Programming Assignment #2

Build a Decision Tree

2013011457 컴퓨터전공

Jisoo Song

1. Summary of Algorithm

I implemented decision tree algorithm using ‘Information Entropy’ and ‘Information Gain’ theory.

First of all, my program parses given input file to train the tree, and save it as a form of a dictionary. That data has to be preprocessed, including creating metadata about attributes and dividing attributes into a set of deterministic attributes and a class attribute.

Next, the program calculates information gain of each attributes, and it selects an attribute that has the highest information gain. Then it divides training data by the selected attribute.

The program calculates information gain, selects an attribute, and divides training data using the divided data recursively until there’s no remaining attribute to divide or information gain of all attributes is 0.

After the decision tree built completely, the program predicts a class of given test data. Follow the decision tree by the value of each record in test data, and if there exists a leaf node, it returns predicted class by majority voting. If the record doesn’t arrive to a leaf node and there’s no child node which can go, it does majority voting using all the leaf node of current node. Then writes the predicted class on a result file.

1. Details about each Functions

1. parser



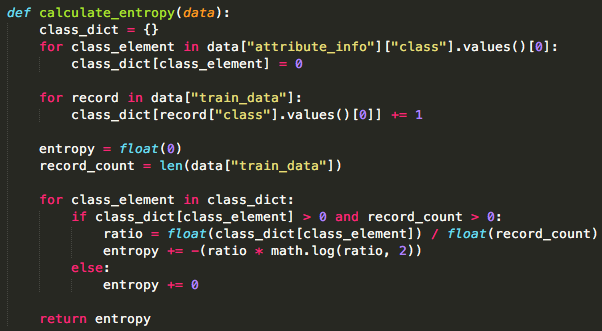
parser function parses input data and returns parsed data. It reads the first line and add it to the attribute list. Then It reads following lines and append the values to the dataset.

2. preprocessing



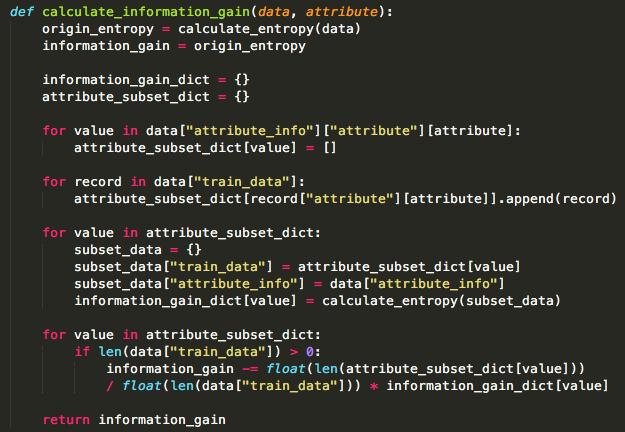
Preprocess function divides the attributes into a set of deterministic attributes and a class attribute.

3. calculate\_entropy



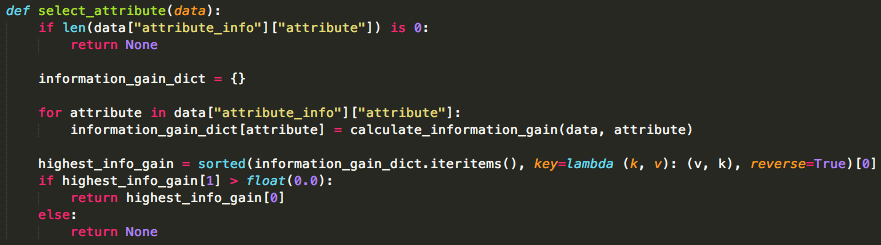
calculate\_entropy function calculates entropy of given dataset. It counts the record by each class value and calculates an entropy.

4. calculate\_information\_gain



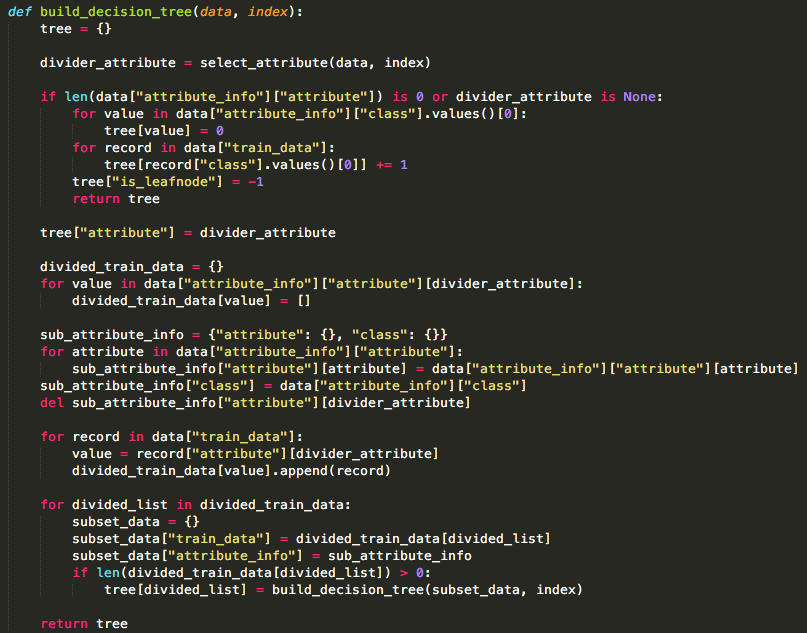
calculate\_information\_gain function calculate information gain using given dataset and attribute. It uses calculate\_entropy function to each value of given attribute.

5. select\_attribute



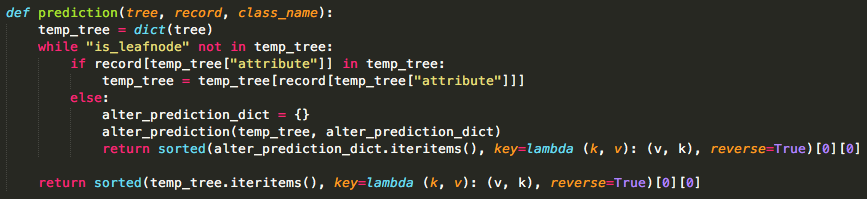
select\_attribute function calculates information gain of all the existing attribute and save it as the form of dictionary. Then it sorts the dictionary by the information gain, and it returns the attribute that has highest information gain. If information gain of all the existing attribute is 0, it returns None.

6. build\_decision\_tree



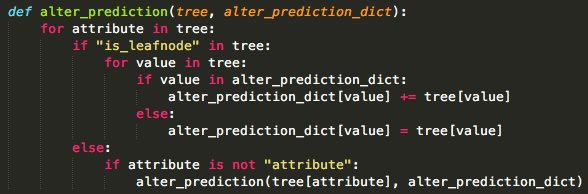
build\_decision\_tree is a recursive function. First it picks an attribute using select\_attribute function, and it divides given training dataset by that selected attribute. Then it calls the function itself recursively to each divided training dataset. If there’s no remaining attribute or select\_attribute function returns None, it makes leaf node and exits the function.

7. prediction



prediction function predicts the class of test data using built decision tree. If each data meets leaf node, it returns class value by majority voting on this leaf node. Or not, it calls alter\_prediction function.

8. alter\_prediction



prediction function calls alter\_prediction function in some cases. If a data doesn’t meet leaf node and cannot go to any next node, this function counts the class value of all the child node recursively.

1. How to Compile the codes

I implemented this code using Python 2.7. You can compile the codes like this:

/Users/songjisu/Desktop/스크린샷 2017-04-20 오후 9.22.26.png

My main function’s name is decision\_tree\_main.py. I followed the rules of arguments that written on the notice file.