MATH5472 CASI Report: Hierarchical Multi-Label Classification Networks

StuID:21003122 LI,Meng

December 15, 2024

1 Introduction

One of the most challenging problems in machine learning is data classification. In this challenging scenario, there exists a complicated problem called hierarchical multi-labels classification(HMC). Unlike multi-label classification where an object is associated with particular classes from a set of disjoint classes, classes in the HMC task are organized in a hierarchical structure. Specifically, an object is assigned to many labels, and the hierarchical structure of these labels are formed in a tree for a Directed Acyclic Graph(DAG).

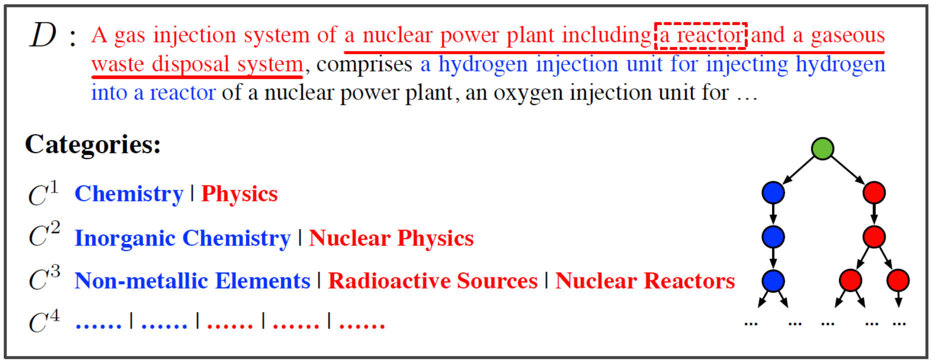
This problem arises in many fields, such as text classification, image annotation and bioinformatics tasks. For instance, in Figure **??**, the *red* words are related to *Physics* and the *blue* ones are related to *Chemistry* of C^1 (level-1 category). Additionally, the *red underlines* are related to *Nuclear Physics*, and *blue underlines* related to *Inorganic Chemistry* where these two labels are in C^2 (level-2 category). The tree diagram on the right-hand side illustrates the hierarchical structure of the multi-labels.

2 Recent Methods

Algorithms that perform HMC must be able to not only assign correct labels to objects, but also follow the given hierarchical structure. There are two paradigms to perform HMC.

Local approaches attempt to find local information in each class hierarchy. So these methods need classifiers for particular nodes or particular hierarchy[2] to discover the class hierarchical relationship and gain local prediction. And then algorithm combine all local prediction to generate the final classification. This kind of approaches,namely top-down paradigm, is much suitable for gaining local hierarchical relationship. However, these methods are computationally expensive since they rely on a cascade of classifiers. Additionally, these methods neglect the hierarchical structure, which may lead to error propagation and membership inconsistency[3].

The global approaches are much cheaper than the local ones. And also they do not suffer from error propagation and menbership inconsistency. For example, Clus-HMC[4] is a global approach that



D : A gas injection system of a nuclear power plant including a reactor and a gaseous waste disposal system, comprises a hydrogen injection unit for injecting hydrogen into a reactor of a nuclear power plant, an oxygen injection unit for …

Categories:

C^1 **Chemistry** | **Physics**

C^2 **Inorganic Chemistry** | **Nuclear Physics**

C^3 **Non-metallic Elements** | **Radioactive Sources** | **Nuclear Reactors**

C^4 …… | …… | …… | …… | ……

Figure 1: A toy example of a patent document in HTMC problem[1]

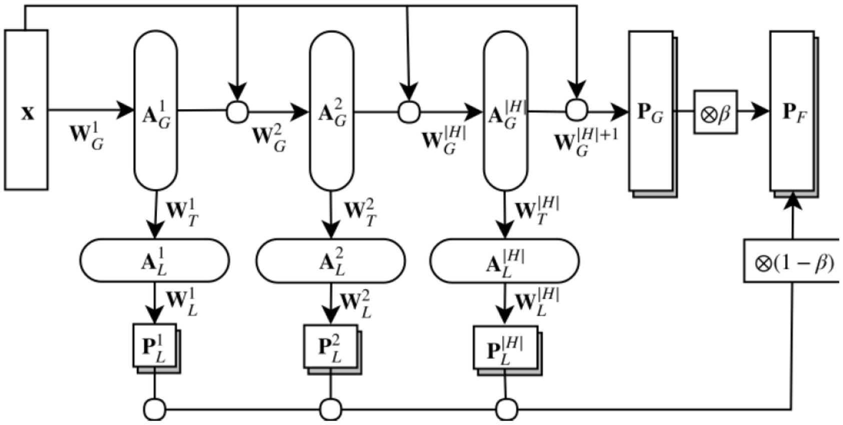


Figure 2: HMCN-F architecture

builds a single decision tree to classify all categories simultaneously. But they are not able to capture the local relationship.

So in this article[5], authors propose a new method called HMCN(Hierarchical Multi-label Classification Networks), which combines the advantage of local and global approaches. This is a hybrid method which is capable of simultaneously optimizing both local and global loss functions.

Actually authors implement two neural networks. One is based on fully connected neural network architecture(HMCN-F), the other is based on recurrent neural network(HMCN-R).

In both versions, the architecture of networks is related to the classes hierarchy. And the network have multiple outputs which are responsible for tracking local and global information of hierarchical structure of classes. There are two information flows in the model.

- In the main flow, the information of features starts at the input layer and traverses all fully connected(FC) layers until it reaches the global output.
- In the local flow, similar to the main flow, the information starts at the input layer, then passes through its own local fully connected (FC) layer sequentially, and finally ends as the local logits (scores) for predicting the local hierarchical class.
- Eventually, the two flow interacts at the last FC layer to generate final global prediction.

At each hierarchical level, the layers receive both the original features and the outputs from the previous layer, producing multiple outputs. One output, A_G^h, carries the global information from the $(h)^{th}$ level to the$(h+1)^{th}$ level, and it also influences the output P_L^h in the local flow. The remaining outputs contain related local information.

3 HMCN-F

We denote $x \in R^{|D|}$ as input features and $|D|$ as the number of features. $|H|$ the total number of hierarchical levels, and $|C|$ the total number of classes.

The computation of main flow is:

$$\begin{aligned}
A_G^1 &= \phi(W_G^1 x + b_G^1) \\
A_G^h &= \phi(W_G^h (A_G^{h-1} \odot x) + b_G^h) \\
&\vdots \\
A_G^{|H|} &= \phi(W_G^{|H|-1} (A_G^{|H|-1} \odot x) + b_G^{|H|}) \\
P_G &= \sigma(W_G^{|H|+1} (A_G^{|H|} \odot x) + b_G^{|H|+1})
\end{aligned}$$

And for local flow, the authors design a transition layer A_L^h to transit information into right shape. And the local information is learned by P_L^h. The computation of local flow is:

$$A_L^h = \phi(W_T^h A_G^h + B_T^h)$$
$$P_L^h = \sigma(W_L^h A_L^h + B_L^h)$$

Then the final prediction is

$$P_F = \beta(P_L^1 \odot P_L^2 \odot \dots P_L^{|H|}) + (1-\beta)P_G,$$

where σ is necessarily a sigmoidal function. And β is 0.5 by default since it gives us the equal importance of local and global information extracting from the HMCN.

3.1 Loss Function

HMCN minimizes the sum of local($\mathcal{L}_L$), global($\mathcal{L}_G$) loss function and hierarchical violation penalty ($\mathcal{L}_H$):

$$\mathcal{L} = \mathcal{L}_L + \mathcal{L}_G + \lambda\mathcal{L}_H$$
$$\mathcal{L}_L = \sum_{h=1}^{|H|}[\varepsilon(P_L^h, Y_L^h)]$$
$$\mathcal{L}_G = \varepsilon(P_G, Y_G)$$
$$\mathcal{L}_{H_i} = \max\{0, Y_{in} - Y_{ip}\}^2$$

where $\varepsilon(P, Y)$ is the binary cross-entropy.

Also, in order to keep hierarchical consistency, the authors introduce hierarchical violation penalty to constrain the network. However, this penalty does not improve the performance of HMCN as my opinion. So when I reproduce the results of this article, I ignore this penalty $\mathcal{L}_H$. Instead, I only consider local and global loss.

4 Reproduce of HMCN-F

To make things more simple, I only do experiments on *CELLCYCLE* [1] since it is not difficult to apply the code on different datasets.

1. load the dataset [2], and organize the hierarchical labels Y_i of each object in a tree structure.
2. implement HMCN-F with pytorch
3. train model
4. evaluate the model with $AU(\overline{PRC})$ on test set

Table 1: Number of features (D),number of classes (n), and number of datapoints

TAXONOMY	DATASET	D	n	TRAINING	TEST
FUNCAT(FUN)	CELLCYCLE	77	499	1625	848
FUNCAT(FUN)	DERISI	63	499	1605	842
FUNCAT(FUN)	EXPR	551	499	1636	849
FUNCAT(FUN)	GASCH1	173	499	1631	846
FUNCAT(FUN)	GASCH2	52	499	1636	849
FUNCAT(FUN)	SEQ	478	499	1692	876
FUNCAT(FUN)	SPO	80	499	1597	837

[1] The weblink of dataset is from https://dtai.cs.kuleuven.be/software/clus/hmcdatasets/

[2] I use this github repository https://github.com/EGiunchiglia/C-HMCNN to parse data in Weka's arff format

4.1 Experimental Results

I think I make some mistake in implementation of HMCN-F, and that's why it is so inconsistent with the authors' results.

Table 2: Comparison of my implement and the author's with $\lambda = 0$. The performance is measured as the $AU(\overline{PRC})_{micro}$ on test set

DATASET	HMCN-F(author)	HMCN-F(my Implement)
CELLCYCLE	0.250	0.139
DERISI	0.191	0.125
EXPR	0.296	0.157
GASCH1	0.281	0.150
GASCH2	0.252	0.138
SEQ	0.283	0.171
SPO	0.210	0.126

4.2 Github Link

https://github.com/caizihuahua/myHMCN

5 HMCN-R

Since the reproduction of the code HMCN-F does not perform well, I have not completed this part yet. I am still working on debugging HMCN-F implemented by myself.

References

[1] W. Huang, E. Chen, Q. Liu, Y. Chen, Z. Huang, Y. Liu, Z. Zhao, D. Zhang, S. Wang, Hierarchical multi-label text classification: An attention-based recurrent network approach, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 1051–1060. doi:10.1145/3357384.3357885.
URL https://doi.org/10.1145/3357384.3357885

[2] M. Ruiz, P. Srinivasan, Hierarchical text categorization using neural networks, Information Retrieval 5 (2002) 87–118. doi:10.1023/A:1012782908347.

[3] C. Silla, A. Freitas, A survey of hierarchical classification across different application domains, Data Mining and Knowledge Discovery 22 (2011) 31–72. doi:10.1007/s10618-010-0175-9.

[4] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, H. Blockeel, Decision trees for hierarchical multi-label classification, Machine Learning 73 (2008) 185–214. doi:10.1007/s10994-008-5077-3.

[5] J. Wehrmann, R. Cerri, R. Barros, Hierarchical multi-label classification networks, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 5075–5084.
URL https://proceedings.mlr.press/v80/wehrmann18a.html