# Multi-Label Learning Based on Transfer Learning and Label Correlation

**Kehua Yang[1, *], Chaowei She[1], Wei Zhang[1], Jiqing Yao[2] and Shaosong Long[1]**

**Abstract:** In recent years, multi-label learning has received a lot of attention. However, most of the existing methods only consider global label correlation or local label correlation. In fact, on the one hand, both global and local label correlations can appear in real-world situation at same time. On the other hand, we should not be limited to pairwise labels while ignoring the high-order label correlation. In this paper, we propose a novel and effective method called GLLCBN for multi-label learning. Firstly, we obtain the global label correlation by exploiting label semantic similarity. Then, we analyze the pairwise labels in the label space of the data set to acquire the local correlation. Next, we build the original version of the label dependency model by global and local label correlations. After that, we use graph theory, probability theory and Bayesian networks to eliminate redundant dependency structure in the initial version model, so as to get the optimal label dependent model. Finally, we obtain the feature extraction model by adjusting the Inception V3 model of convolution neural network and combine it with the GLLCBN model to achieve the multi-label learning. The experimental results show that our proposed model has better performance than other multi-label learning methods in performance evaluating.

**Keywords:** Bayesian networks, multi-label learning, global and local label correlations, transfer learning.

## 1 Introduction

Nowadays, we live in an information age. An instance cannot be labeled with just single label, so the instance is often associated with more than one class label. For example, an image can be annotated with several labels [Su, Chou, Lin et al. (2011)], a piece of music can belong to many types [Turnbull, Barrington, Torres et al. (2008)], a text can reflect different themes [Ueda and Saito (2002)]. Therefore, multi-label classification attracts more and more researchers to research.

There are two categories in multi-label learning algorithms [Zhang and Zhou (2007)]: problem transformation and algorithm adaption. Problem transformation is a straightforward method. The main idea is to convert multi-label problem as one or more traditional single label problems. Algorithms include Binary Relevance (BR) [Boutell,

---

[1] College of Computer Science and Electronic Engineering and Key Laboratory for Embedded and Network Computing of Hunan Province, Hunan University, Changsha, 410082, China.

[2] Oath Verizon Company, Manhattan, New York, 10007, USA.

[*] Corresponding Author: Kehua Yang. Email: khyang@hnu.edu.cn.

Luo, Shen et al. (2004)], Pruned Problem Transformation (PPT) [Read, Pfahringer and Holmes (2009)] and so on. Algorithm adaptation is an adaptive method. The main idea is to use single-label classification algorithm to adapt to multi-label classification. Classic algorithms include C4.5 Decision Tree [Quinlan (1992)], Multi-label Dimensionality reduction via Dependence Maximization (MDDM) [Zhang and Zhou (2010)], Multi-label Informed Latent Semantic Indexing (MLSL) [Yu, Yu and Tresp (2005)] and so on.

Label correlation can provide important information for multi-label classification. For example, "blue sky" and "white cloud" are highly symbiotic labels, while "sunny" and "black clouds" are highly exclusive labels. "ocean" and "sailboat" appear at the same time, it is highly likely that the "fish" label will be included, while the "desert" label will not appear. However, most of the existing methods mainly focus on the sharing characteristics of global label and ignore the label correlation of local data sets. For example, "Jack Ma" is associated with "Alibaba" in the IT company data set [Liu, Peng and Wang (2018)], but it is weakly related to global label correlation. Therefore, according to the above analysis, it is more practical and comprehensive to consider the global and local label correlations in multi-label classification.

Each instance has characteristics of multi-dimensional label in multi-label learning. If the label of instance is annotated simply by manual labeling, human may sometimes ignore labels that they do not know or of little interest, or follow the guide by some algorithm to reduce labeling costs [Huang, Chen and Zhou (2015)]. Some labels may be missing from the training set, which is a kind of weakly supervised learning. So subjectivity factors are unavoidable in the labels. As a result, some labels may be missing from the data set, resulting in label imbalance, which makes it more difficult and potentially negatively impacting performance to estimate label correlations.

In this paper, we propose a novel and effective method called "Bayesian Networks with Global and Local Label Correlation" (GLLCBN). The main idea of GLLCBN is to use the global label semantics correlation and local label correlation of data set to balance label correlation and reduce the impact of label noise on data set. First of all, the probability of each individual label is obtained by analyzing the data set. Similarly, we get the probability between pairwise labels by using the data set. And then, the global label correlation matrix is constructed by label semantic similarity. After that, according to the relevant probability information received in the first to three steps, the initial Bayesian networks topology is constructed to obtain the high-order label correlation. In addition, redundant edge (label correlation) in the network structure are optimized by graph theory and probability theory. Subsequently, GLLCBN model is constructed. Finally, the initial prediction label is obtained by using transfer learning to adjust and train the Inception V3 model, and then, the prediction result is combined with GLLCBN to achieve multi-label classification.

The remainder of this paper is organized as follows. Section 2 introduces the related work of multi-label learning. Section 3 presents our proposed algorithm in detail. We experimented to verify the performance of our proposed method in Section 4. Finally, conclusions and future work are given in Section 5.

## 2 Related work

In recent years, multi-label learning has been extensively studied and many methods have been proposed [Zhang and Zhou (2007)]. In addition, the role of label correlation has gradually become the focus of researchers. Methods can be divided into three categories according to the degree of label correlation [Zhang and Zhang (2010)].

First-order method is to convert a multi-label classification into multiple one-dimensional independent classifiers. For example, the classic BR [Boutell, Luo, Shen et al. (2004)] trained a corresponding classifier for each label independently. Obviously, the advantage of this approach is its simplicity, but it ignores label correlation. Second-order method refers to the correlation between pairs of labels. For example, the CLR [Brinker (2008)] achieved conversion of multi-label classification problems by analyzing the correlation of pairwise labels and establishing label rankings. Although the advantage of this method is that it considers the internal pairwise label correlations, which has a certain efficiency improvement. However, multi-label learning generally has high dimensions, we should not be limited only to consider the existence of pairwise labels. Therefore, higher-order methods are proposed. High-order method refers to analyzing the correlation between the high-dimensional of the labels and is not limited to the pairwise labels. For example, ML-LRC [Huang and Zhou (2012)] solves multi-label classification problem by using characteristics of the matrix rank. Obviously, the advantage of high-order method is to extract the intrinsic connection of label and strengthen the dependency of labels, but label correlation analysis is more difficult and the label correlation structure is more complicated.

Labeling of an instance may result in label imbalance due to subjectivity factors. For example, the actual label for this image should contain "bull", "mountain" and "road" in Fig. 1. By manual labeling, on the one hand, the picture on the left can be marked in the order of "cattle", "mountain", "road". On the other hand, the picture on the right may be marked as "mountain", "road", "bull". Sometimes the label of "bull" even be lost by visual effects. GLOCAL [Zhu, Kwok and Zhou (2017)] indicated that missing label and label order are influential factors for multi-label classification.
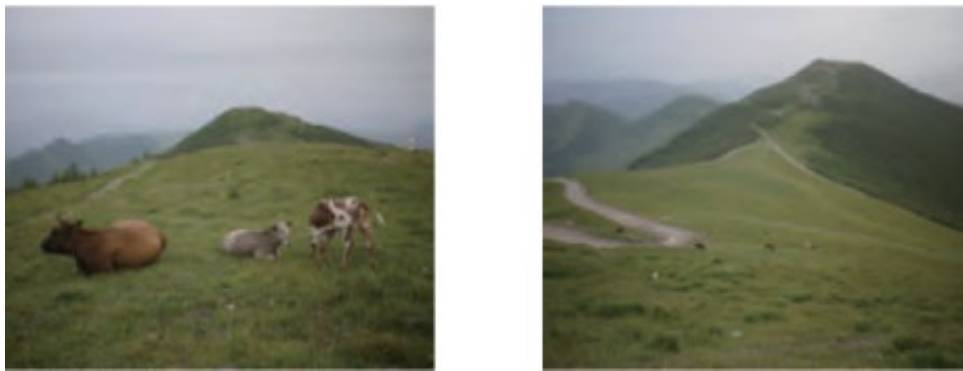


**Figure 1:** Image annotation

In summary, for the study of multi-label classification, not only global label correlation should be considered, but also local label correlation. Therefore, a more balanced and

comprehensive label correlation can be received.

## 3 The proposed approach

In this section, details of the proposed approach GLLCBN will be presented. Firstly, we perform predefined of model and analysis global and local label correlation to obtain GLLCBN model. Secondly, we combine the optimized Inception V3 [Szegedy, Vanhoucke, Ioffe et al. (2016)] model by transfer learning with GLLCBN to achieve multi-label classification.

### *3.1 Preliminaries*

Since multi-label classification has high-dimensional features, this is the difference between multi-label classification and single-label classification, we have following predefined processing. Let $D = R^n$ be n-dimensional sample space and $L_i = \{l_1, l_2, ..., l_m\}$, where $m$ is the number of labels in dataset. On the one hand, the correspondence between the data set instance and the sample label is defined as $Q = \{(N_i, M_i) \mid i = 1, 2, ..., n\}$, where $n$ represents the total number of data set and $N_i \in D$ is an n-dimensional feature vector. So we define $N_i = [N_i^1, N_i^2, ..., N_i^n]$ to represent the feature vector of a sample instance. On the other hand, we denote $M = [M_1, M_2, ..., M_n]^T \in \{-1, 1\}^{n \times m}$ as sample label matrix, where $M_i = [l_i^1, l_i^2, ..., l_i^m]$ is the label vector of instance associated with $N_i$. In addition, we denote $l_i^1, l_i^2, ..., l_i^m \in \{-1, 1\}$ as each element $l_i^j = 1$ if the $i$-th instance has $j$-th label, otherwise $l_i^j = -1$.

### *3.2 Label correlation*

Label correlation contains potentially important information for multi-label classification problem, so label correlation is an essential part of our analysis [Punera, Rajan and Ghosh (2005)]. However, there are certain difficulties in the analysis of this aspect, then how to solve this problem has become a new research direction. In order to analyze label correlation more reasonably and comprehensively, we deal with local correlation of the data set and global correlation of the label semantics.

### *3.2.1 Local label correlation*

We consider local label correlation from the data set. Since data in the data set is random, the probability of different labels is inconsistent. According to this feature, we denote $P = [p(l_1), p(l_i), ..., p(l_m)]$ as the probability of each label occurrence, where $m$ represents the total number of sample labels and $p(l_i)$ indicate the probability of the $i$-th label in the data set. Since label correlation is at least second-order, we need to calculate the probability of pairwise labels. The local label correlation is defined as:

$$p(l_i \mid l_j) = \frac{T(N_{l_i \mid l_j})}{T(N_{l_j})} = X_{(l_i \mid l_j)} \in [0,1] \tag{1}$$

where $i, j$ are a single label in the label set and $i, j \in m$. We denote $T(N_{l_j})$ as the number of sample instances with the label $l_j$. However , to avoid anomalous expressions, if $T(N_{l_j})$ value is equal to 0, it means that $p(l_i \mid l_j)$ is also equal to 0. Similarly, $T(N_{l_i \mid l_j})$ represents the number of sample instances that simultaneously have both labels $l_i$ and $l_j$. In addition, we denote $p(l_i \mid l_j)$ and $X_{(l_i \mid l_j)}$ as the probability of pairwise label correlations. It is important to note that pairwise label correlations is not a symmetric equivalent relationship, which is defined as:

$$p(l_i \mid l_j) \neq p(l_j \mid l_i) \tag{2}$$

For example, there is a data set as shown in Tab. 1:

**Table 1:** Data set

| Instance order of data set | Label of data set(A,B,C,D,E) |
|:---:|:---:|
| 1 | 1，-1，-1，1，-1 |
| 2 | 1，1，1，-1，-1 |
| 3 | -1，1，1，1，-1 |
| 4 | 1，1，-1，-1，1 |
| 5 | 1，-1，1，-1，-1 |

According to the above table, $p(l_A \mid l_B) = \frac{2}{3}$, and $p(l_B \mid l_A) = \frac{2}{4} = \frac{1}{2}$, there is $p(l_A \mid l_B) \neq p(l_B \mid l_A)$, so Eq. (2) is correct.

*3.2.2 Global label correlation*

We obtain global correlation by analyzing the word similarity. At present, the correlation between words mainly uses context semantics of words, the word vector is used to judge correlation between two words and Word2vec [Mikolov, Chen, Corrado et al. (2013)] is a classic algorithm. For example, words "man" and "woman" are highly relevant to "man" and "beautiful", because they are used in a similar context. For example the word "man" can be used in the position of the sentence and the word "woman" can be replaced. Therefore, we define $W = [W_1, W_2, ..., W_m]^T \in [0,1]^{m \times m}$ as word matrix, where $W_1 = [w(l_1 \mid l_1), w(l_1 \mid l_2), ..., w(l_1 \mid l_m)]$ is a vector of pairwise words correlation and $w(l_i \mid l_j)$ is the word correlation probability between labels $i$ and $j$. The process is defined as:

$$W = \begin{bmatrix} 1 & w(l_1 \mid l_2) & \dots & \dots & w(l_1 \mid l_m) \\ w(l_2 \mid l_1) & 1 & w(l_2 \mid l_3) & \dots & w(l_2 \mid l_m) \\ \dots & \dots & 1 & \dots & \dots \\ \dots & \dots & \dots & 1 & \dots \\ w(l_m \mid l_1) & w(l_m \mid l_2) & \dots & \dots & 1 \end{bmatrix} \tag{3}$$

As shown in Eq. (3), each label is perfectly correlated with itself, so the value is 1 and a small value means that the label correlation is low, otherwise the opposite.

### 3.3 GLLCBN model

According to the analysis in Section 3.2, we have dealt with global and local label correlations. The relationship between them is defined as Eq. (4):

$$E(l_i \mid l_j) = \lambda_1 p(l_i \mid l_j) + \lambda_2 w(l_i \mid l_j) \tag{4}$$

where $i$, $j$ are pairwise labels, and $\lambda_1, \lambda_2 \in [0,1]$, $\lambda_1 + \lambda_2 = 1$ are trade-off parameters for controlling the weight between global and local label correlations. Then that, $E(l_i \mid l_j)$ is the comprehensive label correlation.

It is not enough to finally acquire pairwise label correlations through the $E(l_i \mid l_j)$, because according to the global and local label correlation, this will result in a cyclic relationship between pairwise labels, the labels $l_i$ and $l_j$ have a relationship between $E(l_i \mid l_j)$ and $E(l_j \mid l_i)$. When a symmetrical relationship occurs, it becomes ambiguous because it is impossible to determine which side of the pairwise labels is strongly dependent on the other. Therefore, in order to solve this problem, it is necessary to eliminate the ambiguous dependencies of pairwise labels, so that the definition can be defined as Eq. (5):

$$E(l_{x_1} \mid l_{x_2}) = \max(E(l_i \mid l_j), E(l_j \mid l_i)) \tag{5}$$

where $l_i, l_j$ are pairwise labels, and $l_{x_1} \mid l_{x_2}$ is finalized label dependency.

For example, there is a structure in Fig. 2, where a circle represents a label (e.g., A, B) and the edges between the circles represent the probability of pairwise label correlations (e.g., $E(l_B \mid l_A)$, $E(l_A \mid l_B)$).
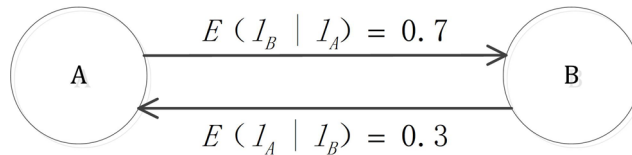


**Figure 2:** Correlation between label A and B

According to the description of the Eq. (5), since correlation follow the principle of maximum value, correlation between label A and label B should be such that the

structure of Fig. 2 should be optimized to the structure of Fig. 3. Therefore, this shows that label A is more dependent on label B.
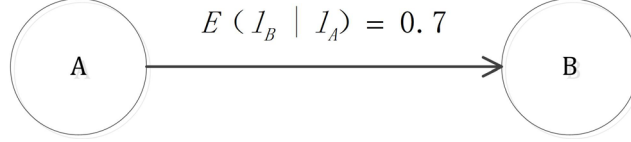


**Figure 3:** Optimized label A and B correlation

As shown in Fig. 4, it is worth noting that if there are multiple reachable paths for one label to another. As shown in the Eq. (6), it is used to determine label dependencies of multiple reachable edges.
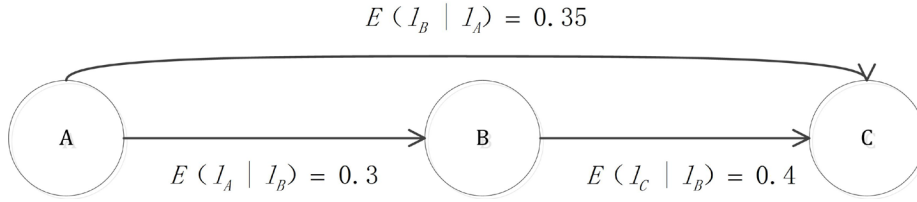


**Figure 4:** Multiple reachable paths between label

$$E(l_i \mid l_j) = \max(E(l_i \mid l_j), E(l_j \mid l_{k1} \mid l_{k2}...\mid l_i)) \tag{6}$$

Eq. (6) can be equivalent to the form as shown in the Eq. (7):

$$E(l_i \mid l_j) = \begin{cases} E(l_i \mid l_j), & Q_1 \geq Q_2 \\ E(l_j \mid l_{k1} \mid l_{k2},...,\mid l_i), & Q_1 < Q_2 \end{cases} \tag{7}$$

where $k1, k2, ...$ are the middle label node on the path from label $i$ to $j$. Then that, $Q_1 = \lambda_1 \dfrac{N(i,j)}{N(j)} + \lambda_2 w_{l_i \mid l_j}$ and $Q_2 = \lambda_1 \dfrac{N(i,k_1,...,j)}{N(k_1,k_2...,j)} + \lambda_2(w(l_{k_1} \mid l_j) \times w(l_{k_2} \mid l_{k_1}) \times,...,\times w(l_i \mid l_{k_n}))$. The specific proof of the Eq. (7) is as follows:

**Proof.** First, we are based on Fig. 4. On the one hand, N(A), N(B), N(C), N(A,B), N(A,C), N(B,C), N(A,B,C) are the number of sample instances. On the other hand, $p(l_C \mid l_A)$, $p(l_B \mid l_A)$, $p(l_C \mid l_B)$ are the probability of pairwise local label correlations. Then, $w(l_B \mid l_A)$, $w(l_C \mid l_A)$, $w(l_C \mid l_B)$ are the probability of label semantic correlation. Moreover, their values are known according to the analysis from Eq. (1) to Eq. (5).

Second, according to graph theory and probability theory, we have the following defines:

$$E(l_C \mid l_A) = \lambda_1 p(l_C \mid l_A) + \lambda_2 w(l_C \mid l_A) \tag{8}$$

$$(E(l_B \mid l_A), E(l_C \mid l_B)) = \lambda_1 p(l_{ABC} \mid l_{AB}) + \lambda_2 w(l_{ABC} \mid l_{AB}). \tag{9}$$

If we have to prove $E(l_C \mid l_A) \geq (E(l_B \mid l_A), E(l_C \mid l_B))$, according to Eq. (8) and Eq. (9),

we only need to prove $\lambda_1 \dfrac{N(A,C)}{N(A)} + \lambda_2 w(l_C \,|\, l_A) \geq \lambda_1 \dfrac{N(ABC)}{N(AB)} + \lambda_2 w(l_{ABC} \,|\, l_{AB})$.

We observe that the data set can be able to guarantee $N(A,C) \geq N(A,B,C)$, because both of them belong to the inclusion relationship, and the former has a larger scope. By the same logic, we know that $N(A) \geq N(A,B)$ and the values of $\lambda_1$ and $\lambda_2$ on both sides of the equation are the same, so it does not require additional consideration. In addition, $w(l_{ABC} \,|\, l_{AB}) = w(l_B \,|\, l_A) \times w(l_C \,|\, l_B) \times (l_C)$, where $w(l_C)$ is equal 1, thus,

$$w(l_{ABC} \,|\, l_{AB}) = w(l_B \,|\, l_A) \times w(l_C \,|\, l_B).$$

According to the above analysis, we know that only if the sample data set satisfies the value of $\lambda_1 \dfrac{N(A,C)}{N(A)} + \lambda_2 w(l_C \,|\, l_A) \quad \geq \quad \lambda_1 \dfrac{N(ABC)}{N(AB)} + \lambda_2 (w(l_B \,|\, l_A) \times w(l_C \,|\, l_B))$,

then $E(l_C \,|\, l_A) \geq (E(l_B \,|\, l_A), E(l_C \,|\, l_B))$ can be obtained, otherwise, the opposite is true. Therefore，we prove that Eq. (7) is true.

According to the Eq. (7), the reachable path for eliminating the dependency of the label correlation can be performed, but the following two cases require special handling.

Case 1. $E(l_i \,|\, l_j) = E(l_i \,|\, l_{k1} \,|\, l_{k2} ... \,|\, l_j)$

According to the principle of maximum label correlation, Fig. 4 is optimized as shown in Fig. 5.
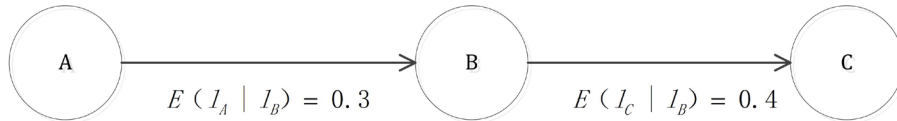


**Figure 5:** Optimized GLLCBN model

Case 2. $E(l_i \,|\, l_j) = E(l_i \,|\, l_j)$

Fig. 4 does not need to be changed. Because there may be pairwise label correlations, the intermediate nodes in them cannot be eliminated, and the correlation structure of each intermediate node should be retained.

In summary, directed graph model of GLLCBN can be constructed by analyzing label correlation and Bayesian networks [Friedman, Linial, Nachman et al. (2000)]. The GLLCBN model can be used to optimize the label correlation and facilitate the extraction of potential association information between labels, and reduce the impact of label imbalances in the sample data set.

### 3.4 Adjustment of Inception V3 model

Convolutional Neural Network (CNN) plays a very important role in the research of image classification [Song, Hong, Mcloughlin et al. (2017)]. There are many excellent models of CNN, such as AlexNet [Krizhevsky, Sutskever and Hinton (2012)], VGGNet [Russakovsky, Deng, Su et al. (2015)], ResNet [He, Zhang, Ren et al. (2015)] and

GoogleNet [Szegedy, Liu, Jia et al. (2014)]. Among them, Inception V3 [Szegedy, Vanhoucke, Ioffe et al. (2016)] created by Google is a very portable and highly usable model. Therefore, we use transfer learning approach to make related adjustments for the Inception V3 model to improve the performance of the multi-label classification problem.

In order to adjust the Inception V3 model, we need to make some adjustments. First of all, since the Inception V3 model was initially trained for single classification, but our images are multi-label attributes, we need to treat the label storage of the input data as multi-dimensional, rather than only as single label. Secondly, in order to achieve applicability, it is generally necessary to remove the top-level structure and then add some new various layers of customization. Therefore, we add a fully connected layer of 1024 nodes for association with the last pooling layer. Finally, since the softmax in Inception V3 outputs 1000 nodes (the ImageNet [Deng, Dong, Socher et al. (2009)] data set has 1000 categories), we need to modify the last layer of the network and convert it to the number of nodes (it equivalent to the label type in the data set) so that label classification is achieved through our model.

## 4 Experiments

In this section, to evaluate the performance of GLLCBN, a description of the multi-label data set used in the experiments, the performance evaluation of multi-label classification and comparative algorithm with GLLCBN model are explained. Finally, the experimental results and analysis are presented.

### 4.1 Data sets

In order to verify the performance of GLLCBN, we chose the open source data set collected by Nanjing University. The download link for the data set is *http://lamda.nju.edu.cn/files/miml-image-data.rar*. The data set contains 2,000 landscape images and five labels (desert, ocean, sunset, mountains, trees). In addition, each instance has an average of two labels. It is worth noting that the original data set archive contains a file called miml_data with a .mat suffix (Matlab file format). It contains three files: bags.mat, targets.mat and class_name.mat. The first file can be ignored directly because it has no special effect. The second file is the label definition for each image, which means a matrix of 5×2000, each column represents label for an instance image, and the value 1 indicates the presence of the label, and -1 means that there is no corresponding label, and the label order is the same as the class_name.mat file. In addition, we need to deal with label format content and convert the matrix into a .txt format file in order to facilitate follow-up training of Inception V3 model. The last file shows all possible label names for the data set.

### 4.2 Performance evaluation

Since an instance has multiple label attributes in the multi-label classification, prediction label may belong to a subset of the actual label, which is represented as $L_{prediction} \in L_{actual}$.

To evaluate the performance of GLLCBN model, we select five widely-used evaluation metrics [Gibaja and Ventura (2015)].

Hamming Loss expresses the degree of inconsistency between the prediction label and the actual label. Eq. (10) shows the expression of the Hamming Loss.

$$Hamming\ Loss = \frac{1}{N}\sum_{i=1}^{N}\frac{XOR(Y_{i,j},P_{i,j})}{L} \tag{10}$$

Coverage evaluates how far it is needed, on average, to go down the ranked list of labels in order to cover all ground true labels. Eq. (11) shows the expression of the Coverage.

$$Coverage = \frac{1}{Q}(\frac{1}{N}\sum_{i=1}^{N}\max_{l_k\in y_i} rank(x_i,l_k)-1) \tag{11}$$

Ranking Loss evaluates the average fraction of mis-ordered label pairs. Ranking Loss is defined as follows：

$$Ranking\ Loss = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\left\{(l_k,l_j)\,|\,f_k(x_i)\le f_j(x_i),(l_k,l_j)\in y_i\times\bar{y}_i\right\}\right|}{|y_i\|\bar{y}_i|} \tag{12}$$

Average Precision represents the average accuracy of the predicted instance label set, just like Eq. (13).

$$Average\ Precision = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\left\{l_j|rank(x_i,l_j)\le rank(x_i,l_k),l_j\in y_i\right\}\right|}{rank(x_i,l_k)} \tag{13}$$

Average Predicted Time expresses the average time to predict each instance and it time unit is second, which is expressed as follows:

$$Average\ Predicted\ Time = \frac{T_n-T_0}{N} \tag{14}$$

It is worth noting that in the above five performance evaluation, Hamming Loss, Ranking Loss, Coverage, Ranking Loss and Average Predicted Time, the smaller value means the better performance. But for Average Precision, the larger value means better performance.

### 4.3 Comparative algorithms

In order to validate the validity of GLLCBN model, we compare GLLCBN to the following most advanced multi-label learning algorithms:

1. Binary Relevance (BR) [Boutell, Luo, Shen et al. (2004)] is first-order method. The main idea is to train a binary linear SVM classifier independently for each label.

2. Calibarated Label Ranking (CLR) [Brinker (2008)] is second-order method. The main idea is to establish a label ranking by analyzing the pairwise labels.

3. Multi-label Learning Using Local Correlation (ML-LOC) [Huang and Zhou (2012)] is high-order method. The main idea is to analyze the local label correlation by encoding instance features.

4. Random K-Labelsets (RAKEL) [Tsoumakas, Katakis and Vlahavas (2011)] is high-order method. The main idea is to transform the multi-label classification problem

into several multi-class learning problems by exploiting high-order global label correlation.

All compared algorithms are summarized in Tab. 2.

**Table 2:** Compared methods

| Method | Label Correlation | Degree of label correlation |
|--------|-------------------|----------------------------|
| BR | NO | First-order |
| CLR | YES | Second-order |
| ML-LOC | YES | High-order |
| RAKEL | YES | High-order |
| GLLCBN | YES | High-order |

### 4.4 Experimental results

In our experiments, we randomly use 30%, 50% and 70% of the data in data set as the training set, and the rest of data as test data set. Experimental results are shown from Tab. 3 to Tab. 5. In addition, all of our experimental methods are studied by using the Python or Matlab environment.

**Table 3:** Performance evaluation of different algorithms for randomly marking 30% data sets as training data sets: mean±std(rank)

| Algorithms | Hamming Loss | Cove rage | Ranking Loss | Average Precision | Average Predicted Time |
|------------|--------------|-----------|--------------|-------------------|------------------------|
| BR | **0.1624±0.0124(1)** | **0.2925±0.0213(1)** | **0.2905±0.0202(1)** | **0.6724±0.0211(1)** | **0.5292±0.0123(1)** |
| CLR | 0.1763±0.0154(2) | 0.3634±0.0201(3) | 0.3129±0.0297(2) | 0.5781±0.0215(3) | 0.6914±0.0105(3) |
| ML-LOC | 0.1846±0.0136(4) | 0.3683±0.0198(4) | 0.3356±0.0268(4) | 0.5221±0.0298(4) | 0.7117±0.0113(4) |
| RAKEL | 0.1835±0.0145(3) | 0.3612±0.0214(2) | 0.3149±0.0191(3) | 0.6138±0.0256(2) | 0.6721±0.0121(2) |
| GLLCBN | 0.1958±0.0187(5) | 0.3968±0.0287(5) | 0.3905±0.0208(5) | 0.4923±0.0249(5) | 1.0302±0.0103(5) |

**Table 4:** Performance evaluation of different algorithms for randomly marking 50% data sets as training data sets: mean±std(rank)

| Algorithms | Hamming Loss | Cove rage | Ranking Loss | Average Precision | Average Predicted Time |
|---|---|---|---|---|---|
| BR | **0.1678±0.0076(1)** | 0.2698±0.0181(5) | 0.2791±0.0159(5) | 0.6902±0.0145(5) | **0.5131±0.0116(1)** |
| CLR | 0.1698±0.0168(2) | 0.2431±0.0124(4) | 0.2324±0.0188(4) | 0.7171±0.0192(4) | 0.7105±0.0122(3) |
| ML-LOC | 0.1741±0.0121(3) | 0.2312±0.0119(2) | 0.2223±0.0109(2) | **0.7328±0.0176(1)** | 0.7141±0.0109(4) |
| RAKEL | 0.1824±0.0119(4) | **0.2223±0.0102(1)** | **0.2149±0.0145 (1)** | 0.7323±0.0125（2） | 0.6924±0.0117(2) |
| GLLCBN | 0.1867±0.0121(5) | 0.2401±0.0123(3) | 0.2305±0.0157(3) | 0.7309±0.0118(3) | 1.1165±0.0149(5) |

**Table 5:** Performance evaluation of different algorithms for randomly marking 70% data sets as training data sets: mean±std(rank)

| Algorithms | Hamming Loss | Cove rage | Ranking Loss | Average Precision | Average Predicted Time |
|---|---|---|---|---|---|
| BR | 0.1692±0.1328(3) | 0.2608±0.0179(5) | 0.2705±0.0218(5) | 0.6988±0.0212(5) | **0.5023±0.0169 (1)** |
| CLR | **0.1663±0.0154(1)** | 0.2125±0.0188(4) | 0.1926±0.0196(4) | 0.7303±0.0186(4) | 0.6924±0.0122(2) |
| ML-LOC | 0.1725±0.0126(4) | 0.1983±0.0142(2) | 0.1723±0.0123(2) | 0.7784±0.0098(2) | 0.7212±0.0143(4) |
| RAKEL | 0.1752±0.0098(5) | 0.2012±0.0108(3) | 0.1749±0.0123(3) | 0.7693±0.0123(3) | 0.6935±0.0153（3） |
| GLLCBN | 0.1689±0.0069(2) | **0.1943±0.0114(1)** | **0.1705±0.0162(1)** | **0.7909±0.0104(1)** | 1.1102±0.0121(5) |

## 4.5 Experimental analysis

According to experimental results in Section 4.4, we get the following summary:

1.  When the training data set is 30% of the data set, BR algorithm has some advantages in the performance evaluation of Hamming Loss, Coverage, Ranking Loss, Average Precision and Average Prediction Time. Global label correlation algorithm is superior to local label correlation algorithm.

2.  When the training data set is 50% of the data set, BR algorithm has better performance evaluation in Hamming Loss than the label correlation algorithm. In terms of Coverage, Ranking Loss, Average Precision and Average Prediction Time, the CLR,

ML-LOC and GLLCBN algorithms that consider local label correlation have a better advantage than the RAKEL algorithm that considers global label correlation.

3.  When the training data set is 70% of the data set, the advantages of BR are gradually replaced by other algorithms. CLR algorithm in Hamming Loss is higher than other algorithms. For the label correlation algorithm, GLLCBN has better performance evaluation in terms of Coverage, Ranking Loss and Average Precision than ML-LOC and RAKEL algorithms, but it is longer in Average Prediction Time.

4.  By analyzing the above three points, we can know that when the training data set is gradually increased, the advantage of the label correlation algorithm in the performance evaluation is gradually reflected, indicating that the label correlation has a certain influence on the multi-label classification problem.

## 5 Conclusion and future work

For the study of multi-label classification, how to mine the potential label correlation information is still a worthy direction in the future. In this paper, we propose a novel and effective approach named GLLCBN for multi-label learning. In the GLLCBN model, the node represents label space, and edge represents global and local comprehensive label correlation. We firstly obtain a complex model by analyzing label, global semantic relevance and local label correlation of data set (This process is called building node association graph), and secondly by using probability theory, Bayesian networks and graph theory to optimize label dependency graph (This process is called eliminating redundant edges), thus we construct a label-dependent network called GLLCBN model. Finally, the multi-label classification is solved by combining the initial prediction results by the Inception V3 model with the GLLCBN model. In addition, experimental results show that our proposed approach has certain effectiveness in performance evaluation.

In the future, we consider optimizing the performance of our proposed methods in large scale label space data sets and applying this approach to more different multi-label data sets.

## References

**Boutell, M. R.; Luo, J.; Shen, X.; Brown, C. M.** (2004): Learning multi-label scene classification. *Pattern Recognition*, vol. 37, no. 9, pp. 1757-1771.

**Brinker, K.** (2008): Multilabel classification via calibrated label ranking. *Machine Learning*, vol. 73, no. 2, pp. 133-153.

**Friedman, N.; Linial, M.; Nachman, I.; Pe, D.** (2000): Using bayesian networks to analyze expression data. *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 601-620.

**Deng, J.; Dong, W.; Socher, R.; Li, L. J.** (2009): Imagenet: a large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248-255.

**Gibaja, E.; Ventura, S.** (2015): A tutorial on multilabel learning. *ACM Computing Surveys*, vol. 47, no. 3, pp. 1-38.

**He, K.; Zhang, X.; Ren, S.; Sun, J.** (2015): Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778.

**Huang, S. J.; Chen, S.; Zhou, Z. H.** (2015): Multi-label active learning: query type matters. *International Conference on Artificial Intelligence*, pp. 946-952.

**Huang, S. J.; Zhou, Z. H.** (2012): Multi-label learning by exploiting label correlations locally. *26th AAAI Conference on Artificial Intelligence*, pp. 949-955.

**Krizhevsky, A.; Sutskever, I.; Hinton, G. E.** (2012): Imagenet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems*, pp. 1097-1105.

**Liu, Y.; Peng, H.; Wang, J.** (2018): Verifiable diversity ranking search over encrypted outsourced data. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 37-57.

**Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J.** (2013): Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111-3119.

**Punera, K.; Rajan, S.; Ghosh, J.** (2005): Automatically learning document taxonomies for hierarchical classification. *Special Interest Tracks and Posters of the International Conference on World Wide Web*, pp. 1010-1011.

**Quinlan, J. R.** (1992): C4.5: programs for machine learning. *Morgan Kaufmann Publishers*, vol. 1.

**Read, J.; Pfahringer, B.; Holmes, G.** (2009): Multi-label classification using ensembles of pruned sets. *IEEE International Conference on Data Mining*, pp. 995-1000.

**Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S. et al.** (2015): Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252.

**Song, Y.; Hong, X.; Mcloughlin, I.; Dai, L.** (2017): Image classification with cnn based fisher vector coding. *IEEE Conference on Visual Communications and Image Processing*, pp. 1-4.

**Su, J. H.; Chou, C. L.; Lin, C. Y.; Tseng, V. S.** (2011): Effective semantic annotation by image-to-concept distribution model. *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 530-538.

**Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S. et al.** (2014): Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.

**Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z.** (2016): Rethinking the inception architecture for computer vision. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818-2826.

**Tsoumakas, G.; Katakis, I.; Vlahavas, I.** (2011): Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge & Data Engineering*, vol. 23, no. 7, pp. 1079-1089.

**Turnbull, D.; Barrington, L.; Torres, D.; Lanckriet, G.** (2008): Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio Speech & Language Processing*, vol. 16, no. 2, pp. 467-476.

**Ueda, N.; Saito, K.** (2002): Parametric mixture models for multi-labeled text. *International Conference on Neural Information Processing Systems*, pp. 737-744.

**Yu, K.; Yu, S.; Tresp, V.** (2005): Multi-label informed latent semantic indexing. *International ACM Sigir Conference on Research & Development in Information Retrieval*, pp. 258-265.

**Zhang, M. L.; Zhang, K.** (2010): Multi-label learning by exploiting label dependency. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 999-1008.

**Zhang, M. L.; Zhou, Z. H.** (2007): ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recognition*, vol. 40, no. 7, pp. 2038-2048.

**Zhang, Y.; Zhou, Z. H.** (2010): Multi-label dimensionality reduction via dependence maximization. *ACM Transactions on Knowledge Discovery from Data*, vol. 4, no. 3, pp. 1-21.

**Zhu, Y.; Kwok, J. T.; Zhou, Z. H.** (2017): Multi-label learning with global and local label correlation. *IEEE Transactions on Knowledge & Data Engineering*, vol. 1, no. 99, pp. 1.