Assignment 2
CS 6375: Machine Learning

Spring 2016

Due: 11:59 p.m., Friday February, 12 via eLearning

Inducing Decision Trees

In this homework you will implement and test the decision tree learning algorithm (See Mitchell, Chapter 3). You can use either C/C++, Java or Python to implement your algorithms. Your C/C++ implementations should compile on Linux gcc/g++ compilers. **Remember, you must write your own code.**

Building a Decision Tree model

The main idea in this homework is to implement the ID3 decision tree algorithm presented in class. You will use the *training dataset* to learn the tree, *validation dataset* to validate and prune the tree, and finally the *testing dataset* to test the tree. For the training phase, you can load the entire dataset in the root node and recursively find out the best attribute at each internal node.
The terminating condition can be one of the following:

- No more attributes to split on. In this case, you should find the majority class, and label it as that class. For example, if you reach a leaf node with 10 instances of class + and 2 instances of class -, it should be labeled class +.
- The node is pure i.e. it has only one class. In this case, it is easy to infer the class of instances.

Remember to use only the training_set files to learn the tree. Also, be sure to number the nodes of your tree as this will make it easier to identify which nodes to prune. For example, you can label the nodes as shown in figure 1

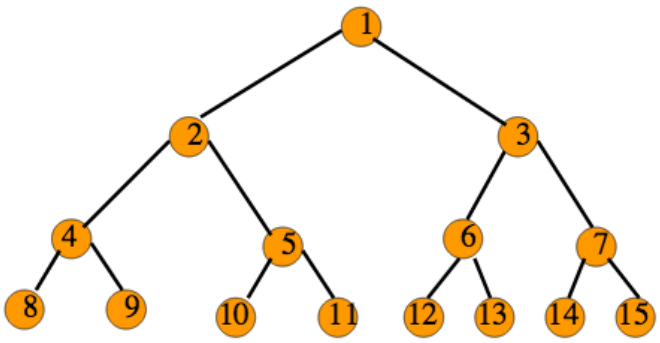

Figure 1: Example of labeling decision tree nodes

Data provided

You will be using the following datasets for programming and testing:

- Dataset 1 - Uploaded on eLearning. It's divided into 3 parts: training (for learning the model), validation (for pruning), test (for testing performance).
- Dataset 2 - Uploaded on eLearning. It's divided into 3 parts: training (for learning the model), validation (for pruning), test (for testing performance).

Pruning your model

After you have constructed your model i.e. decision tree, it is time to see if pruning would help your model. This means checking if there is any overfitting to the training data. You will do this by randomly deleting the subtree below a node. Note that if you delete a node, you delete all the child nodes of that node i.e. the sub-tree starting at that node, and not that node. In that case, that node becomes a leaf node. The classification at that node will be done by taking the majority vote e.g. if that node contains 10 instances of class 1 and 5 instances of class 2, then it will classify instances as

class 1.In this assignment, you will read in an argument that indicates how many nodes to prune. You will then randomly select that many number of nodes and remove the sub-tree below that node. For example, if the value of the argument is 3, then you will randomly delete sub-trees below 3 random nodes.

Outputting your model

Implement a function to print the decision tree to standard output. We will use the following format.

```
wesley = 0 :
| honor = 0 :
| | barclay = 0 : 1
| | barclay = 1 : 0
| honor > 0 :
| | tea = 0 : 0
| | tea = 1 : 1
wesley = 1 : 0
```

According to this tree, if wesley = 0 and honor = 0 and barclay = 0, then the class value of the corresponding instance should be 1. In other words, the value appearing before a colon is an attribute value, and the value appearing after a colon is a class value.

Testing your model

The final step would be to test your pruned model on the *test dataset*. This means finding the accuracy on the test dataset. The actual class labels for each of the instances in the test dataset are provided. You have to compute the accuracy as follows:

$$\text{Accuracy} = \frac{\text{Number of instances correctly classified}}{\text{Total number of instances}}$$

Running your program

Your code should be runnable from the command line. It should accept the following arguments (in-order)

1. Number of nodes to prune.
2. The path of the training set
3. The path of the validation set
4. The path of the test set
5. A Boolean (0 or 1) indicating whether to print the model (after pruning)

As an example,

```
<YourProgramName> 10 <training_file> <validate_file> <test_file> 1
```

Output of your program

It should output the accuracies on the test set for decision trees constructed.

What to Turn in

- Your code and a Readme file for compiling the code.

On the two datasets provided:

- Report the accuracy on the test set for decision trees constructed for 5 different values of "number of nodes to prune parameter". One of the values should be 0.