features to participate in the training and predict the target label to finally obtain a multilabel classifier chains. The key desired outcome of using the CC algorithm is to find the optimal label ordering. If the predecessors of a label are highly correlated to it, then the additional features can help improve the performance of the corresponding classifier. The traditional CC algorithm determines the label ordering randomly, which has low classification performance and low robustness. To solve the this problem, many variant algorithms of the CC algorithm have been proposed such as probabilistic classifier chains (PCC) (Cheng et al., 2010), ensemble classifier chains (ECC) (Rokach, 2010), conditional entropy-based classifier chains (CEbCC) (Jun et al., 2019), and group sensitive classifier chains (GCC) (Huang et al., 2015). These algorithms improve the classification performance of CC algorithms, but the time complexity is high. Also, they mostly consider only the positive relationship between labels and ignore the negative correlation. Another problem to be considered is how to define the backward and forward order of two labels with correlation.

To address problems of the e CC-related algorithms, we propose a multilabel classifier chains algorithm based on a maximum spanning tree and directed acyclic graph (maxSTCC). The contributions of this paper are listed as follows:

1. The Pearson correlation coefficient (Sinhashthita & Jearanaitanakij, 2020) is used to calculate the degree of correlation between the labels, and the absolute value is taken to consider both positive and negative correlations between the labels as correlations between the labels. An undirected weighted graph of labels is constructed, where the vertices represent labels, and the weights of the edges indicate the degree of correlation among connected labels.
2. The maximum spanning tree algorithm is used to transform the undirected weighted graph of labels into a maximum spanning tree to maximize the utilization of the correlation information between labels.
3. Conditional entropy is used to define the mutual decision difficulty between two connected labels in the maximum spanning tree, and it takes the direction with lower decision difficulty as the dependence direction between the two labels and finally transforms the maximum spanning tree of labels into a directed acyclic graph (DAG). This process solves the problem of how to order two related labels.
4. To illustrate the contribution made by the maximum spanning tree, the classifier chains algorithm for constructing DAG based on conditional entropy (CEDAGCC) is proposed as a control algorithm. The algorithm directly constructs the directed cyclic graph (DCG) of labels by conditional entropy, and it then converts the DCG into DAG of labels by removing the rings in DCG.
5. The labels in the DAG are realized as a label ordering using topological ordering, and the CC algorithm is used to train and predict based on that label ordering. The maxSTCC algorithm is experimentally compared with other related algorithms, and the experimental results show that the algorithm in this study can obtain more stable and excellent label ranking and better classification performance.

## RELATED WORK

### Preliminaries

We let $D = \{(x_i, y_i) \mid 1 \le i \le n\}$ denote the training data set, which consists of $n$ instances and use $x_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,d}]$ and $y_i = [y_{i,1}, y_{i,2}, \cdots, y_{i,q}]$ to denote the feature data and label data of the *i*th sample $(x_i, y_i)$, where $d$ and $q$ denote the number of features and the number of labels, respectively. We let $X = [x_1, x_2, \cdots, x_n]^T$ and $Y = [y_1, y_2, \cdots, y_n]^T$ denote the feature dataset and label dataset and use $L = \{l_1, l_2, \cdots, l_q\}$ to denote the set of $q$ labels. When a training sample $(x_i, y_i)$ is tagged with $l_j$, then $y_{i,j} = 1$; otherwise, $y_{i,j} = 0$.

### Multilabel Classification

The utilization of information brought by the correlation between labels to improve the classification performance has been a research topic in recent years. According to the level of label correlation considered by multilabel classification algorithms, existing algorithms can be classified into first-order strategies, second-order strategies, and higher-order strategies (Zhang & Zhou, 2013).

First-order strategies do not consider correlations between labels, they train and predict each label in turns. The BR algorithm and the ML-KNN algorithm are classification first-order strategy algorithms. The BR algorithm treats each label classification problem as a separate single-label problem and trains a classifier for each label using the full feature dataset. The ML-KNN algorithm handles the multilabel classification problem by making a simple improvement to the KNN algorithm. It counts the number of labels included in the k-nearest neighbors for each label independently without considering the dependencies between labels.

Second-order strategies examine correlations between pairs of labels. Multilabel algorithms included in this strategy have improved classification results compared with that of first-order strategies. The calibrated label ranking (CLR) algorithm (Fürnkranz et al., 2008) and the RankSVM algorithm are two second-order strategy algorithms. The CLR algorithm sorts and splits the labels by comparing them in order to deal with multilabel classification problems. The RankSVM algorithm measures the correlation between relevant and irrelevant label pairs by the label ranking loss function and constructs a convex quadratic optimization problem to solve the multilabel classification problem.

Higher-order strategies consider higher-order correlations between labels (e.g., the relevance of a label to all the remaining labels). Multilabel algorithms involved with higher-order strategies obtain the best classification results, but at the same time, the time complexity increases due increased label correlations. The 2BR algorithm and CC algorithm address the problem of the BR algorithm in that it cannot exploit label relevance by using labels as input features for the feature space. Both are higher-order strategy algorithms that can improve the BR algorithm. The BR algorithm considers label relevance by stacking two layers of the BR algorithm, where the predicted labels of the first layer are used as input features to the feature space of the second layer, and the predicted labels of the second layer are used as the final result. The CC algorithm trains the current label classifier by forming a chain of all labels ranked before that label as input features for the feature space to train the classifier. The label-related information is passed through the chain while retaining the low time complexity of the BR algorithm, but the classification effect of the CC algorithm is vulnerable to the label sequence.

The focus of this research is to establish an optimal sequence of label ordering for the improvement of the CC algorithm. The probabilistic classifier chains (PCC) algorithm (Cheng et al., 2010) finds the sequence of labels with the highest confidence by iterating all label orderings. This algorithm, however, can only be used for datasets with a small number of labels due to its high time complexity. The ECC algorithm determines the final prediction by training multiple chains of random classifiers and voting the prediction results of each classifier chains. The CEbCC algorithm first calculates the conditional entropy between labels and then counts the sum of conditional entropy of each label to rank them. The Bayesian chain classifiers (BCC) algorithm (Zaragoza et al., 2011) and the Bayesian network-based label

correlation analysis for multilabel classifier chain (BNCC) algorithm (Wang et al., 2021) determine the label ordering by building a Bayesian network of labels. The association rules-based classifier chains method (ARECC) algorithm (Jiaman et al., 2022) ranks labels by mining association rules between them.

## CC Algorithm

The main task of multilabel classification is to establish the correspondence from the data feature space to the label space. Supposing $h_j$ denotes the mapping of feature data $X$ to the *j*th label: $h_j : X \to l_j$, where $l_j$ takes the value 0 or 1, then $h_j$ is taken as the binary classifier for the *j*th label. The classical BR algorithm trains a separate classifier for each label independently for a total of $q$ binary classifiers: $h_1, h_2, \cdots h_q$.

Using the label relevance to improve classification performance is the focus of multilabel classification research. To address the problem that the BR algorithm cannot utilize the label relevance, the CC algorithm introduces label relevance by using labels as an additional dimension of features. The specifics are shown in Table 1.

Table 1 illustrates the CC algorithm training process by an example with feature $x = [0.8, 0.5, 0.47, 1.3]$ and label $y = [1, 0, 0, 1, 0]$, and label ordering $l_1, l_2, \cdots, l_5$, where $x'$ denotes a new feature which is formed by the label as an input feature to $x$.

For the $1 \le j \le q$ label, the following equation describes the training process of the CC algorithm.

$$\{(x_i, y_{i,1}, y_{i,2}, \cdots, y_{i,j-1}; y_{i,j}) \mid 1 \le i \le n\} \mapsto h_j \tag{1}$$

Similarly, the following equation describes the predicted label $\hat{y}$ of $x$.

$$\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_q] = [h_1(\mathbf{x}), h_2(\mathbf{x}, \hat{y}_1), \cdots, h_q(\mathbf{x}, \hat{y}_1, \hat{y}_2, \cdots, \hat{y}_{q-1})] \tag{2}$$

# PROPOSED METHOD

## Undirected Weighted Graph of Labels

The Pearson correlation coefficient is widely used to measure the degree of correlation between two variables and takes on a value between -1 and 1. When the value is 0, the two variables are not correlated at all. When its value is greater than 0, the two variables show positive correlation, and the

**Table 1. Training phase by CC algorithm**

| Classifiers | Features $\mathbf{x}'$ | Label Value |
|---|---|---|
| $h_1$ | $[0.8, 0.5, 0.47, 1.3]$ | 1 |
| $h_2$ | $[0.8, 0.5, 0.47, 1.3, \mathbf{1}]$ | 0 |
| $h_3$ | $[0.8, 0.5, 0.47, 1.3, \mathbf{1}, \mathbf{0}]$ | 0 |
| $h_4$ | $[0.8, 0.5, 0.47, 1.3, \mathbf{1}, \mathbf{0}, \mathbf{0}]$ | 1 |
| $h_5$ | $[0.8, 0.5, 0.47, 1.3, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1}]$ | 0 |

higher the positive correlation is, the closer the value is to 1. When the value is less than 0, the two variables show negative correlation; and the higher the negative correlation, the closer the value is to -1.

A small modification to the Pearson correlation coefficient is used to calculate the degree of correlation between two labels (Tsoumakas et al., 2009). The relevance of labels $l_j$ and $l_k$ in this paper is defined as follows:

$$\phi(l_j, l_k) = \frac{AD - BC}{\sqrt{(A+B)(C+D)(A+C)(B+D)}} \tag{3}$$

where $A$, $B$, $C$ and $D$ denote the four combinations of statistics for labels $l_j$ and $l_k$, respectively. The calculation formulae are as follows:

$$A = \sum_{i=1}^{n} [\![ y_{i,j} = 1 \ and \ y_{i,k} = 1 ]\!] \tag{4}$$

$$B = \sum_{i=1}^{n} [\![ y_{i,j} = 1 \ and \ y_{i,k} = 0 ]\!] \tag{5}$$

$$C = \sum_{i=1}^{n} [\![ y_{i,j} = 0 \ and \ y_{i,k} = 1 ]\!] \tag{6}$$

$$D = \sum_{i=1}^{n} [\![ y_{i,j} = 0 \ and \ y_{i,k} = 0 ]\!] \tag{7}$$

where $[\![\cdot]\!]$ indicates that the entire bracket takes the value of 1 when the condition inside the bracket holds; otherwise, the value is 0.

In order to measure the correlation between labels more comprehensively, both negative and positive correlations between labels are considered as the correlation measure between labels. Then, by calculating the correlation degree between two labels, a label correlation matrix $R$ can be obtained and defined as follows:

$$R = [r_{j,k} = \mid \phi(l_j, l_k) \mid], 1 \le j \ne k \le q \tag{8}$$

By calculating the correlation degree between labels a weighted undirected connected graph G=(V, E, W) can be constructed with labels as vertices and label correlations as weighted edges, where the set of vertices is $V = \{l_1, l_2, \cdots, l_q\}$, the set of edges is $E = \{(l_j, l_k) \mid 1 \le j \ne k \le q\}$, and the set of edge weights is $W = \{w_{j,k} = w(l_j, l_k) = r_{j,k} \mid r_{j,k} \in R, 1 \le j \ne k \le q\}$. Then the adjacency matrix $A$ of the weighted undirected graph G of labels is

$$A_{j,k} = \begin{cases} r_{j,k} & r_{j,k} \in R, 1 \le j \ne k \le q \\ 0 & 1 \le j = k \le q \end{cases} \tag{9}$$

The values in the adjacency matrix $A$ are the weights of the corresponding two vertices. For the weighted undirected graph G of labels the adjacency matrix $A$ is a symmetric matrix.

## Maximum Spanning Tree of Labels

A connected graph without loops is called a tree, and a spanning tree is a connected spanning subgraph of an undirected connected graph without loops. As an important problem in the graph theory, the spanning tree is widely used in fields such as network optimization, data structure, engineering, and combinatorial optimization. The spanning tree with the largest sum of edge weights among all spanning trees of a graph is the maximum spanning tree. Commonly used maximum spanning tree algorithms include Kruskal's algorithm, Prim's algorithm, and the broken circle method. We use the idea of the Prim algorithm to construct the maximum spanning tree $T = (V_{tree}, E_{tree})$ of labels. The specific process is shown in Algorithm 1.

## Directed Acyclic Graph of Labels

The maximum spanning tree $T = (V_{tree}, E_{tree})$ of the labels is obtained as above. The maximum spanning tree of labels is constructed to maximize the consideration of label relevance and thus optimizes the label ordering to improve the classification performance of the classifier chains algorithm. In order to derive the label ordering, the maximum spanning tree of labels is converted into a DAG. The key issue in this process is determining the direction of each edge of the maximum spanning tree.

We first use information entropy to define the uncertainty of the label. The uncertainty of label $l_j \in L$ is

$$H(l_j) = -\sum_{y_j \in \{0,1\}} p(y_j) \log_2 p(y_j) \tag{10}$$

The uncertainty $H(l_j)$ of label $l_j$ is minimized when all values of label $l_j$ are 1 or 0. The uncertainty $H(l_j)$ of label $l_j$ is maximized when half of the values of label $l_j$ are 1 and half are 0. Conditional on the given label $l_k \in L$, the uncertainty of label $l_j$ is defined by the conditional entropy as follows:

**Algorithm 1. Generate maximum spanning tree of labels**

| |
|---|
| Input: Weighted graph of labels G = (V, E, W). |
| Output: Maximum spanning tree $T = (V_{tree}, E_{tree})$ of labels. |
| 1. Let $V_{tree} = \{l_1\}$, $E_{tree} = \varnothing$ |
| 2. while $V_{tree} \neq V$ do |
| 3 if Any $l_j \in V_{tree}$, any $l_k \in V$ - $V_{tree}$, and $w(l_j, l_k)$ reaches the maximum weight |
| 4. do $V_{tree} \leftarrow V_{tree} \cup l_k$, $E_{tree} \leftarrow E_{tree} \cup (l_j, l_k)$ |
| 5. endwhile |

$$\begin{aligned} H(l_j \mid l_k) &= \sum_{y_k \in \{0,1\}} p(y_k) H(l_j \mid y_k) \\ &= -\sum_{y_k \in \{0,1\}} p(y_k) \sum_{y_j \in \{0,1\}} p(y_j \mid y_k) \log_2 p(y_j \mid y_k) \\ &= -\sum_{y_k \in \{0,1\}} \sum_{y_j \in \{0,1\}} p(y_j, y_k) \log_2 p(y_j \mid y_k) \\ &= -\sum_{y_k \in \{0,1\}} \sum_{y_j \in \{0,1\}} p(y_j, y_k) \log_2 \frac{p(y_j, y_k)}{p(y_k)} \\ &= \sum_{y_k \in \{0,1\}} \sum_{y_j \in \{0,1\}} p(y_j, y_k) \log_2 \frac{p(y_k)}{p(y_j, y_k)} \end{aligned} \tag{11}$$

From the above equation, we obtain the following properties of the label conditional uncertainty $H(l_j \mid l_k)$:

1. $H(l_j \mid l_k)$ indicates the size of the information carried by $l_k$ to $l_j$. The larger the value is, the more the carried information is.
2. When $H(l_j \mid l_k)$ takes the minimum value, that label $l_k$ can completely predict the value of label $l_j$.
3. When $H(l_j \mid l_k)$ takes the maximum value, that label $l_k$ has no contribution on predicting the value of label $l_j$.
4. $H(l_j \mid l_k) \neq H(l_k \mid l_j)$ indicates that the conditional entropy is asymmetric. There is a difference between the uncertainty of $l_j$ given $l_k$ and the uncertainty of $l_k$ given $l_j$.

Based on the nature of the analysis $H(l_j \mid l_k)$, it is known that $H(l_j \mid l_k)$ represents the level of decision difficulty for label $l_j$ given label $l_k$ and also reflects the degree of independence of $l_j$ from $l_k$. Accordingly, for a set of labels $l_j, l_k \in L$, we use $I(l_k \rightarrow l_j)$ to evaluate the decision difficulty of the directed edges $< l_k, l_j >$ (directed edges from label $l_k$ to label $l_j$). $I(l_k \rightarrow l_j)$ is defined as follows:

$$I(l_k \rightarrow l_j) = H(l_j \mid l_k) \tag{12}$$

The maximum spanning tree is a connected graph without loops. The maximum spanning tree can be transformed into a DAG by determining the direction of each edge. For the edge $(l_j, l_k)$ in the maximum spanning tree, we calculate $I(l_k \rightarrow l_j)$ and $I(l_j \rightarrow l_k)$, then compare them and define the direction of the edge using the direction with less decision difficulty. The direction of each edge in the maximum spanning tree is determined, and finally, the maximum spanning tree is transformed into a DAG. The specific process is shown in Algorithm 2.

## Topological Sorting

We have obtained a DAG of labels in which two connected labels have an anterior-posterior ordering. To obtain the final label ordering, we use the topological ordering, which provides an efficient solution for the output vertices of the DAG. Topological ordering is commonly used to solve engineering advancement

**Algorithm 2. Transformation of maximum spanning tree into DAG**

| |
|---|
| Input: Maximum spanning tree $\mathrm{T}=(\mathrm{V}_{\mathrm{tree}},\mathrm{E}_{\mathrm{tree}})$. |
| Output: directed acyclic graph $\mathrm{DAG}=(\mathrm{V}_{\mathrm{DAG}},\mathrm{E}_{\mathrm{DAG}})$. |
| 1. Let $\mathrm{V}_{\mathrm{DAG}}=\mathrm{V}_{\mathrm{tree}}, \mathrm{E}_{\mathrm{DAG}}=\varnothing$ |
| 2. for $e(l_j,l_k)\in \mathrm{E}_{\mathrm{tree}}$ |
| 3. Calculate $I(l_k \to l_j)$ and $I(l_j \to l_k)$ |
| 4. if $I(l_k \to l_j) \le I(l_j \to l_k)$ |
| 5. $\mathrm{E}_{\mathrm{DAG}} \leftarrow \mathrm{E}_{\mathrm{DAG}} \bigcup e<l_k,l_j>$ |
| 6. else |
| 7. $\mathrm{E}_{\mathrm{DAG}} \leftarrow \mathrm{E}_{\mathrm{DAG}} \bigcup e<l_j,l_k>$ |
| 8. endfor |

problems in AOV nets, where the tasks that are ranked first are the ones that need to be completed first. In this study, in the DAG of labels, it is necessary to place the labels that have less difficulty (i.e., a greater degree of influence) on the target label decision ahead of the target label so that the label information can be delivered correctly along the label ordering. The specific algorithm is as follows:

## Time Complexity Analysis of Optimal Label Ordering

In order to optimize the label ordering of the classifier chains algorithm, we construct the relevance matrix of labels, the maximum spanning tree of labels, and the DAG of labels, respectively. And the outputs are the optimized label ordering through topological sorting. Since the Pearson correlation coefficient is symmetric, the time complexity of constructing the relevance matrix of labels is $O(q^2/2)$. The time complexity of constructing the maximum spanning tree of labels using Prim is $O(q^2)$. The time complexity of transforming the maximum spanning tree of labels into the DAG of labels is $O(q)$. The time complexity of using topological sorting to output the optimized label ordering is $O(q+e)$, where $e$ is the number of edges. The time complexity of the optimized label ordering is $O(3q^2/2+2q+e)$.

## Control Experimental Algorithm

To investigate whether building a maximum spanning tree of labels can effectively optimize label ordering and improve the final classification performance, we designed a classifier chains algorithm based on conditional entropy to construct a directed acyclic graph (CEDAGCC) as a control algorithm, which directly constructs a DAG of labels based on the mutual decision difficulty between labels.

By calculating the mutual decision difficulty $I(l_j \to l_k)$ and $I(l_k \to l_j)$ between two labels, $l_j$ and $l_k$, a directed cyclic graph (DCG) of the labels can be obtained. There are two types of links $<l_j,l_k>$ and $<l_k,l_j>$ between each pair of labels, $l_j$ and $l_k$. Their weights are $I(l_j \to l_k)$ and $I(l_k \to l_j)$, indicating the difficulty of mutual decision between $l_j$ and $l_k$. This shows that the DCG

**Algorithm 3. Output label ordering using topological sorting**

| |
|---|
| Input: directed acyclic graph $DAG = (V_{DAG}, E_{DAG})$. |
| Output: Label ordering $L_o : l_{o1}, l_{o2}, \cdots, l_{oq}$. |
| 1. Let $IN = \varnothing$, $q = \mid V_{DAG} \mid$ |
| 2. for $l_j \in V_{DAG}$ |
| 3. Calculate the entry degree $in(l_j)$ of label $l_j$ in DAG |
| 4. $IN \leftarrow IN \cup in(l_j)$ |
| 5. endfor |
| 6. for $l_j \in V_{DAG}$ do |
| 7. if $in(l_j) = 0$, $< l_j, l_k > \in E_{DAG}$, $l_k \in V_{DAG}$ |
| 8. $l_{oj} \leftarrow l_j$, $IN \leftarrow IN - in(l_j)$ |
| 9. $in(l_k) \leftarrow in(l_k) - 1$ |
| 10. $V_{DAG} \leftarrow V_{DAG} - l_j$ |
| 11. $E_{DAG} \leftarrow E_{DAG} - < l_j, l_k >$ |
| 12. endfor |

has a total of $q(q-1)$ directed edges and $q$ is the number of labels. To obtain the label ordering, the DCG of labels is converted to DAG. According to the above analysis, the smaller $I(l_j \rightarrow l_k)$ is the greater the effect of $l_j$ on $l_k$ is. Therefore, the DCG can be transformed into DAG by removing the edge in each ring of the DCG that has the greatest decision difficulty (i.e., least influence). Algorithm 4 illustrates the process of converting the DCG of labels into DAG.

The linear time complexity of this algorithm to disconnect $Cyc$ is $O(\mid Cyc \mid)$, where $\mid Cyc \mid$ denotes the total number of edges in $Cyc$. After obtaining the DAG of labels, the labels in the DAG are outputs as label ordering using Algorithm 3, and finally, this label ordering is trained and predicted using the CC algorithm.

## EXPERIMENTS

### Datasets

To verify the effectiveness of the algorithm proposed in this paper, seven datasets were selected from the publicly available multilabel dataset Mulan (Tsoumakas et al., 2011). Mulan is a Java library for learning from multilabel data and is widely used to test the performance of multilabel classifiers.

**Algorithm 4. Converting DCG to DAG**

| |
|---|
| Input: DCG=(V, E, W) |
| Output: $DAG = (V, E_{DAG}, W_{DAG})$ |
| 1. Let DAG=DCG |
| 2. while DAG has rings |
| 3. $Cyc \in DAG$ // $Cyc$ denotes a ring inside the DAG |
| 4. $E' \leftarrow \{e \mid e \in Cyc\}$ // $E'$ a denotes all weighted edges in the ring |
| 5. $e' \leftarrow \arg\max_{e \in E'} I(l_{start} \rightarrow l_{end})$ // $e'$ denotes the edge with the highest weighted value |
| 6. Remove edge $e'$ from the DAG |
| 7. endwhile |

These seven datasets are related to several domains including music, image, bioinformation, and text. Basic statistical information of the selected datasets is shown in Table 2.

Cardinality indicates the average number of labels in the sample. The calculation formula is as follows:

$$LCard(D) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{q} y_{i,j} \tag{13}$$

## Evaluation Metrics and Comparable Algorithms

Since each sample has multiple labels at the same time in multilabel classification, the common single-label evaluation metrics cannot fully and accurately evaluate the results of multilabel classification. In order to measure the advantages and disadvantages of multilabel classification algorithms we use six evaluation metrics that are widely used in multilabel classification.

**Table 2. Multilabel dataset statistics**

| Dataset | Instances | Features | Labels | Cardinality | Domain |
|---|---|---|---|---|---|
| CAL500 | 502 | 68 | 174 | 26.044 | Music |
| Birds | 645 | 260 | 19 | 1.014 | Audio |
| Scene | 2407 | 294 | 6 | 1.074 | Image |
| Enron | 1702 | 1001 | 53 | 3.378 | Text |
| Yeast | 2417 | 103 | 14 | 4.237 | Biology |
| Bibtex | 7395 | 1836 | 159 | 2.402 | Text |
| Medical | 978 | 1449 | 45 | 1.245 | Text |

$$Jaccard\ \ Similarity = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|y_i \wedge \hat{y}_i\right|}{\left|y_i \vee \hat{y}_i\right|} \tag{14}$$

Jaccard Similarity is used to compare the similarity and difference between sample sets and to evaluate the average proportion of correctly predicted labels to all labels in each sample, requiring that the predicted label sequence and the actual label sequence are identical.

$$Exact\ \ Match = \frac{1}{n}\sum_{i=1}^{n}(y_i = \hat{y}_i) \tag{15}$$

When all labels of a sample are correctly predicted the sample is correctly predicted. Exact Match indicates the percentage of correct sample prediction, and a higher value indicates a better classification.

$$F1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2p_i r_i}{p_i + r_i} \tag{16}$$

F1 indicates the composite index of classification effectiveness, which is the summed average of precision and recall of samples on the label. A higher F1 indicates better classification effectiveness.

$$macroF1 = \frac{1}{q}\sum_{j=1}^{q}\frac{2p_j r_j}{p_j + r_j} \tag{17}$$

The macro F1 score represents the weighted average of precision and recall for all labels. Higher scores indicate that the algorithm performs well on low-frequency labels.

$$microF1 = \frac{2\sum_{i=1}^{n}\sum_{j=1}^{q} y_{i,j}\hat{y}_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{q} y_{i,j} + \sum_{i=1}^{n}\sum_{j=1}^{q}\hat{y}_{i,j}} \tag{18}$$

The micro F1 focuses on the prediction of each label and is affected by false negatives and false positives. It represents the mean value of the weighted sum of precision and recall under all labels. Higher scores indicate better performance of the algorithm on high frequency labels.

$$Ranking\ \ Loss = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\left|Y_i\right|\left|\bar{Y}_i\right|}\left|\left\{(y_a, y_b) : r_i(y_a) > r_i(y_b), (y_a, y_b) \in Y_i \times \bar{Y}_i\right\}\right| \tag{19}$$

where $\bar{Y}_i$ is the complementary set of $Y_i$ with respect to the set of labels $L$ and $r_i$ denotes the ranking function. Ranking loss (Tsoumakas et al., 2011) examines the number of times when the irrelevant labels are ranked higher than the relevant ones. The smaller the ranking loss is, the higher the probability of correct ranking and the better the classification model will be.

The six evaluation metrics above measure the classification results from different perspectives. The larger the value of the first five evaluation metrics, the better the classification performance of the algorithm. The last evaluation metric, ranking loss, has a smaller value, which means that the algorithm has better classification performance.

Five related algorithms were selected for comparison with the two experimental algorithms proposed in this paper. The details are as follows:

1. BR algorithm: as a classical first-order policy algorithm, it trains a classifier for each label independently, without considering the relationship between labels.
2. 2BR algorithm: this algorithm is a stacked structure algorithm that uses the predicted labels of the first layer as extended features of the features in the second layer to exploit label correlation.
3. CC algorithm: classifier chains algorithm.
4. ECC algorithm: the algorithm improves the classification effect by training multiple classifier chains. In this paper, we uniformly set the number of classifier chains trained for each dataset to 5.
5. CEbCC algorithm: the algorithm obtains the label ordering by counting the conditional entropy between the labels and then by statistical means.
6. CEDAGCC algorithm: a controlled experimental algorithm is proposed in this paper to illustrate whether constructing a maximum spanning tree of labels can effectively utilize label relevance information.
7. maxSTCC algorithm: the proposed algorithm in this paper. The algorithm optimizes label ordering by constructing a maximum spanning tree and a directed acyclic graph.

We chose the BR algorithm, 2BR algorithm, and CC algorithm as comparison algorithms because both 2BR algorithm and CC algorithm use labels as extra features of features to solve the BR algorithm's lack of ability to utilize label correlation information. The 2BR algorithm uses all labels as extra features of features to utilize label correlation, and the CC algorithm only uses labels ranked before the target labels as extra features of features to utilize label correlation. The ECC algorithm and CEbCC algorithm, along with the CEDAGCC algorithm and maxSTCC algorithm proposed in this paper, both improve the classification performance by optimizing the label ordering of the CC algorithm.

## Experiment Setup

The experimental dataset is randomly disrupted and divided into five equal parts, four of which are selected as the training dataset, and the remaining one as the test data set. Then the experiment is conducted using five-fold cross-validation, and the mean value of the five experiments is counted as the result of one experiment.

Since the base classifier trained by all algorithms is binary, the algorithm in this study and the comparison algorithm uniformly use a linear kernel-based support vector machine (SVM) as the base classifier (Sun et al., 2014; Tsoumakas et al., 2010; Vapnik, 1996; Wang et al., 2013). The penalty factor C in SVM is a key parameter that affects its performance. When C is large it may lead to overfitting, and when C is small it may lead to underfitting. CAL500 and birds datasets were selected as experimental subjects to analyze the effect of penalty coefficient C values on the experiments. The effect of the C value on the CC algorithm is studied by adjusting the C value to vary in the range of [1E-1, 3E-2, 1E-2, 1E-3, 3E-4, 1E-4, 3E-5, 1E-5, 3E-6, 1E-6]. The proposed algorithm and the comparison algorithm in this paper both use the CC algorithm as the base. All the algorithms related to CC are optimized for label ordering, so studying the impact of C value on the CC algorithm is of generality. Figures 1 and 2 show the experimental results.

In Figures 1 and 2, macro F1, which can evaluate the multilabel classification results more comprehensively, is selected as the evaluation metric to test the effect of penalty coefficient C in the classifier SVM on the CC algorithm. In the bird's data set in Figure 1, the macro F1 evaluation metric achieves the maximum value when the C value is taken as 1E-1, indicating the best classification effect, and the minimum value when the C value is taken as 1E-4, indicating the worst classification effect. In the CAL500 dataset in Figure 2, the maximum value of the macro F1 evaluation index is obtained when

**Figure 1. Effect of C-Value in CAL500 data on the CC algorithm**

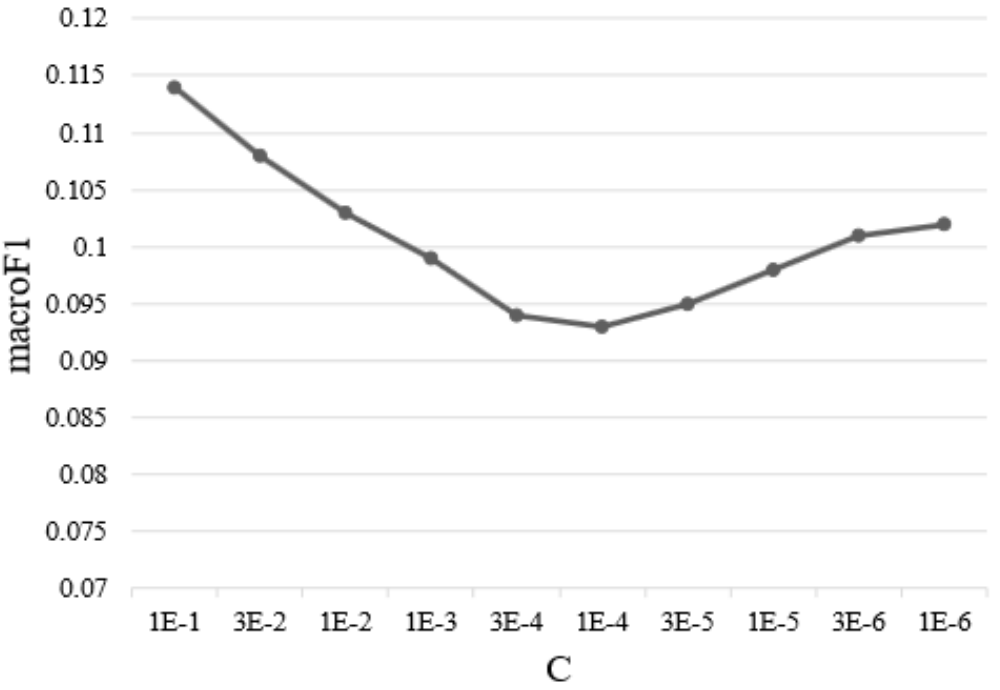
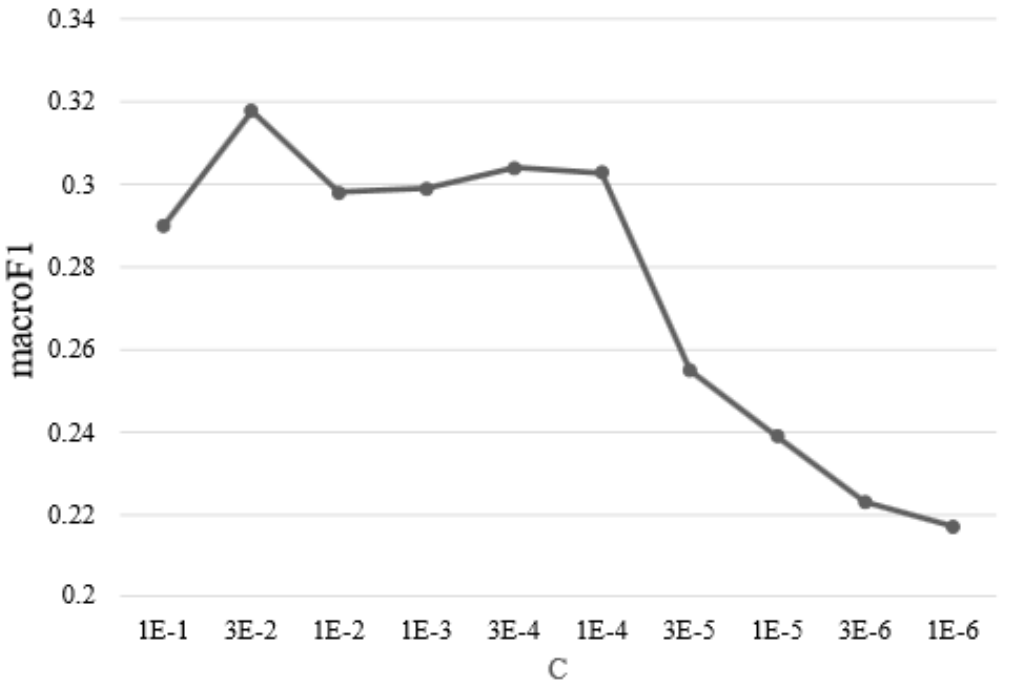



**Figure 2. Effect of C-Value in birds data on the CC algorithm**



the C value is 3E-2, which indicates the best classification effect, and the minimum value of macro F1 evaluation index is obtained when the C value is 1E-6, which indicates the worst classification effect.

Observing Figures 1 and 2, the value of penalty coefficient C directly affects the classification results of the CC algorithm, and the C values to achieve the best classification performance are different in different datasets. In order to make each algorithm obtain the best classification performance, the C values are adjusted in the range of [1E-1, 3E-2, 1E-2, 1E-3, 3E-4, 1E-4, 3E-5, 1E-5, 3E-6, 1E-6] in the experiments, and two-fold cross-validation is performed on the training set to select the C values that obtain the best validation performance.

To avoid the random effect in the experiment, we conducted 10 repetitions of the experiment for each algorithm on each dataset, and took the mean and standard deviation of each metric as the final result of the experiment. The experimental hardware and software facilities are Intel Core i7 4790 for the central processor, 8G of memory, and 64-bit Windows 10 for the operating system. All experiments presented in this paper were developed using the python language with the help of the toolkit provided by the scikit-learn platform.

## RESULTS AND DISCUSSION

Tables 3 to 8 below show the evaluation results and standard deviations of different evaluation metrics for the algorithms used in this study and the comparison algorithms on seven publicly available datasets. Where ↑ indicates that the larger the evaluation index, the better the classification effect; ↓ indicates that the smaller the value of the evaluation index, the better the classification effect. The bold in the table indicates the best evaluation result, and the numbers in small brackets indicate the ranking of the algorithms with the same evaluation criteria in the same dataset.

As seen from Tables 3 to 8, the algorithm maxSTCC achieved relatively good performance over all datasets. Among the evaluation results in the seven datasets, the maxSTCC algorithm ranked first and second with 76.2% and 16.7%, respectively.

The algorithm maxSTCC achieves optimal results on six datasets and suboptimal results on the Enron dataset in Table 3. Table 3 illustrates that the algorithms are able to maximize the prediction of the correct label category for each label. In the CAL500 dataset in Table 4, the exact match evaluation metric for all algorithms was 0, indicating that none of the samples were correctly predicted. And in the remaining six datasets, the algorithms obtained suboptimal results only on the yeast dataset. From Tables 3 and 4, the algorithms are shown to improve the accuracy of label prediction as well as the correct prediction rate of samples. In Tables 5 to 7, the maxSTCC algorithm achieves good performance on the comprehensive evaluation metrics F1, macro F,1 and micro F1 and achieves suboptimal performance on some labels only. In Table 8, the algorithm maxSTCC algorithm also achieves better results on the ranking loss evaluation index.

**Table 3. Jaccard similarity (↑) of different algorithms on seven datasets**

| Dataset | Comparison Algorithm | | | | | Proposed Algorithm | |
|---|---|---|---|---|---|---|---|
| | BR | 2BR | CC | ECC | CEbCC | CEDAGCC | maxSTCC |
| Cal500 | 0.2347±0.017(3) | 0.2262±0.013(7) | 0.2341±0.022(4) | 0.2318±0.015(5) | 0.2306±0.012(6) | 0.2350±0.011(2) | 0.2371±0.009(1) |
| Birds | 0.1736±0.023(7) | 0.1754±0.018(6) | 0.1755±0.019(5) | 0.1757±0.013(2) | 0.1756±0.015(3) | 0.1755±0.016(4) | 0.1770±0.020(1) |
| Scene | 0.6338±0.020(7) | 0.6461±0.030(6) | 0.6679±0.026(3) | 0.6676±0.017(4) | 0.6671±0.036(5) | 0.6680±0.018(2) | 0.6698±0.013(1) |
| Enron | 0.4327±0.039(7) | 0.4333±0.021(6) | 0.4487±0.021(5) | 0.4498±0.020(4) | 0.4527±0.028(1) | 0.4499±0.022(3) | 0.4511±0.023(2) |
| Yeast | 0.4978±0.018(6) | 0.4960±0.110(7) | 0.5019±0.033(5) | 0.5049±0.022(4) | 0.5071±0.036(3) | 0.5086±0.066(2) | 0.5113±0.031(1) |
| Bibtex | 0.3378±0.034(6) | 0.3377±0.026(7) | 0.3396±0.030(3) | 0.3402±0.033(2) | 0.3392±0.025(5) | 0.3395±0.022(4) | 0.3419±0.022(1) |
| Medical | 0.7415±0.059(7) | 0.7426±0.019(6) | 0.7503±0.012(5) | 0.7522±0.010(3) | 0.7516±0.019(4) | 0.7526±0.019(2) | 0.7580±0.017(1) |

**Table 4. Exact match (↑) of different algorithms on seven datasets**

| Dataset | Comparison Algorithm | | | | | Proposed Algorithm | |
|---|---|---|---|---|---|---|---|
| | BR | 2BR | CC | ECC | CEbCC | CEDAGCC | maxSTCC |
| CAL500 | 0.0000±0.000(4) | 0.0000±0.000(4) | 0.0000±0.000(4) | 0.0000±0.000(4) | 0.0000±0.000(4) | 0.0000±0.000(4) | 0.0000±0.000(4) |
| Birds | 0.4884±0.055(7) | 0.4915±0.037(5) | 0.4913±0.043(6) | 0.4933±0.030(4) | 0.4935±0.035(2) | 0.4935±0.040(3) | 0.4944±0.049(1) |
| Scene | 0.5559±0.012(7) | 0.5712±0.020(6) | 0.6194±0.019(3) | 0.6166±0.025(5) | 0.6171±0.027(4) | 0.6200±0.017(2) | 0.6218±0.013(1) |
| Enron | 0.1297±0.026(7) | 0.1311±0.019(6) | 0.1497±0.018(3) | 0.1480±0.021(5) | 0.1499±0.034(2) | 0.1495±0.023(4) | 0.1513±0.028(1) |
| Yeast | 0.1724±0.029(7) | 0.1765±0.017(6) | 0.1855±0.024(3) | 0.1911±0.029(1) | 0.1835±0.033(5) | 0.1850±0.037(4) | 0.1903±0.030(2) |
| Bibtex | 0.1635±0.017(7) | 0.1635±0.012(6) | 0.1730±0.019(5) | 0.1731±0.015(4) | 0.1733±0.020(2) | 0.1732±0.018(3) | 0.1740±0.010(1) |
| Medical | 0.7022±0.039(7) | 0.7030±0.027(6) | 0.7057±0.043(5) | 0.7073±0.050(2) | 0.7065±0.044(4) | 0.7070±0.040(3) | 0.7098±0.041(1) |

**Table 5. F1 (↑) of different algorithms on seven datasets**

| Dataset | Comparison Algorithm | | | | | Proposed Algorithm | |
|---|---|---|---|---|---|---|---|
| | BR | 2BR | CC | ECC | CEbCC | CEDAGCC | maxSTCC |
| Cal500 | 0.3699±0.017(2) | 0.3600±0.015(7) | 0.3676±0.015(4) | 0.3651±0.011(5) | 0.3635±0.010(6) | 0.3688±0.006(3) | 0.3713±0.005(1) |
| Birds | 0.2136±0.028(7) | 0.2151±0.020(4) | 0.2150±0.023(6) | 0.2166±0.024(1) | 0.2151±0.025(5) | 0.2160±0.018(3) | 0.2165±0.020(2) |
| Scene | 0.6602±0.014(7) | 0.6714±0.016(6) | 0.6870±0.017(5) | 0.6874±0.013(4) | 0.6876±0.015(3) | 0.6880±0.013(2) | 0.6900±0.015(1) |
| Enron | 0.5407±0.027(7) | 0.5420±0.025(6) | 0.5533±0.019(5) | 0.5554±0.020(3) | 0.5506±0.023(4) | 0.5601±0.019(2) | 0.5616±0.022(1) |
| Yeast | 0.6040±0.014(6) | 0.6020±0.015(7) | 0.6049±0.017(5) | 0.6074±0.015(2) | 0.6070±0.017(3) | 0.6061±0.018(4) | 0.6092±0.016(1) |
| Bibtex | 0.4059±0.018(6) | 0.4058±0.020(7) | 0.4060±0.022(5) | 0.4064±0.019(4) | 0.4068±0.023(2) | 0.4065±0.021(3) | 0.4133±0.019(1) |
| Medical | 0.7511±0.008(7) | 0.7548±0.015(6) | 0.7590±0.012(5) | 0.7604±0.010(3) | 0.7600±0.009(4) | 0.7608±0.007(2) | 0.7644±0.012(1) |

**Table 6. Macro F1 (↑) of different algorithms on seven datasets**

| Dataset | Comparison Algorithm | | | | | Proposed Algorithm | |
|---|---|---|---|---|---|---|---|
| | BR | 2BR | CC | ECC | CEbCC | CEDAGCC | maxSTCC |
| CAL500 | 0.1115±0.017(2) | 0.1065±0.018(5) | 0.1100±0.016(4) | 0.1044±0.011(7) | 0.1060±0.008(6) | 0.1111±0.010(3) | 0.1124±0.009(1) |
| Birds | 0.3132±0.052(7) | 0.3148±0.039(6) | 0.3165±0.029(4) | 0.3178±0.044(1) | 0.3160±0.043(5) | 0.3166±0.051(3) | 0.3176±0.035(2) |
| Scene | 0.7137±0.017(7) | 0.7200±0.018(6) | 0.7246±0.017(3) | 0.7254±0.015(1) | 0.7225±0.016(5) | 0.7230±0.014(4) | 0.7246±0.016(2) |
| Enron | 0.1899±0.022(7) | 0.1913±0.019(5) | 0.1915±0.014(3) | 0.1917±0.022(2) | 0.1914±0.019(4) | 0.1910±0.013(6) | 0.1920±0.017(1) |
| Yeast | 0.3542±0.006(5) | 0.3480±0.008(7) | 0.3509±0.007(6) | 0.3599±0.004(3) | 0.3569±0.005(4) | 0.3602±0.009(2) | 0.3609±0.008(1) |
| Bibtex | 0.3285±0.033(7) | 0.3285±0.024(6) | 0.3286±0.025(5) | 0.3287±0.020(4) | 0.3293±0.019(2) | 0.3288±0.020(3) | 0.3300±0.018(1) |
| Medical | 0.2714±0.005(7) | 0.2722±0.008(6) | 0.2735±0.004(5) | 0.2740±0.006(3) | 0.2738±0.007(4) | 0.2741±0.009(2) | 0.2753±0.005(1) |

From Tables 3 to 8, the 2BR algorithm, CC algorithm, ECC algorithm, CEbCC algorithm, and the CEDAGCC algorithm and the maxSTCC algorithm, which consider label correlation, improve the classification results compared with the BR algorithm, which does not consider label correlation at all. This indicates that using label correlation can improve the classification results of multilabel classification algorithms. The 2BR algorithm uses the predicted labels of the first layer as input features for the second layer features to exploit label correlations. If the first layer classifier predicts incorrect labels, it may introduce incorrect label correlations in the second layer, thus training a second layer classifier with poor performance and ultimately leading to poor classification results. It is observed from the following tables that the 2BR algorithm is only superior to the BR algorithm as a whole. The

**Table 7. Micro F1 (↑) of different algorithms on seven datasets**

| Dataset | Comparison Algorithm | | | | | Proposed Algorithm | |
|---|---|---|---|---|---|---|---|
| | BR | 2BR | CC | ECC | CEbCC | CEDAGCC | maxSTCC |
| CAL500 | 0.3735±0.008(3) | 0.3633±0.012(7) | 0.3723±0.014(5) | 0.3725±0.013(4) | 0.3714±0.015(6) | 0.3730±0.009(2) | 0.3754±0.011(1) |
| Birds | 0.4300±0.041(7) | 0.4395±0.026(3) | 0.4380±0.036(6) | 0.4397±0.035(2) | 0.4394±0.028(4) | 0.4390±0.022(5) | 0.4401±0.019(1) |
| Scene | 0.7088±0.017(7) | 0.7154±0.018(6) | 0.7157±0.018(5) | 0.7193±0.016(2) | 0.7168±0.013(4) | 0.7188±0.015(3) | 0.7194±0.022(1) |
| Enron | 0.5588±0.022(7) | 0.5588±0.019(6) | 0.5667±0.017(1) | 0.5660±0.014(3) | 0.5663±0.017(2) | 0.5654±0.024(5) | 0.5660±0.022(4) |
| Yeast | 0.6238±0.006(7) | 0.6260±0.008(6) | 0.6266±0.007(5) | 0.6295±0.003(3) | 0.6275±0.005(4) | 0.6300±0.005(2) | 0.6318±0.008(1) |
| Bibtex | 0.4393±0.013(6) | 0.4391±0.025(7) | 0.4402±0.027(4) | 0.4424±0.019(1) | 0.4400±0.021(5) | 0.4411±0.021(3) | 0.4420±0.020(2) |
| Medical | 0.7786±0.005(7) | 0.7790±0.008(6) | 0.7799±0.007(5) | 0.7812±0.009(3) | 0.7809±0.008(4) | 0.7830±0.003(2) | 0.7839±0.004(1) |

**Table 8. ranking loss(↓) of different algorithms on seven datasets**

| Dataset | Comparison Algorithm | | | | | Proposed Algorithm | |
|---|---|---|---|---|---|---|---|
| | BR | 2BR | CC | ECC | CEbCC | CEDAGCC | maxSTCC |
| Cal500 | 0.7189±0.038(3) | 0.7264±0.027(7) | 0.7174±0.034(2) | 0.7239±0.021(6) | 0.7234±0.019(5) | 0.7190±0.022(4) | 0.7110±0.015(1) |
| Birds | 0.3153±0.019(2) | 0.3146±0.011(1) | 0.3160±0.016(7) | 0.3156±0.018(5) | 0.3157±0.017(6) | 0.3155±0.015(3) | 0.3156±0.015(4) |
| Scene | 0.3208±0.010(7) | 0.3207±0.013(6) | 0.3102±0.015(2) | 0.3104±0.019(3) | 0.3108±0.017(4) | 0.3110±0.016(5) | 0.3093±0.016(1) |
| Enron | 0.4912±0.015(7) | 0.4909±0.017(6) | 0.4774±0.019(3) | 0.4778±0.011(4) | 0.4790±0.019(5) | 0.4750±0.012(2) | 0.4712±0.022(1) |
| Yeast | 0.4578±0.018(6) | 0.4603±0.017(7) | 0.4510±0.016(5) | 0.4484±0.014(3) | 0.4506±0.017(4) | 0.4480±0.015(2) | 0.4474±0.010(1) |
| Bibtex | 0.5997±0.020(7) | 0.5976±0.022(6) | 0.5949±0.025(1) | 0.5956±0.022(3) | 0.5960±0.021(4) | 0.5965±0.019(5) | 0.5955±0.015(2) |
| Medical | 0.2221±0.009(7) | 0.2210±0.010(6) | 0.2200±0.014(5) | 0.2193±0.009(3) | 0.2195±0.006(4) | 0.2177±0.012(2) | 0.2150±0.011(1) |

CC algorithm, ECC algorithm, and CEbCC algorithm use real labels as input features for the training phase, avoiding the drawbacks of the 2BR algorithm, and thus the classification results are improved.

## Average Rank

We calculated the average ranking of these algorithms and comparison algorithms in order to show the experimental effects visually. Figure 3 shows the average ranking of these seven algorithms on the six metrics. Figure 4 shows the average ranking of these seven algorithms on the seven datasets. Overall, the proposed algorithm maxSTCC achieves the best ranking, which verifies the effectiveness of our proposed method. The control algorithm CEDAGCC proposed in this paper achieves the second ranking on accuracy, exact match, F1, micro F1 and ranking loss, and the third ranking on macro F1 only. This shows that exploring dependencies between labels by exploiting the decision difficulty between labels and building DAG of labels can utilize relevance. And the superior achievement of the maxSTCC algorithm affirms the contribution of constructing a maximum spanning tree of labels in utilizing label relevance.

## Friedman Test

The Friedman test (Friedman, 1940) is used to analyze whether there is a significant difference between the individual algorithms. The Friedman statistic value $F_F$ is calculated as follows:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{20}$$

**Figure 3. Average ranking of seven algorithms on six metrics**

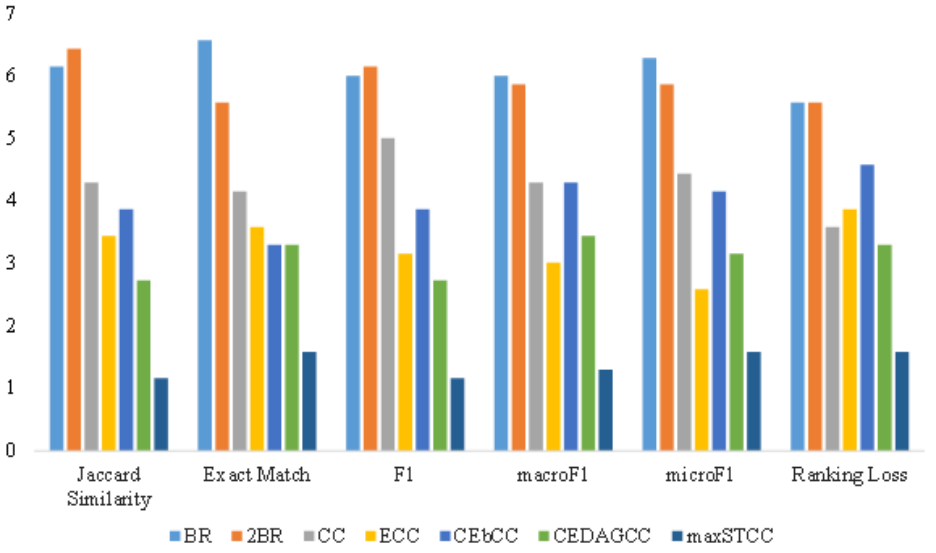
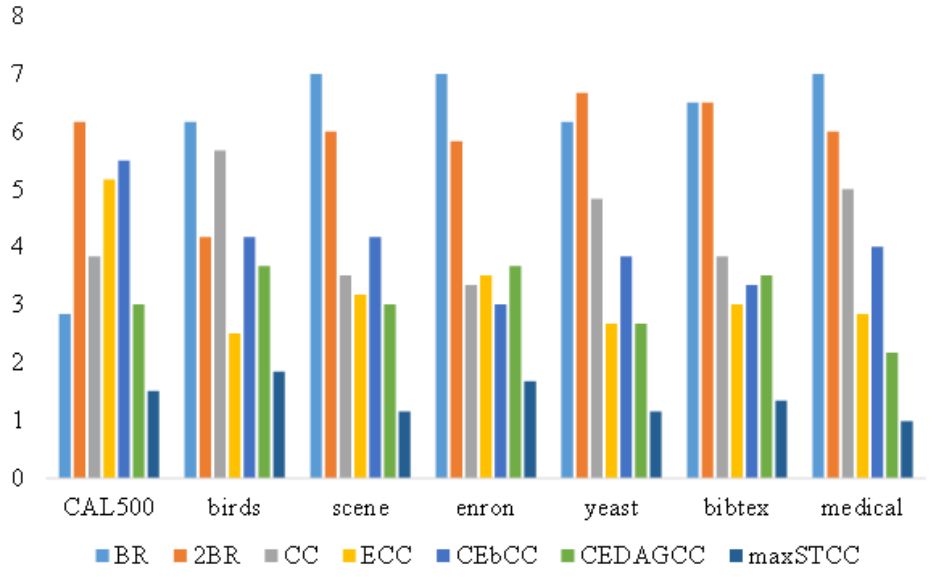



**Figure 4. Average ranking of seven algorithms on seven datasets**




where $\chi_F^2$ is calculated as

$$\chi_F^2 = \frac{12N}{k(k+1)}\left(\sum_{j=1}^{k} R_j^2 - \frac{(k(k+1)^2)}{4}\right) \tag{21}$$

where $R_j = 1/N \cdot \sum_{i=1}^{N} r_i^j$ denotes the average rank of the *j*th algorithm among the comparison algorithms. In this paper, $N = 7$ and $k = 7$ denotes the number of datasets and the number of all algorithms, respectively. By consulting the table of the Friedman test critical values, the critical value of $F\left(k-1,\left(k-1\right)\times\left(N-1\right)\right) = F\left(6,36\right)$ for rejecting the null hypothesis at significant level $\alpha = 0.05$ is 2.364. When $F_F$ is larger than the critical value of 2.364, it indicates a significant difference in the classification performance of the seven algorithms. The Friedman test results are shown in Table 9.

### Stability Analysis of the maxSTCC Algorithm

In terms of algorithm stability, the classifier chains and its improvement algorithms are prone to unstable performance when the label dimension of the dataset is too high. The maxSTCC algorithm differs from the CC algorithm in randomly selecting the label ordering, but it explores the dependency information among labels by constructing the maximum spanning tree and DAG of labels and then obtains a more stable label ordering. Thus, it has a more stable classification performance.

To effectively illustrate the stability of the maxSTCC algorithm, two datasets, CAL500 and bibtex, which have 174 and 159 labels, respectively, were selected for observation. As shown in Tables 3 to 8, the stability (standard deviation) of CEDAGCC algorithm and maxSTCC algorithm proposed in this paper achieves better results compared with the CC algorithm, ECC algorithm, and CEbCC algorithm. In particular, the maxSTCC algorithm maximizes the utilization of correlation information among labels by constructing the maximum spanning tree of labels, which can further improve the stability of label ordering compared with the CEDAGCC algorithm. Therefore, the performance of maxSTCC algorithm is more stable.

## CONCLUSION AND FUTURE WORK

In this paper, we propose a new multilabel classification algorithm (maxSTCC), which improves the classification performance of the CC algorithm by building a maximum spanning tree of labels and transforming it into a directed acyclic graph to obtain a better label ordering. The maxSTCC algorithm has the following main contributions: 1) using the Pearson correlation coefficient to measure the degree of correlation between labels and taking the absolute value to consider the positive correlation and negative correlation, 2) constructing a maximum spanning tree of labels to maximize the utilization of the correlation information between labels, and 3) using conditional entropy to define the mutual decision difficulty between two related labels and using the direction with less decision difficulty as the dependency direction of these two labels to solve the ranking problem between two related labels.

**Table 9. Friedman test for seven algorithms**

| Metrics | $F_F$ | Critical Value( $\alpha = 0.05$ ) |
|---|---|---|
| Jaccard Similarity | 19.40 | 2.364 |
| Exact Match | 8.06 | |
| F1 | 15.0 | |
| Macro F1 | 9.54 | |
| micro F1 | 9.65 | |
| Ranking Loss | 4.32 | |

The experimental results show that the maxSTCC algorithm effectively optimizes the label ordering and improves the classification effect of the CC algorithm with more stable performance and stronger competitiveness than other related algorithms. To illustrate the effect of constructing the maximum spanning tree of labels on the maxSTCC algorithm, a control experimental algorithm CEDAGCC was designed in this study, which directly calculates the decision difficulty between labels to construct the directed acyclic graph of labels. The experimental results of CEDAGCC algorithm and maxSTCC algorithm affirm that the maximum spanning tree of labels effectively utilizes the correlation information between labels.

Future research on this topic may include the following: 1) the directed acyclic graph of labels has multiple topological orderings, and it would be interesting to study the effect of different topological orderings on the classification performance; 2) this paper describes that the base classifier of the maxSTCC algorithm is SVM, and it would be interesting to discuss the effect of different base classifiers on the maxSTCC algorithm; 3) we used mutual decision difficulty between labels to solve the ranking problem between two related labels, and applying this method to other multilabel classification algorithms should be the focus of future work.

## AUTHOR NOTE

The authors declare that there is no conflict of interest regarding the publication of this paper.

This work was supported by the National Natural Science Foundation of China (No. 12063002).

The data used to support the findings of this study are included within the article.

## REFERENCES

Boutell, M. R., Luo, J., Shen, X., & Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, *37*(9), 1757–1771. doi:10.1016/j.patcog.2004.03.009

Cheng, W., Hüllermeier, E., & Dembczynski, K. J. (2010). Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. Semantic Scholar.

Elisseeff, A., & Weston, J. (2001). A kernel method for multi-labelled classification. *Advances in Neural Information Processing Systems*, 14.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, *11*(1), 86–92. doi:10.1214/aoms/1177731944

Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., & Brinker, K. (2008). Multilabel classification via calibrated label ranking. *Machine Learning*, *73*(2), 133–153. doi:10.1007/s10994-008-5064-8

Godbole, S., & Sarawagi, S. (2004). Discriminative methods for multi-labeled classification. In *Proceedings of Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference (PAKDD 2004).* Springer Berlin Heidelberg. doi:10.1007/978-3-540-24775-3_5

Guan, R., Wang, X., Yang, M. Q., Zhang, Y., Zhou, F., Yang, C., & Liang, Y. (2018). Multi-label deep learning for gene function annotation in cancer pathways. *Scientific Reports*, *8*(1), 267. doi:10.1038/s41598-017-17842-9 PMID:29321535

Huang, J., Li, G., Wang, S., Zhang, W., & Huang, Q. (2015). Group sensitive classifier chains for multi-label classification. In *Proceedings of the 2015 IEEE International Conference on Multimedia and Expo (ICME).* IEEE. doi:10.1109/ICME.2015.7177400

Jiaman, D., Shujie, Z., Runxin, L., Xiaodong, F., & Lianyin, J. (2022). Association rules-based classifier chains method. *IEEE Access : Practical Innovations, Open Solutions*, *10*, 18210–18221. doi:10.1109/ACCESS.2022.3149012

Jun, X., Lu, Y., Lei, Z., & Guolun, D. (2019). Conditional entropy based classifier chains for multi-label classification. *Neurocomputing*, *335*, 185–194. doi:10.1016/j.neucom.2019.01.039

Lanchantin, J., Wang, T., Ordonez, V., & Qi, Y. (2021). General multi-label image classification with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* IEEE. doi:10.48550/arXiv.2011.14027

Liu, H., Chen, G., Li, P., Zhao, P., & Wu, X. (2021). Multi-label text classification via joint learning from label embedding and label correlation. *Neurocomputing*, *460*, 385–398. doi:10.1016/j.neucom.2021.07.031

Markatopoulou, F., Mezaris, V., & Patras, I. (2018). Implicit and explicit concept relations in deep neural networks for multi-label video/image annotation. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(6), 1631–1644. doi:10.1109/TCSVT.2018.2848458

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys*, *54*(3), 1–40. doi:10.1145/3439726

Nam, J., Kim, J., Loza Mencía, E., Gurevych, I., & Fürnkranz, J. (2014). Large-scale multi-label text classification: Revisiting neural networks. In *Proceedings of Machine Learning and Knowledge Discovery in Databases: European Conference (ECML PKDD 2014).* Springer Berlin Heidelberg. doi:10.1007/978-3-662-44851-9_28

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, *85*(3), 333–359. doi:10.1007/s10994-011-5256-5

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, *33*(1-2), 1–39. doi:10.1007/s10462-009-9124-7

Sinhashthita, W., & Jearanaitanakij, K. (2020). Improving KNN algorithm based on weighted attributes by Pearson correlation coefficient and PSO fine tuning. In *Proceedings of the 2020-5th International Conference on Information Technology (InCIT)*. IEEE. . 2020.9310938 doi:10.1109/InCIT50588.2020.9310938

Sun, F., Tang, J., Li, H., Qi, G. J., & Huang, T. S. (2014). Multi-label image categorization with sparse factor representation. *IEEE Transactions on Image Processing*, *23*(3), 1028–1037. doi:10.1109/TIP.2014.2298978 PMID:24474372

Tiple, B., & Patwardhan, M. (2022). Multi-label emotion recognition from Indian classical music using gradient descent SNN model. *Multimedia Tools and Applications*, *81*(6), 8853–8870. doi:10.1007/s11042-022-11975-4

Tsoumakas, G., Dimou, A., Spyromitros, E., Mezaris, V., Kompatsiaris, I., & Vlahavas, I. (2009). Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Proceedings of the 1st International Workshop on Learning From Multi-Label Data*. Semantic Scholar.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2010). Mining multi-label data. *Data Mining and Knowledge Discovery Handbook*, 667-685. 10.1007/978-0-387-09823-4_34

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., & Vlahavas, I. (2011). Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, *12*, 2411–2414. https://mulan.sourceforge.net/datasets.html

Vapnik, V. (1999). *The nature of statistical learning theory*. Springer Science & Business Media.

Wang, R., Kwong, S., Chen, D., & Cao, J. (2013). A vector-valued support vector machine model for multiclass problem. *Information Sciences*, *235*, 174–194. doi:10.1016/j.ins.2013.02.001

Wang, R., Ye, S., Li, K., & Kwong, S. (2021). Bayesian network based label correlation analysis for multi-label classifier chain. *Information Sciences, 554*, 256-275. https://doi.org/. 12.01010.1016/j.ins.2020

Zaragoza, J. C., Sucar, E., Morales, E., Bielza, C., & Larranaga, P. (2011). Bayesian chain classifiers for multidimensional classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence.* Research Gate.

Zhang, M. L., & Zhou, Z. H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, *40*(7), 2038–2048. doi:10.1016/j.patcog.2006.12.019

Zhang, M. L., & Zhou, Z. H. (2013). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *26*(8), 1819–1837. doi:10.1109/TKDE.2013.39

Zhu, F., Li, H., Ouyang, W., Yu, N., & Wang, X. (2017). Learning spatial regularization with image-level supervisions for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. doi:10.1109/CVPR.2017.219