University of Nottingham
UK | CHINA | MALAYSIA

Machine Learning COMP3009 Coursework 2 – Decision Tree

Runzhuo Li, Xixuan Wang, Zike Li, Yu Zhang, Pinyuan Feng
Group YOLO, School of Computer Science, University of Nottingham
{psyrl6, scyxw3, scyzl2, scyyz4, scypf1}@nottingham.ac.uk

Abstract

This is the second coursework of Machine Learning COMP3009 on the topic of Decision Tree (DT). We applied the standard ID3 algorithm on Iris dataset[1] and Beijing PM2.5 dataset[2]to solve classification and regression problems respectively. For the evaluation part, we utilised 10-fold cross validation to evaluate our tree model to calculate average Root Mean Square Error (RMSE) for the regression problem and F1-Score for the classification problem. In this report, the original dataset information is briefly described in the first section, followed by an illustration about data pre-processing. Besides, the details of our experiments, including experiment configurations and evaluation, are demonstrated. Lastly, answers to the questions on the coursework sheet are presented.

Dataset Description

Classification

Iris dataset contains 3 different types of Iris flowers (Iris Setosa, Iris Versicolour, Iris Virgnica) with 50 plants for each. And 4 attributes are collected to predict the iris category.

Attribute	Explanation
Class	Class of flower (Setosa, Versicolour, Virginica)
sepallength.	sepal length in cm
sepalwidth.	sepal width in cm
petallength.	petal length in cm
petalwidth.	petal width in cm

Figure 1

Regression

The Beijing PM2.5 dataset contains the PM2.5 data of the US Embassy in Beijing, as well as meteorological data from Beijing Capital International Airport. There are 43824 instances with 13 attributes in the raw dataset.

Attribute	Explanation	Attribute	Explanation
No.	row number	DEWP.	Dew Point (â„ƒ)
year.	year of data in this row	TEMP	Temperature (â„ƒ)
month.	month of data in this row	PRES.	Pressure (hPa)
day.	day of data in this row	cbwd.	Combined wind direction
hour.	hour of data in this row	Iws.	Cumulated wind speed (m/s)
pm2.5.	PM2.5 concentration (ug/m^3)	Is.	Cumulated hours of snow
.		Ir.	Cumulated hours of rain

Figure 2

Data Pre-processing

This section shows how we pre-processed the data to prepare for the regression and classification implementation. Since the performance of tree is less sensitive to the data scale, we did not apply data normalization to the datasets.

Classification

Following the requirements of coursework description, 3 types of iris flowers should be divided into positive and negative class. Therefore, "Iris setos" and "iris virgnica" are set to be the positive class, and "iris versicolour" belongs to the negative class. After data pre-processing, we obtain a data matrix of 150 * 3, including the class label.

Regression

For Beijing PM2.5 dataset, we carefully selected the useful attribute based on the result of previous coursework. Since time-based attribute should be considered, we converted the time-based attributes {"day", "month", "year", "hour"} into "season" and "morning/afternoon" categories, with {1, 2, 3, 4} indicating the season and {1, 2} representing the morning and afternoon, respectively. Additionally, we converted cumulated attribute (e.g. "lr" represents accumulated hourly precipitation) into discrete values (e.g. "0" represents no precipitation, "1" otherwise), because the original data is dependent on time-based attributes. Finally, we obtain a data matrix of 43800 * 10, including the PM 2.5 attribute.

Besides, we found that PM2.5 values in the original dataset has a high variability, so we adjusted the values based on $log_2(y+1)$, where y represents a vector of PM 2.5 values, and we added one to avoid negative outcomes.

Experiment

In the course of implementation, we referred to the decision tree pseudocode provided in the coursework sheet and did some modifications particularly on the CHOOSE-ATTRIBUTE function. Then, we implemented the ID3 algorithm for the attribute selection by iterating all attributes.

The information gain for each attribute is calculated as follows: Entropy of current dataset minus the remainder entropy of that attribute after the best threshold is selected. We chose the attribute with the greatest information gain and recorded the attribute as well as the threshold.

Suppose the training set contains p positive and n negative examples, entropy is calculated based on $equation\ (1)$。

$$I\left(\frac{p}{p+n},\frac{n}{p+n}\right) = -\frac{p}{p+n}log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}log_2\left(\frac{n}{p+n}\right) \quad (1)$$

Testing of any attribute A will split the training set E into v subsets, where v is dependent on values of attribute A. Since some attributes has a continuous property, we dealt with the continuous values according to the $equation\ (2)$. Suppose the attribute A has $\{a_1, a_2, \dots, a_n\}$, then v equals $n-1$. The values are sorted in an ascending order to calculate each threshold.

$$Threshold_k = \frac{a_k + a_{k+1}}{2}, a_k < a_{k+1}\ and\ 1 \le k \le n-1 \quad (2)$$

$$Remainder(A) = \sum_{i=1}^{v} \frac{p_i}{p_i+n_i}\ I\left(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i}\right) \quad (3)$$

Based on the $equation\ (1)(2)(3)$, the information gain is calculated as $equation\ (4)$.

$$Gain(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - Remainder(A) \tag{4}$$

Classification

The classification tree is used to identify the "class" within which a target variable would most likely fall. It is to attain a homogeneous set of labels in each partition. Starting from the root node, the information gain values are caculated for all features. Then, we select the feature with the largest information gain as the node feature and establish sub-nodes according to the feature's different values. The process continues until no more useful splits can be found or reach the preset termination condition. If the label in the final leaf node is not unique, the label of most data is taken as the value of the leaf node.

In the first place, “iris setosa” was chosen to be the negative class, while iris versicolour and iris virgnica were chosen to be the positive class. After running the program, the tree always ended with one root node with a leaf on each side (*Figure* 3). This is caused by the fact that the values of “petalwidth” attribute of the “iris versicolour” are extremely smaller than that of others. The simple result is hard for us to see the tree splitting process as well as evaluate our code. To avoid similar situations, we chose “iris versicolour” to be the negative class instead.

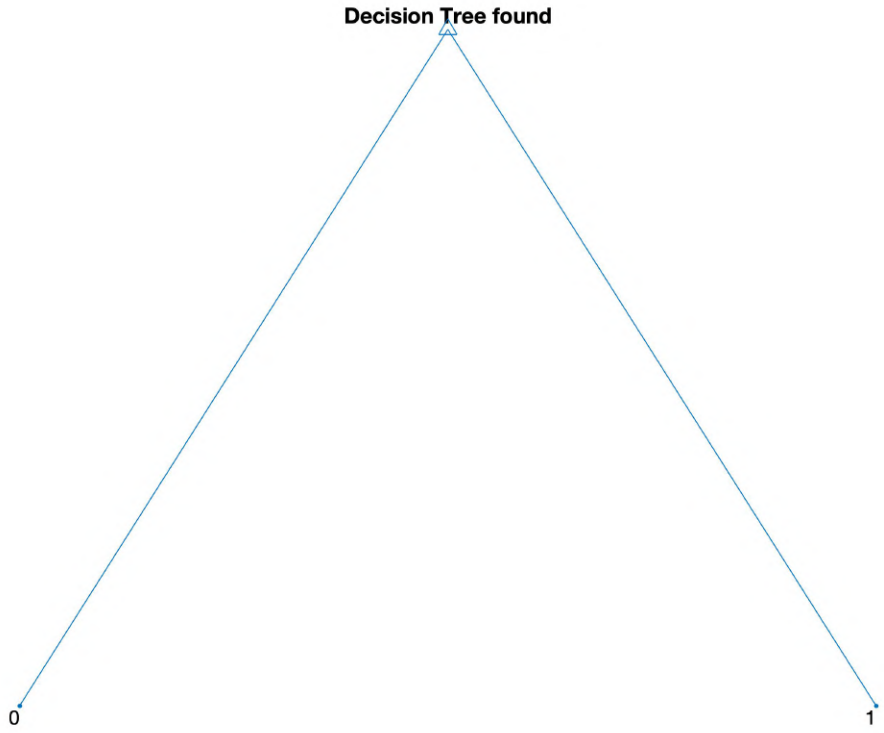

Figure 3

Regression

In this course work, the regression tree is also applied entropy to measure how "good" each attribute is in the set. Unlike the classification tree, the regression tree returns the mean value of all sub dataset variables as the value of the leaf node if it meets the stop condition. As for the stopping criterion, it stops growing the tree when they would result in nodes containing less than the pre-defined number of data. Other parts are the same as the classification tree.

Evaluation (10-fold cross validation)

The entire data set was divided into 10 folds to implement the 10-fold cross validation. The 10-fold cross-validation results for regression and classification problem are presented below as well as the tree graphs. Here, the resulting decision trees are from a MATLAB structure with fields that include *tree.op*, *tree.kids*, *tree.class*, *tree.attribute* and *tree.threshold*.

Regression			***Classification***			
Fold	***RMSE***		***Fold***	***Precision***	***Recall***	***F1-Score***
1	0.282082		*1*	1.000000	1.000000	1.000000
2	0.287488		*2*	1.000000	1.000000	1.000000
3	0.273737		*3*	0.933333	1.000000	0.965517
4	0.276298		*4*	0.866667	0.777778	0.819820
5	0.287960		*5*	0.933333	0.800000	0.861538
6	0.282630		*6*	0.866667	0.875000	0.870813
7	0.278890		*7*	1.000000	1.000000	1.000000
8	0.280928		*8*	0.933333	0.833333	0.880503
9	0.283258		*9*	0.933333	1.000000	0.965517
10	0.281110		*10*	1.000000	1.000000	1.000000
Average	0.2814381		***Average***	0.9466666	0.9286111	0.9363708

Table 1

The performance of classification tree and the regression tree is evaluated by 10-fold cross-validation. The table shows results of different folds based on corresponding evaluation metrics.

For classification problem, the precision values float between 86.7% and 100%, the recall values range from 77.8% to 100% and the results of F1-Score measure are in the interval between 86.1% and 100%. For regression problem, we obtain low RMSE results with the average of 28.14%. In general, the results indicate that our trees achieve a relatively good performance.

Classification

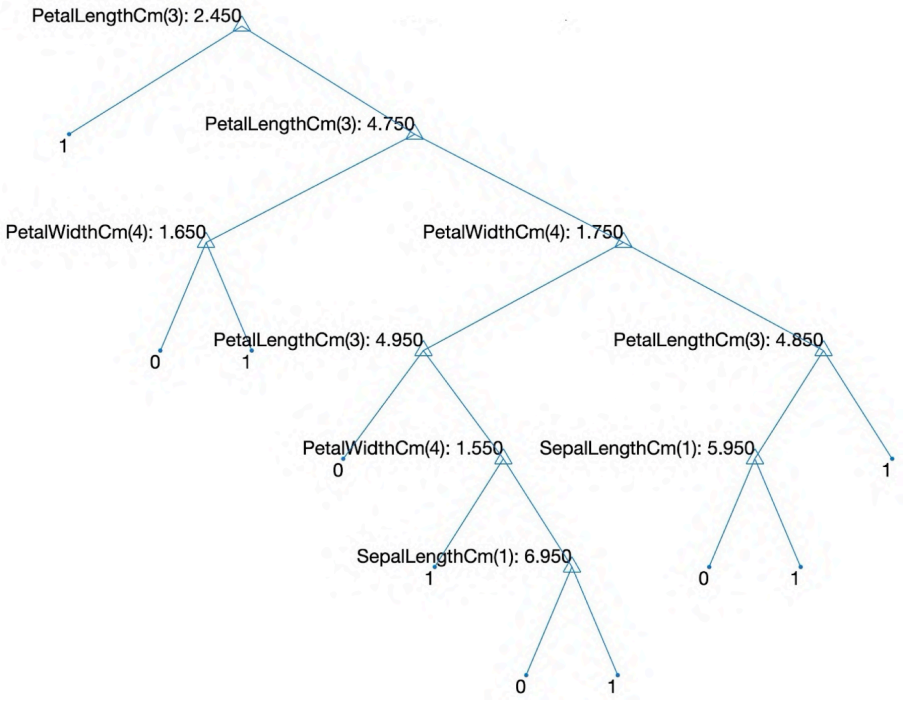

Figure 4

Figure 4 shows an example of the decision tree for iris classification. This tree, with depth of 6, owns 19 nodes in total, including 1 root node, 8 internal nodes and 10 leaf nodes. Its precision is 1, recall is 1 and f1 measure is 1.

Regression

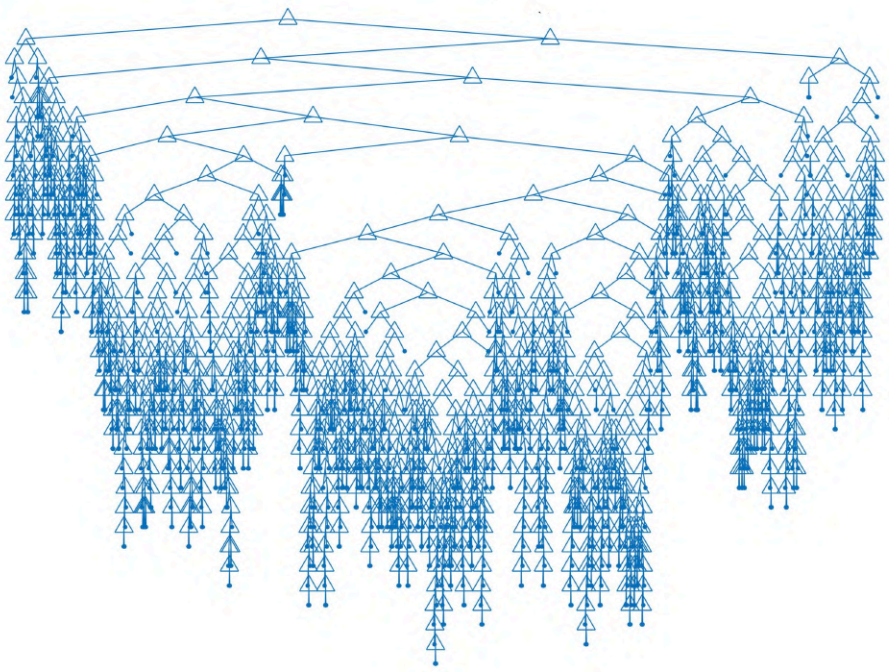

Figure 5

Figure 5 shows an example of the decision tree for regression. This tree, with depth of 42, owns 2419 nodes in total, including 1 root node, 1208 internal nodes and 1210 leaf nodes. Its RMSE is 0.278890.

University of Nottingham
UK | CHINA | MALAYSIA

Answers to Questions on the coursework sheet

Pruning

Pruning is a useful approach that shapes a decision tree. This approach can be further categorized into pre-pruning and post-pruning. Pre-pruning requires a stopping criterion when choosing the best split, while post-pruning simplifies the tree until reaching a leave node. And the leaves will be tested with data from the test set. Pruning helps to prevent overfitting and improve generalization by reducing the complexity of a decision tree. The pruning strategy has a huge impact on the decision tree, and the correct pruning strategy is the core of the optimized decision tree algorithm.

The method of pruning the node is to compare the accuracy when the leaf node is merged into the parent node from children nodes. If the accuracy improves, then the original nodes are replaced by the current single node. Otherwise, the tree will roll back to the last edition and merge into different nodes again until there are no more nodes merged. When the pruning process is accomplished, the current tree is simpler from the structure, the depth, nodes and branches. In addition, the accuracy and the generalization are also improved.

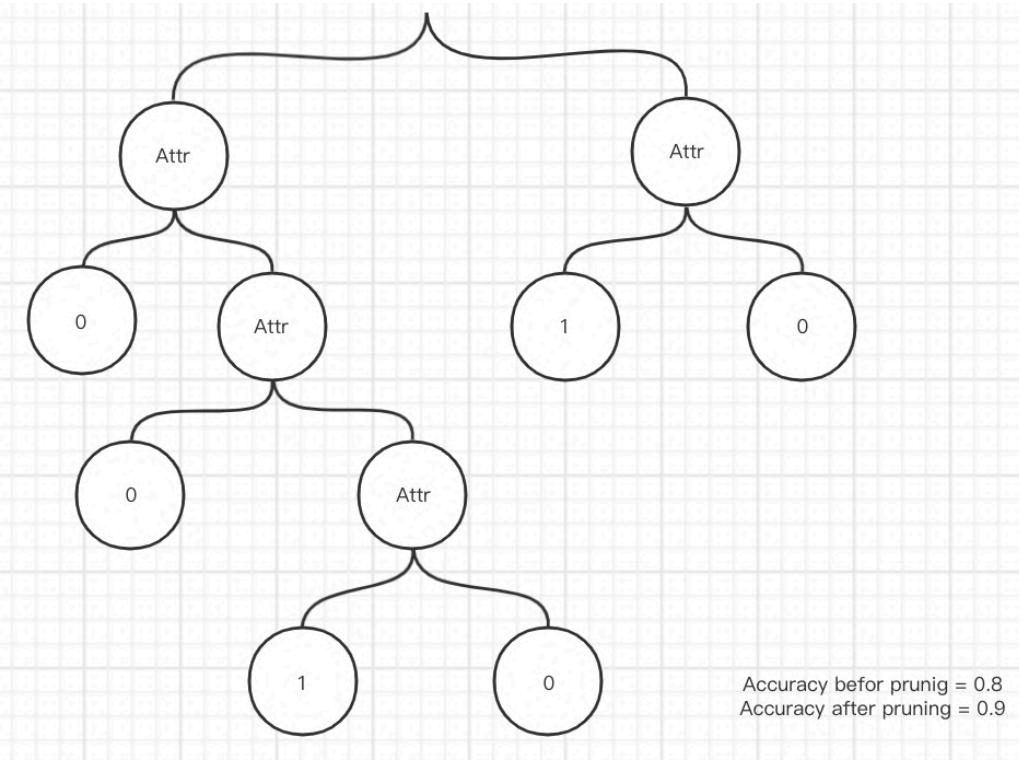

Figure 6 Original Tree

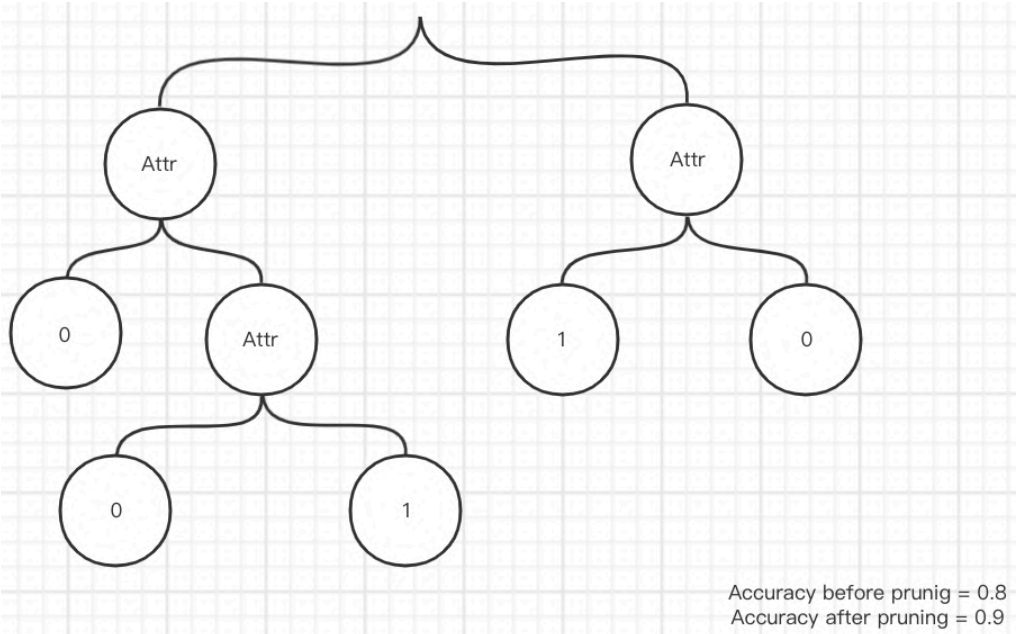

Figure 7 Pruned Tree with higher performance

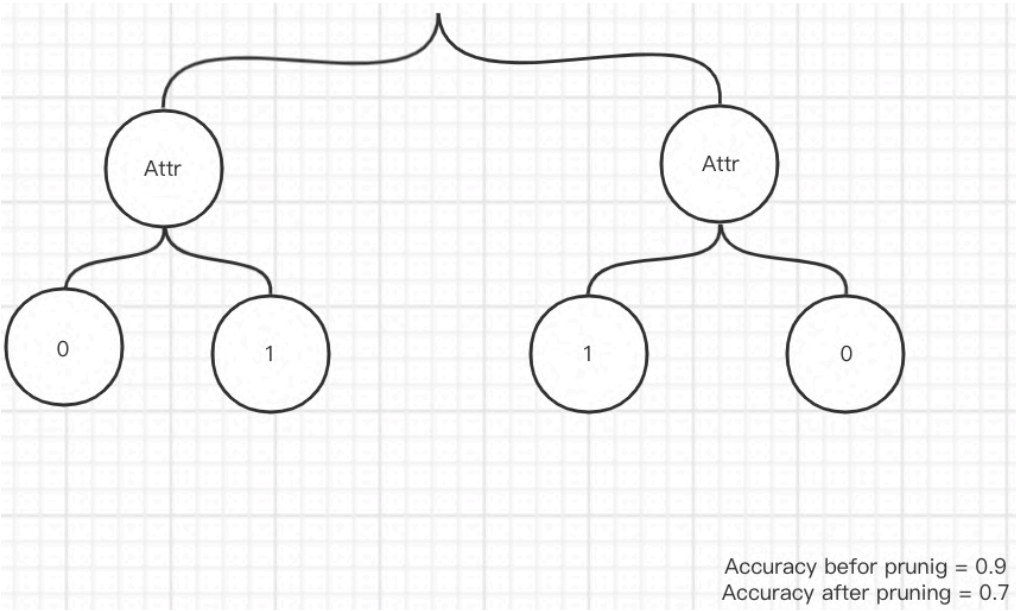

Figure 8 Pruned Tree with lower performance

Multiple Independent Classification with a Single Binary Tree

In order to predict multiple independent labels with a single binary tree, the learning algorithm can be adjusted based on the branch splitting strategy. Firstly, it trains on one label and regards the rest labels as a whole. When it reaches a leaf for that label, instead of generating a leaf and terminating splitting, it proceeds to classify samples in this node.

Suppose we have class A, B, C, we firstly regard A as positive class and B,C together as negative class. If the branch reaches the end and all the classes are A, then it becomes a leaf and stop splitting; if the branch indicates all the classes are B and C, it continues to split following the rule that B is positive and C is negative until B and C are completely separated.

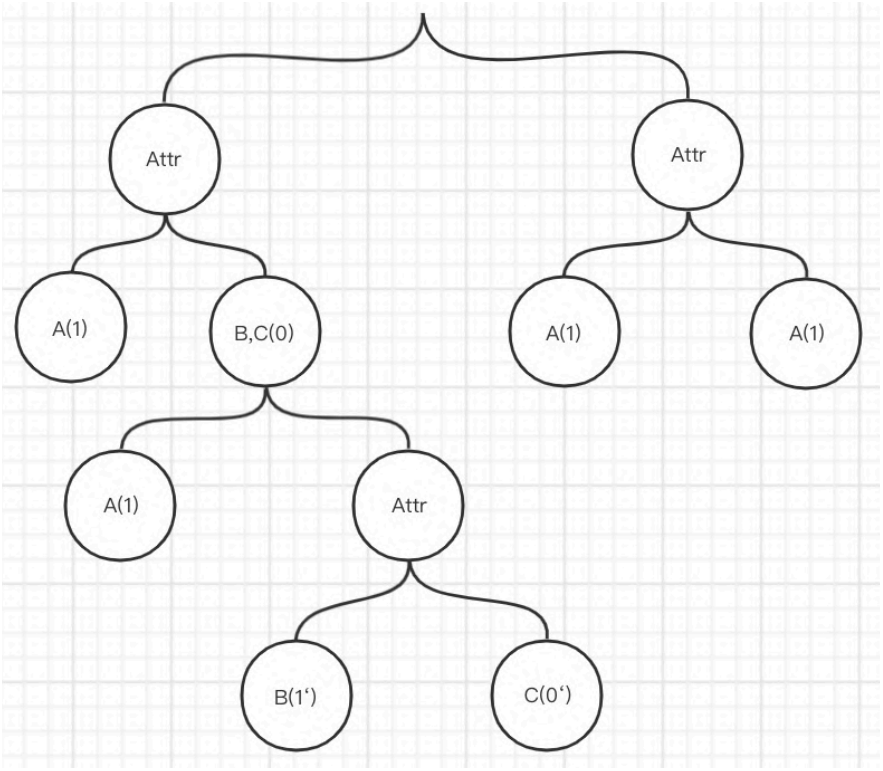

Figure 9

For the query search algorithm for each node, the method we adopt is to choose the best attribute and threshold for one label at a time. Besides, we would use some termination condition to reduce its structural complexity. Otherwise, this would likely be overfitting and lead to poor generalisation. For example, if the number of class A is larger than k times of the number of class B, where k is a threshold, we stop splitting.

Reference

[1] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Liang, X., Zou, T., Guo, B., Li, S., Zhang, H., Zhang, S., Huang, H. and Chen, S. X. (2015). Assessing Beijing's PM2.5 pollution: severity, weather impact, APEC and winter heating. Proceedings of the Royal Society A, 471, 20150257.