A deep neural network based hierarchical multi-label classification method

Cite as: Rev. Sci. Instrum. 91, 024103 (2020); doi: 10.1063/1.5141161
Submitted: 4 December 2019 • Accepted: 19 January 2020 •
Published Online: 25 February 2020

View Online Export Citation CrossMark

Shou Feng,[1,a)] Chunhui Zhao,[1,b)] and Ping Fu[2,c)]

AFFILIATIONS

[1] College of Information and Comminication Engineering, Harbin Engineering University, Harbin 150001, China
[2] School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150080, China

[a)] fengshou@hrbeu.edu.cn
[b)] **Author to whom correspondence should be addressed:** zhaochunhui@hrbeu.edu.cn
[c)] fuping@hit.edu.cn

ABSTRACT

With the accumulation of data generated by biological experimental instruments, using hierarchical multi-label classification (HMC) methods to process these data for gene function prediction has become very important. As the structure of the widely used Gene Ontology (GO) annotation is the directed acyclic graph (DAG), GO based gene function prediction can be changed to the HMC problem for the DAG of GO. Due to HMC, algorithms for tree ontology are not applicable to DAG, and the accuracy of these algorithms is low. Therefore, existing algorithms cannot satisfy the requirements of gene function prediction. To solve this problem, this paper proposes a DAG hierarchical multi-label classification algorithm, C2AE-DAGLabel algorithm. The C2AE-DAGLabel algorithm uses the Canonical Correlated AutoEncoder (C2AE) model as the classifier and designs a DAGLabel algorithm to solve the DAG hierarchical constraint problem. The DAGLabel algorithm can improve the classification accuracy by ensuring that the classification results meet the requirements of the hierarchical constraint. In the experiment, human gene data annotated with GO are used to evaluate the performance of the proposed algorithm. The experimental results show that compared with other state-of-the-art algorithms, the C2AE-DAGLabel algorithm has the best performance in solving the hierarchical multi-label classification problem for DAG.

I. INTRODUCTION

Gene and protein function annotation is a very concerned issue in the field of biology. During the work of gene function annotation, determining the function of a gene requires extensive biological experiments and various biological instruments. These biological instruments produce a lot of data, and processing these biological data manually is time-consuming and laborious. Gene function prediction is to find out functions that a gene may have through some methods. With the increase in the size of the data generated by biological experiments and instruments, how to use machine learning and data mining methods to accelerate gene function prediction has been a research hot spot,[1] which improves the efficiency of processing biological instrumental data and accelerates the task of the gene function prediction.

As a gene may have many functions, from the perspective of machine learning and data mining, gene function prediction can also be treated as hierarchical multi-label classification (HMC) problem. In the early stage, the Munich Information Center for Protein Sequences (MIPS) annotation scheme was used for gene function annotation, which used a tree structure to organize and manage gene function information. As the tree structure is too simple to organize the large amount of different functions of genes, the Gene Ontology (GO) annotation scheme whose ontology is directed acyclic graph (DAG) is widely used in the field of gene function. Therefore, the problem of gene function prediction using GO annotation schemes can be transformed as the DAG hierarchical multi-label classification problem.[2]

In hierarchical multi-label classification, a label set has a certain hierarchical structure, and labels conform to a predefined hierarchy. The hierarchy defines the membership relationship between different labels, which makes these labels no longer independent.[3] In a hierarchy, if an instance belongs to a particular class, then it must belong to all ancestor classes of that class, and if an instance does not belong to a particular class, then it must not belong to all descendants of that class, which is also called the hierarchical constraint. The existence of hierarchy increases the complexity of the multi-label classification problem, so it is more difficult to solve the hierarchical multi-label classification problem than the plat multi-label classification problem. In general, the hierarchical structure can be divided into tree and directed acyclic graph (DAG). The tree structure is relatively simple, which is studied by most literature and research studies at present, mainly aiming at the problem of fault diagnosis, automatic speech recognition, and image classification.

The DAG is more complex, and each node except the root can have more than one parent node. Therefore, the hierarchical multi-label classification algorithms designed for the tree structure are not applicable for the directed acyclic graph. In addition, there are only a few studies on hierarchical multi-label classification for DAG, and these algorithms cannot meet the actual needs in the field of processing instrumental data for gene function prediction.

In this study, a new DAG hierarchical multi-label classification algorithm is proposed, which is called the C2AE-DAGLabel algorithm. This algorithm can process the biological instrument data for gene function prediction from the view of DAG hierarchical multi-label classification. The C2AE-DAGLabel algorithm first transfers the hierarchical multi-label classification problem into a set of simple binary classification problems. This process simplifies and facilitates the effective solution of the complex HMC problem and makes the algorithm more flexible and practical. The Canonical Correlated AutoEncoder (C2AE), which is designed based on the deep neural network (DNN) based model, is used as the base classifier. By taking advantage of its high precision, the accuracy of binary classifiers can be guaranteed, and then, the overall classification accuracy of the C2AE-DAGLabel algorithm can be improved. How to ensure the prediction results according to the hierarchical structure is a tricky problem for hierarchical multi-label classification. In order to solve this problem, a hierarchical constraint algorithm, DAGLabel, is designed and integrated into the C2AE-DAGLabel algorithm. The DAGLabel algorithm can calculate the optimal classification results, and ensure that the prediction results meet the requirements of hierarchical constraints at the same time.

The remainder of this paper is structured as follows. The related work is discussed in Sec. II. Then, we elaborate the proposed method in detail in Sec. III. The experimental results are provided in Sec. IV. Finally, we conclude the paper in Sec. V.

II. RELATED WORKS

How to design a hierarchical multi-label classification algorithm is a complex problem, which has been presented in many applications. According to the involved hierarchy, either the entire or the partial, these algorithms can be categorized as a local classification method and global classification method.

The local classification method is training one or more classifiers for each layer of the hierarchy. The local classification method can decompose the global problem of hierarchical multi-label classification into multiple local problems, and, then, classify it through multiple classifiers. However, due to each classifier being trained and used independently, there is a need to design an algorithm to further process the results of these classifiers; otherwise, it cannot meet the requirements of the hierarchical constraint.

The global classification method is to design only one classifier to complete the classification by considering the entire information of hierarchy during the training. It can directly get the classification results that meet the requirements of the hierarchy constraint. However, the design of the global classifier is difficult.

Generally speaking, the global classification method and local classification method have their own advantages. Therefore, which strategy should be adopted in the design of the algorithm depends on the specific situation.[4] According to these two ideas, researchers have designed some algorithms for DAG hierarchical multi-label classification, and some typical local classification methods are introduced briefly in the following.

Cheng *et al.* studied the prediction of gene function and considered the hierarchical constraint between labels.[5] In the training stage, it selects the naive Bayes classifier and SVM as the basic classifier and trains a classifier for each node in the hierarchy. In the prediction stage, it used a top-down method to classify from top to bottom based on the hierarchy. For a node in the hierarchy, if the result given by the classifier is positive, the sub-node classifier of this node will continue to classify downward. If the result is negative, the classification will be terminated on this path. Through this top-down method, the prediction result can meet the requirement of the hierarchical constraint.

Mallinali *et al.* proposed a chain path evaluation (CPE) method to process hierarchical multi-label classification.[6] Through experiments, it selects the random forest, which has the best effectiveness as the classifier model to predict the gene function of *saccharomyces cerevisiae*. The classification results were compared with similar algorithms under various evaluation indexes. The experimental result shows that the CPE method has a better effectiveness in the hierarchical multi-label classification.

Cerri *et al.* proposed the HMC-LMLP algorithm to solve the hierarchical multi-label classification problem of the tree diagram.[7] This algorithm designs a multi-layer perceptron for each layer in the hierarchy of the tree diagram. It can predict which nodes in this layer the tested sample belongs to, that is, which labels the tested sample belongs to. The HMC-LMLP algorithm is compared with several similar algorithms, and the result shows that HMC-LMLP is an effective hierarchical multi-label classification algorithm.

Zeng *et al.* constructed a mathematical model based on the tree diagram for the problem of hierarchical multi-label classification in fault diagnosis.[8] It analyzes and summarizes the contextual misclassification information, designs the loss function for the tree diagram, transforms the original problem into an optimization problem, and then tries to solve the hierarchical constraint problem. Meanwhile, because there is expert knowledge as a reference in fault diagnosis, the author introduces priori knowledge to improve the accuracy.

In addition to the above local classification methods, there are also a few typical global classification methods.

Vens *et al.* proposed a global algorithm, CLUS-HMC algorithm, to solve the hierarchical multi-label classification problem of the directed acyclic diagram.[9] In the training stage, it uses the decision inductive tree method to generate a decision tree for the whole hierarchy. The author uses the weighted Euler distance to calculate the standard deviation of the classification, and the weight of each node is related to the depth in the hierarchy. This algorithm considers global information and considers all nodes uniformly, so as to ensure that the classification results can meet the requirements of hierarchical constraints.

Alves *et al.* introduced the concept of the artificial immune system into the hierarchical multi-label classification and proposed the MHC-AIS algorithm.[10] It considers the both ideas of local classification and global classification so that the MHC-AIS algorithm includes the version based on the local method and the version based on the global method. The version of the global method designs a classifier for all nodes by considering the overall structure of gene function labels. The version of the local method designs a classifier for each node in the structure of gene function labels. The author used two versions of this algorithm for gene function label prediction experiments, respectively. The results show that compared with the local algorithm, the global algorithm has a better hierarchical recall rate, but relatively low about the hierarchical accuracy.

In summary, these studies basically cover the details of hierarchical multi-label classification and have achieved some research results, which also points out the direction for future research and development. Current research results also show that in the field of gene function prediction, using the hierarchical multi-label classification algorithm to predict gene function is a feasible and valuable method.

Through the analysis, we can also see that the accuracy of existing algorithms is low, and there is a lack of theoretical analysis on DAG hierarchical multi-label classification, and these algorithms cannot meet the requirements of gene function prediction.

III. THE PROPOSED METHOD

In this section, we propose a hierarchical multi-label classification algorithm, C2AE-DAGLabel. This algorithm can be considered as a method of local approach, and it is the realization of the mathematical model for solving the DAG hierarchical multi-label classification problem. This mathematical model transforms the DAG hierarchical multi-label classification problem into a set of binary classification problems, and obtains the final prediction result by solving an optimization problem. Therefore, the C2AE-DAGLabel algorithm trains a classifier for each node in the DAG by utilizing the C2AE model and uses a DAGLabel algorithm to obtain the final results and ensure that the final results are consistent with the DAG hierarchy.

In order to better illustrate the algorithm designed, the mathematical model is introduced first in this section, and then, the details of the C2AE-DAGLabel algorithm are described.

A. A mathematical model for DAG HMC

Assume that $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \in \mathcal{X}$ is the feature vector of an instance from the d-dimensional feature space $\mathcal{X}$ and $\mathcal{L} = \{l_0, l_2, \ldots, l_{N-1}\}$ is a predefined class label set. $\mathbf{y}_i = (y_{i0}, y_{i1}, \ldots, y_{iN-1})$ is the N-dimensional class label vector, where $y_{ij} \in \{0, 1\}$. If the instance belong to the jth class, $y_{ij} = 1$; otherwise, $y_{ij} = 0$. Define that $\tilde{y_{ij}}$ is $\tilde{y_{ij}} = 1 - y_i j$. So, when $y_{ij} = 1$, $\tilde{y_{ij}} = 0$ and when $y_{ij} = 0$, $\tilde{y_{ij}} = 1$.

H is a predefined hierarchy, which conforms to the DAG structure. In the hierarchy H, each node i corresponds to the label l_i. For each node $i \in H$, define $par(i)$ and $child(i)$, respectively, as the parent node set and the child node set of node i. The root node has no parent nodes; generally, we denote it by 0 and its label by l_0. Similarly, the ancestor node set and descendant node set of a node i are denoted $anc(i)$ and $desc(i)$, respectively. Therefore, for an instance $\mathbf{x}_i$, its real label at node i can be written as y_{ii}.

As the labels conform to a certain hierarchical structure H, there will be a hierarchy constraint among labels. The hierarchy constraint is defined as that if a instance belongs to a particular class, then it must belong to all ancestor classes of that class, and if a instance does not belong to a particular class, then it must not belong to all descendants of that class.

In other words, if node i in the hierarchy H is labeled positive, its root node or its parent nodes must be labeled positive. If node i is labeled negative, its child nodes must be labeled negative either. The mathematical form of the hierarchical constraint of DAG can be expressed as

$$\begin{cases} y_i = 1 \Rightarrow \{i = 0 \cup \mathbf{y}_{par(i)} = 1\}, \\ y_i = 0 \Rightarrow \mathbf{y}_{desc(i)} = 0. \end{cases} \tag{1}$$

For an instance $\mathbf{x}$, we denote its prediction label vector by $\hat{\mathbf{y}}$ and the true label vector by $\mathbf{y}$. In a hierarchical multi-label classification problem, given a node i in hierarchy H, the contextual misclassification information depends on whether the parent nodes of i are misclassified when a misclassification error occurs in node i. Through analysis, there may be four kinds of errors between node i and its parent nodes. Taking all the contextual misclassification into consideration, we propose a novel loss function for DAG hierarchical multi-label classification, DAGH loss function. In this function, each item is corresponding to a case of error, and it can be written as follows:

$$\begin{aligned} L_{DAGH}(\hat{\mathbf{y}}, \mathbf{y}) = {} & w_1 \sum_{i=1}^{N-1} C_i y_i \tilde{\hat{y}}_i \prod_{j \in par(i)} y_j \hat{y}_j + w_2 \sum_{i=1}^{N-1} C_i y_i \tilde{\hat{y}}_i \\ & \times \sum_{j \in par(i)} y_j + w_3 \sum_{i=1}^{N-1} C_i \tilde{y}_i \hat{y}_i \prod_{j \in par(i)} y_j \hat{y}_j \\ & + w_4 \sum_{i=1}^{N-1} C_i \tilde{y}_i \hat{y}_i \sum_{j \in par(i)} \tilde{y}_j, \end{aligned} \tag{2}$$

where w_1, w_2, w_3, and w_4 are weight constants that represent the weight of different errors in the loss function. For a DAG hierarchy, as a node may have many parent nodes, C_i is expressed as

$$C_i = \begin{cases} 1, & i = 0, \\ \sum_{j \in par(i)} \frac{C_j}{|child(i)|}, & i > 0. \end{cases} \tag{3}$$

The Bayesian decision theory is a basic method to implement decision under known probability, which is one of the basic methods of statistical machine learning. Bayesian decision theory can make the optimal decisions based on the known probability and misclassification loss.

We denote the misclassification loss by $L_{DAGH}(\hat{\mathbf{y}}, \mathbf{y})$ and the conditional probability that $\mathbf{y}$ holds given $\mathbf{x}$ by $P(\mathbf{y}|\mathbf{x})$. Based on the misclassification loss and the posteriori probability, the conditional risk is defined as follows:

$$R(\hat{\mathbf{y}}|\mathbf{x}) = \sum_{\mathbf{y}\in\{0,1\}^N} L_{DAGH}(\mathbf{y},\hat{\mathbf{y}})P(\mathbf{y}|\mathbf{x}). \tag{4}$$

Let Ψ be a set of label vectors, which meet the requirements of the hierarchy constraint. According to the Bayesian decision theory based on the minimum risk principle, in the hierarchical multi-label classification problem above, the optimal prediction label vector $\hat{\mathbf{y}^*}$ can be expressed as follows:

$$\begin{aligned}\hat{\mathbf{y}^*} &= \arg\min_{\hat{\mathbf{y}}\in\Psi} R(\hat{\mathbf{y}}|\mathbf{x})\\ &= \arg\min_{\hat{\mathbf{y}}\in\Psi} \sum_{\mathbf{y}\in\{0,1\}^N} L_{DAGH}(\mathbf{y},\hat{\mathbf{y}})P(\mathbf{y}|\mathbf{x}).\end{aligned} \tag{5}$$

For an instance $\mathbf{x}$, the probability of i at the node $P(y_i = 1|\mathbf{x})$ is abbreviated as p_i. Through a detailed mathematical derivation and arrangement, the minimization problem above in formula (5) is equivalent to the maximization problem below,

$$\hat{\mathbf{y}}^* = \arg\max_{\hat{\mathbf{y}}\in\Psi} \sum_i \hat{y}_i\sigma(i), \tag{6}$$

where for a node i, $\sigma(i)$ is specifically defined as

$$\sigma(i) = \begin{cases}\sigma_1(i), & i = 0,\\ \sigma_1(i) + \sigma_2(i), & i > 0,\end{cases} \tag{7}$$

in which $\sigma_1(i)$ and $\sigma_2(i)$ are defined as

$$\sigma_1(i) = \sum_{j\in child(i)} w_2 C_j p_j - \prod_{j\in child(i)} w_1 C_j p_j, \tag{8}$$

$$\sigma_2(i) = w_1 C_i p_i - w_3 C_i\Big(\prod_{j\in par(i)} p_j - p_i\Big) - w_4 C_i \sum_{j\in par(i)} (1 - p_j). \tag{9}$$

In summary, the DAG hierarchical multi-label classification problem is converted to the optimization problem described in formula (6). To solve the optimization problem, only the posterior probability of an instance at each node needs to be obtained. In this paper, a classifier is designed based on the C2AE model to get the probability that an instance belongs to a certain node.

B. C2AE model

To solve the optimization problem in formula (6), we should first obtain the probability p_i for each node i by training binary classifiers on the training data and, then, use p_i to compute $\sigma(i)$ according to (7)–(9). We utilize a deep neural network (DNN) based model, Canonical Correlated AutoEncoder (C2AE),[11] for obtaining the probability p_i. C2AE is achieved by integrating the DNN architectures of canonical correlation analysis and autoencoder, which learns a deep latent space from both feature and label domains for multi-label classification. C2AE has the ability to exploit label dependency during prediction. As illustrated in Fig. 1, C2AE integrates two DNN models, deep canonical correlation analysis (DCCA) and autoencoder, with three mapping functions to be determined: feature mapping F_x, encoding function F_e, and decoding function F_d.

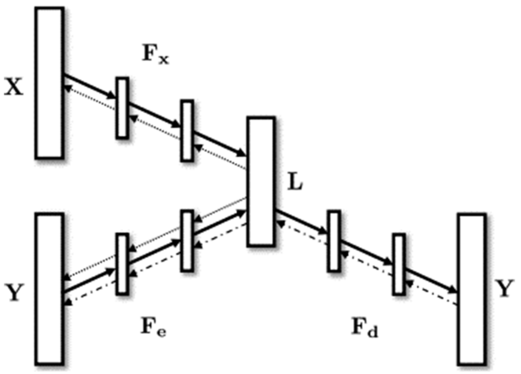


FIG. 1. The architecture of the C2AE model.

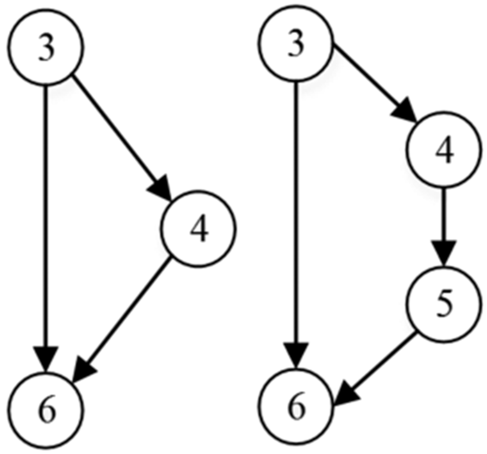


FIG. 2. Special cases in the DAGLabel algorithm.

In the training phase, the input of C2AE is the feature vectors X of the training instances and their label vectors Y, while the output is the label vectors Y. The DCCA module associates X and Y, and the autoencoder module enforcing the output is recovered as Y, which is the same as the input, so that the C2AE can determine the deep latent space. In the prediction phase, the feature mapping function

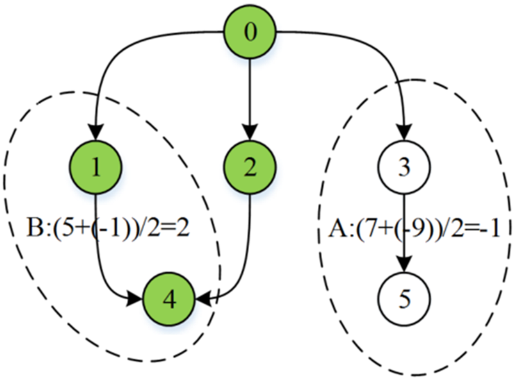


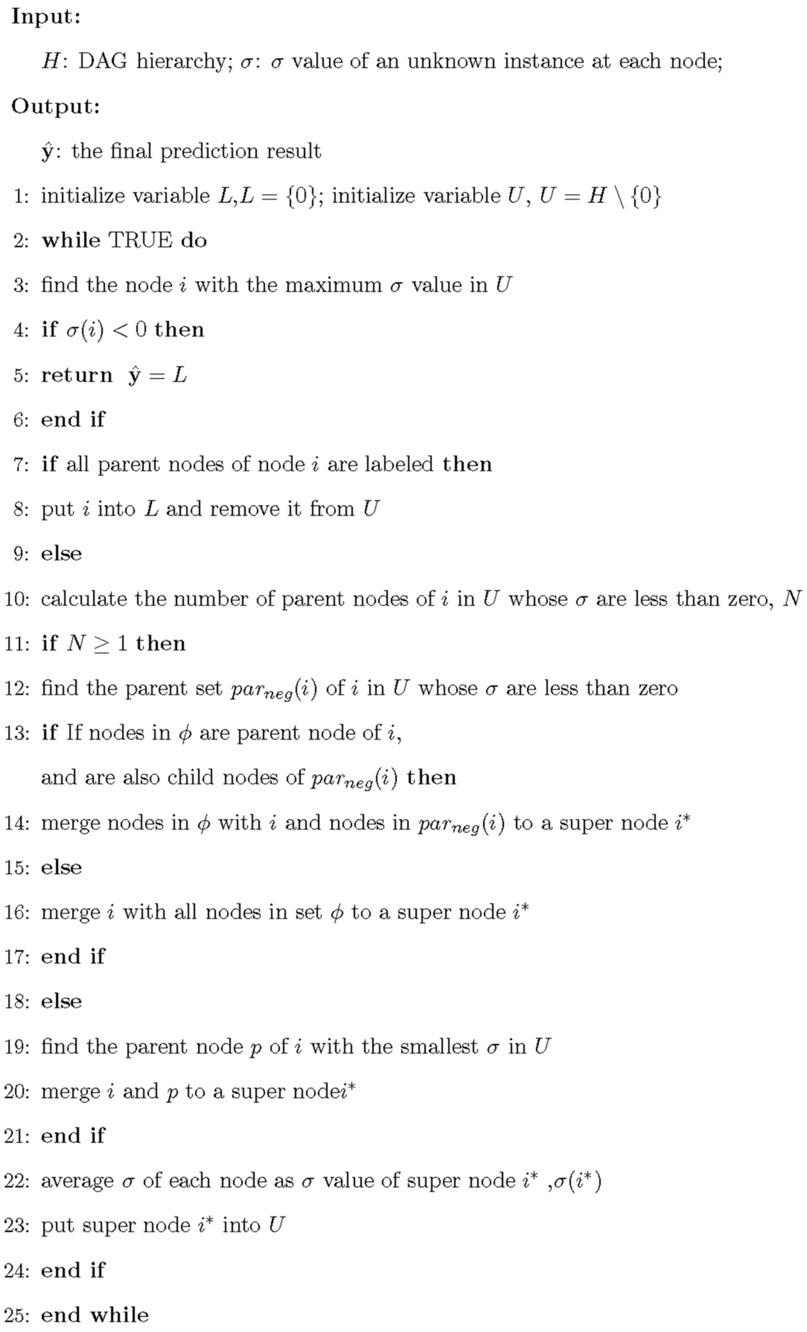
FIG. 3. DAGLabelling steps.

TABLE I. σ value and the order for each node.

Node	0	5	4	2	1	3
σ value	∞	7	5	4	-1	-9
Order	1	2	3	4	5	6

F_x converts the input feature vector $\mathbf{x}$ of the testing instances into the deep latent space, and, then, the predicting labels $\hat{\mathbf{y}}$ are obtained as output by the decoding function F_d as follows:

$$\hat{\mathbf{y}} = F_d(F_x(\mathbf{x})). \tag{10}$$



```
Input:
    $H$: DAG hierarchy; $\sigma$: $\sigma$ value of an unknown instance at each node;
Output:
    $\hat{\mathbf{y}}$: the final prediction result
initialize variable $L$,$L = \{0\}$; initialize variable $U$, $U = H \setminus \{0\}$
while TRUE do
    find the node $i$ with the maximum $\sigma$ value in $U$
    if $\sigma(i) < 0$ then
        return  $\hat{\mathbf{y}} = L$
    end if
    if all parent nodes of node $i$ are labeled then
        put $i$ into $L$ and remove it from $U$
    else
        calculate the number of parent nodes of $i$ in $U$ whose $\sigma$ are less than zero, $N$
        if $N \geq 1$ then
            find the parent set $par_{neg}(i)$ of $i$ in $U$ whose $\sigma$ are less than zero
            if If nodes in $\phi$ are parent node of $i$,
            and are also child nodes of $par_{neg}(i)$ then
                merge nodes in $\phi$ with $i$ and nodes in $par_{neg}(i)$ to a super node $i^*$
            else
                merge $i$ with all nodes in set $\phi$ to a super node $i^*$
            end if
        else
            find the parent node $p$ of $i$ with the smallest $\sigma$ in $U$
            merge $i$ and $p$ to a super node$i^*$
        end if
        average $\sigma$ of each node as $\sigma$ value of super node $i^*$ ,$\sigma(i^*)$
        put super node $i^*$ into $U$
    end if
end while
```

FIG. 4. The framework of the DAGLabel algorithm.

At the same time, as we get the output prediction labels, we can also get the probability that $\hat{\mathbf{y}}$ holds given $\mathbf{x}$ (i.e., p_i).

The datasets used to predict gene function are usually high-dimensional feature vectors. The C2AE model can better process high-dimensional data with the ability to extract features more efficiently. Furthermore, the canonical correlation analysis module in the C2AE model is able to exploit the label dependency. In addition, the C2AE model has strong robustness and high efficiency.

C. DAGLabel algorithm

With p_i available, σ can be computed according to formula (7). By solving the maximization problem (6), we can obtain the optimal prediction label vector $\hat{\mathbf{y}}^*$. Due to the DAG hierarchical constraint, the optimal prediction results must be consistent with the DAG hierarchy.

We propose a greedy algorithm, DAGLabel, to solve the optimization problem above. By merging nodes into a supernode, the DAGLabel algorithm can gradually simplify the hierarchical structure of the directed acyclic graph to solve the maximization problem (6), while ensuring that the classification results meet the requirements of the hierarchy constraint. For an instance, the specific procedure of the DAGLabel algorithm is as follows:

Find the node with the maximum σ value, through traversing each node of the hierarchy. If all of its parent nodes are labeled positive, then label it positive. Otherwise let the number of its parent nodes with the negative σ value as k.

If k is not more than 1, merge the node and its parent node with the minimum σ value as a supernode. Otherwise, merge the node and all of its parent nodes with negative σ value as a supernode. The σ value of the supernode is the average σ value of all the nodes contained in it. In addition, the supernode preserves the interrelationships of all the nodes contained in it in the hierarchy. The procedure of the DAGLabel algorithm stops when the maximum σ value is negative or all nodes in the hierarchy are labeled positive.

Since the algorithm procedure may involve the merging procedure for each node in the hierarchy, if there are N nodes in a DAG, the time complexity is $O(N\log(N))$.

There is a special case to be considered. As illustrated in Fig. 2, the cyclic graph structure will be generated when node 3 and node 6 are merged as a supernode. Due to the complexity of the DAG structure, it could get more complicated. However, the commonality of such problems is that a cyclic graph structure is generated. Therefore, to solve the problem, all the nodes should be merged together as a supernode in this case.

Figure 3 shows a simple example to illustrate the procedure of the DAGLabel algorithm. The σ value of an instance in this example for each node is shown in Table I. It can be seen that nodes 2 and 4 are merged as a supernode, and nodes 3 and 5 are merged as a supernode. The final prediction results are that this instance has labels 0, 1, 2, and 4. The pseudocode of the DAGLabel algorithm is described in Algorithm in Fig. 4.

IV. EXPERIMENT

In this section, the datasets and the experimental settings are introduced first. To evaluate the performance of different algorithms, the evaluation criteria are given in the second part. Finally, the results of the proposed algorithm are analyzed and discussed.

A. Datasets and settings

The purpose of gene function annotation is to annotate the gene function of an organism through extensive biological experiments. Aiming at establishing a term standard for defining and describing the functions of genes and proteins in a variety of species, which can be updated continuously as the research goes on, Gene Ontology (GO) is established by the Gene Ontology Consortium. Different GO terms are associated in the form of DAG. GO provides a systematic definition of the three-tier structure to describe the function of gene products, including molecular function (MF), biological process (BP), and Cellular Component (CC). In this study, we select GO terms in the biological process (BP). Annotations for human proteins were obtained from the Gene Ontology Annotation (GOA) database released on 2018-12-18.

To verify the proposed algorithm, it is used to predict the function of human genes. Human protein datasets selected in the study are available from the paper.[12] The human protein datasets are divided into 7 depending cubes, denoted Ha to Hg. For that the samples with some GO terms are too few to complete the training stage, we abandon the GO terms with samples less than 100. The information of the datasets is shown in Table II.

B. Performance evaluation

In the hierarchical multi-label classification, the prediction is a label set so that the evaluation of the prediction is more complicated. Many evaluation methods have been put forward, but none is considered to be the best. Macro-averaging F_1-score and micro-averaging F_1-score are widely used in many studies, so we select these two evaluation standards in the experiment.[13]

To illustrate macro-averaging F_1-score and micro-averaging F_1-score, the definition of hierarchical precision (hP), hierarchical recall (hR), and hierarchical F_1-score (hF_1) is given first,

$$hPre_i = \frac{|\hat{P}_i \cap \hat{T}_i|}{|\hat{P}_i|}, \tag{11}$$

$$hRec_i = \frac{|\hat{P}_i \cap \hat{T}_i|}{|\hat{T}_i|}, \tag{12}$$

$$hF_{1,i} = \frac{2 \times hPre_i \times hRec_i}{hPre_i + hRec_i}, \tag{13}$$

where, for the ith sample in the sample set, T_i is its true label set and P_i is its prediction label set.

TABLE II. Information of the experimental datasets.

Dataset	Feature No.	Train data No.	Test data No.	GO terms
Ha	258	1000	500	124
Hb	258	1000	500	129
Hc	258	1000	500	123
Hd	258	1000	500	129
He	258	1000	500	128
Hf	258	1000	500	139
Hg	258	1000	500	131

Assuming that there are m instances in the dataset, the micro-averaging version of hierarchical precision, hierarchical recall, and hierarchical F_1-score is defined as follows:

$$hPre^{\mu} = \frac{\sum_{i=1}^{m} |\hat{P}_i \cap \hat{T}_i|}{\sum_{i=1}^{m} |\hat{P}_i|}, \tag{14}$$

$$hRec^{\mu} = \frac{\sum_{i=1}^{m} |\hat{P}_i \cap \hat{T}_i|}{\sum_{i=1}^{m} |\hat{T}_i|}, \tag{15}$$

$$hF_1^{\mu} = \frac{2 \times hPre^{\mu} \times hRec^{\mu}}{hPre^{\mu} + hRec^{\mu}}. \tag{16}$$

In addition, the macro-averaging version of the hierarchical precision, hierarchical recall, and hierarchical F_1-score is defined as follows:

$$hPre^{M} = \frac{\sum_{i=1}^{m} hPre_i}{m}, \tag{17}$$

$$hRec^{M} = \frac{\sum_{i=1}^{m} hRec_i}{m}, \tag{18}$$

$$hF_1^{M} = \frac{\sum_{i=1}^{m} hF_{1,i}}{m}. \tag{19}$$

There is a common characteristic for all metrics; the lager measure value, the better the classifier performance. Hierarchical F_1-score takes hierarchical precision and hierarchical recall into consideration comprehensively. Therefore, macro-averaging and micro-averaging hierarchical F_1-scores are mainly used in our study.

C. Results and analysis

In order to verify the performance of the C2AE-DAGLabel algorithm, through the analysis of related literature, TPR algorithm, HR-SVM algorithm, and CLUS algorithm are selected for comparison. These algorithms are classical and representative methods for DAG hierarchical multi-label classification and are selected from both local and global classification approaches. The main contents of these algorithms are summarized as follows.

TPR algorithm belongs to a local classifier approach.[14] When TPR works for classification, the classification results for each node are not only depending on the result of the base classifier, but also the impact of the bottom nodes and the top nodes is taken into consideration.

TABLE III. Micro-averaging hierarchical F_1-score performance comparison between C2AE-DAGLabel and other methods.

Dataset	C2AE-DAGLabel	TPR	HR-SVM	CLUS
Ha	0.428	0.404	0.433	0.423
Hb	0.442	0.405	0.440	0.431
Hc	0.444	0.375	0.445	0.409
Hd	0.430	0.394	0.430	0.413
He	0.438	0.424	0.431	0.418
Hf	0.427	0.385	0.419	0.412
Hg	0.443	0.432	0.421	0.417

TABLE IV. Macro-averaging hierarchical F_1-score performance comparison between C2AE-DAGLabel and other methods.

Dataset	C2AE-DAGLabel	TPR	HR-SVM	CLUS
Ha	0.396	0.390	0.378	0.394
Hb	0.413	0.385	0.386	0.407
Hc	0.414	0.371	0.390	0.386
Hd	0.396	0.370	0.373	0.380
He	0.404	0.396	0.381	0.396
Hf	0.396	0.376	0.369	0.398
Hg	0.410	0.409	0.371	0.383

The HR-SVM algorithm is also a local classifier approach for DAG hierarchical multi-label classification.[15] The algorithm selects R-SVM of the support vector machine whose threshold is adjustable as the basic classifier. R-SVM can deal with the imbalanced dataset problem. When the preliminary classification results of all nodes are obtained, the HR-SVM algorithm uses the top-down algorithm to ensure that the final classification results of samples meet the requirements of the hierarchy constraint.

The CLUS algorithm is a global classifier approach. It adopts the inductive decision tree method.[9] By assigning different weights to different nodes in the hierarchy, the decision tree is generated to classify all labels at one time.

The results under evaluation are as illustrated in Tables III and IV. To be more explicit, we draw the bar charts in Figs. 5 and 6, and we make the evaluation rank tables as Table V.

By analyzing the results, it can be seen that the C2AE-DAGLabel algorithm achieves the best classification effect in the above 7 datasets under two evaluation standards. The effectiveness of the proposed algorithm is mainly due to three aspects.

First, the mathematical model for solving the DAG hierarchical multi-label classification problem is realized by the C2AE-DAGLabel algorithm, which can theoretically guarantee that the algorithm has a good effectiveness.

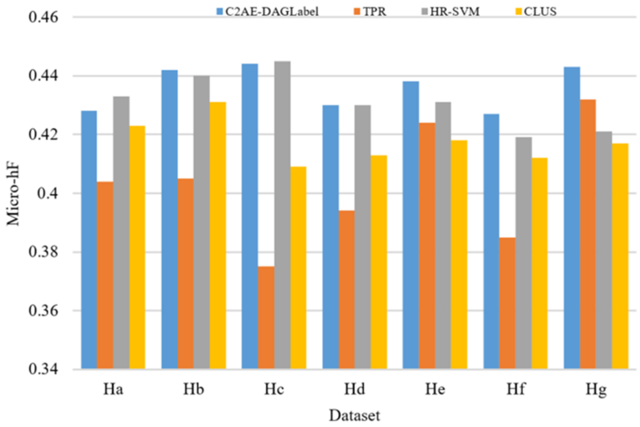


FIG. 5. The bar chart of micro-averaging hierarchical F_1-score performance comparison between C2AE-DAGLabel and other methods.

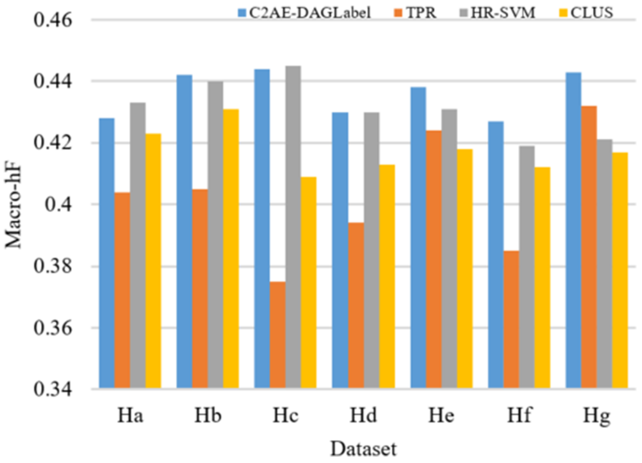


FIG. 6. The bar chart of the macro-averaging hierarchical F1-score performance comparison between C2AE-DAGLabel and other methods.

TABLE V. Micro-averaging and macro-averaging hierarchical F1-score performance rank between C2AE-DAGLabel and other methods.

Algorithm	Micro-averaging order	Macro-averaging order
C2AE-DAGLabel	1.29	1.14
TPR	3.43	3.14
HR-SVM	1.71	3.29
CLUS	3.43	2.14

Second, a C2AE model with good effectiveness and robustness is used as the classifier, which is based on a deep neural network and can mine higher-order relations among multiple labels. The C2AE model can improve the classification ability and ensure the classification accuracy of each node.

Finally, the DAGLabel algorithm can guarantee that the final classification results meet the requirements of hierarchical constraints and at the same time give better final prediction results.

V. DISCUSSION AND CONCLUSION

Gene function prediction based on Gene Ontology can be treated as DAG hierarchical multi-label classification. In DAG hierarchical multi-label classification, there are complex hierarchical constraints among labels. Currently, there are fewer algorithms for DAG hierarchical multi-label classification, with poor accuracy and unsatisfactory performance.

In this paper, a C2AE-DAGLabel algorithm is proposed for DAG hierarchical classification by the idea of problem transformation. The C2AE-DAGLabel algorithm realizes the mathematical model, which changes the DAG hierarchical multi-label classification problem into a set of binary classification problems. In the C2AE-DAGLabel algorithm, the C2AE model based on the deep neural network is selected as the classifier, and a hierarchical constraint algorithm for the DAG hierarchical structure called the DAGLabel algorithm is designed. The DAGLabel algorithm can complete the task of hierarchical multi-label classification, while ensuring that the final classification results meet the requirements of hierarchical constraints.

The C2AE-DAGLabel algorithm is used to predict functions of human genes, and its performance is compared with several other commonly used TPR, HR-SVM, and CLUS algorithms. Experimental results show that the C2AE-DAGLabel algorithm has some advantages for solving DAG hierarchical multi-label classification problem.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61971153).

REFERENCES

[1] J. Meng, J. S. Wekesa, G. L. Shi, and Y. S. Luan, "Protein function prediction based on data fusion and functional interrelationship," Math. Biosci. **274**, 25–32 (2016).

[2] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: Tool for the unification of biology. The gene ontology consortium," Nat. Genet. **25**, 25–29 (2015).

[3] L. Zhang, S. K. Shah, and I. A. Kakadiaris, "Hierarchical multi-label classification using fully associative ensemble learning," Pattern Recognit. **70**, 89–103 (2017).

[4] J. Levatić, D. Kocev, and S. Džeroski, "The importance of the label hierarchy in hierarchical multi-label classification," J. Intell. Inf. Syst. **45**, 247–271 (2015).

[5] L. Cheng, H. Lin, Y. Hu, J. Wang, and Z. Yang, "Gene function prediction based on the gene ontology hierarchical structure," PLoS One **9**, e107187 (2013).

[6] M. Ramírez-Corona, L. E. Sucar, and E. F. Morales, "Hierarchical multilabel classification based on path evaluation," Int. J. Approximate Reasoning **68**, 179 (2016).

[7] R. Cerri, R. C. Barros, and A. C. de Carvalho, "Hierarchical multi-label classification using local neural networks," J. Comput. Syst. Sci. **80**, 39–56 (2014).

[8] C. Zeng, W. Zhou, L. Tao, L. Shwartz, and G. Y. Grabarnik, "Knowledge guided hierarchical multi-label classification over ticket data," IEEE Trans. Network Serv. Manage. **14**, 246–260 (2017).

[9] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," Mach. Learn. **73**, 185–214 (2008).

[10] R. T. Alves, M. R. Delgado, and A. A. Freitas, "Multi-label hierarchical classification of protein functions with artificial immune systems," in *Advances in Bioinformatics and Computational Biology*, edited by A. L. C. Bazzan, M. Craven, and N. F. Martins (Springer Berlin Heidelberg, Berlin, Heidelberg, 2008), pp. 1–12.

[11] C. Yeh, W. Wu, W. Ko, and Y. F. Wang, "Learning deep latent spaces for multi-label classification," Mach. Learn. (published online 2017), arXiv:1707.00418.

[12] R. Fa, D. Cozzetto, C. Wan, and D. T. Jones, "Predicting human protein function with multi-task deep neural networks," PLoS One **13**, e0198216 (2018).

[13] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," Data Min. Knowl. Discovery **22**, 31–72 (2011).

[14] B. Chen and J. Hu, "Hierarchical multi-label classification based on over-sampling and hierarchy constraint for gene function prediction," IEEJ Trans. Electr. Electron. **7**, 183–189 (2012).

[15] P. Vateekul, M. Kubat, and K. Sarinnapakorn, "Hierarchical multi-label classification with SVMs: A case study in gene function prediction," Intell. Data. Anal. **18**, 717–738 (2014).