LING 572 Hw2
Due: 11pm on Jan 20, 2016

The example files are under /dropbox/2015-16/572/hw2/examples/.

Q1 (5 points): Run the Mallet DT learner (i.e., the trainer's name is DecisionTree) with **train.vectors.txt** as the training data and **test.vectors.txt** as the test data. In your note file, write down the following:

(a) The command lines you use for preparing data, training, testing, and getting the training and test accuracy; No need for condor submission scripts. You can use vectors2classify commands to do training, testing and evaluation in one step.

(b) The training accuracy and the test accuracy.

Q2 (10 points): Does the Mallet DT learner treat the features in the training and test files as binary or real-value? Prove your answer by writing a tool that binarizes the features and then comparing the classification results with or without first binarizing the features. To binarizing a feature, just replace a non-zero value with value 1, and keep zero values unchanged. In your note file, write down the commands you use for binarization, training, testing, and comparison. No need for condor submission scripts for Q2.

Q3 (10 points): Run the Mallet DT trainer with different depths; that is, when running vectors2classify, replace **–trainer DecisionTree** with

```
--trainer "new DecisionTreeTrainer(nn)"
```

where nn is the depth of the decision tree. Note that you have to use vectors2classify, instead of "mallet train-classifier" and "mallet classify-svmlight" because "mallet train-classifier" does not process "new DecisionTreeTrainer(nn)" properly.

(a) Fill out Table 1

(b) What conclusion can you draw from Table 1?

Q4 (50 points): Write a program, **build_dt.sh**, that builds a DT tree from the training data, classifies the training and test data, and calculates the accuracy.

- For coding in all the assignments, you can use any programming languages that run on Patas. You just need to write a shell script, build_dt.sh in this case, as a *wrapper*. If you do not know what a wrapper is, please check dropbox/14-15/572/code-examples/ and the slides for hw2.

Table 1: Decision Tree with different depths

Depth	Training accuracy	Test accuracy
1		
2		
4		
10		
20		
50		
100		
1000		

- This DT learner should treat all features as binary; that is, the feature is considered present if its value is nonzero, and absent if its value is zero.
- Use information gain to select features when building DT.
- The format of the command line would be: build_dt.sh training_data test_data max_depth min_gain model_file sys_output > acc_file
- training_data and test_data are the vector files in the text format (cf. **train.vectors.txt**).
- max_depth is the maximum depth of the DT,[1] and min_gain is the minimal gain. Those parameters are used to determine when to stop building DT; that is, split the current training data set at the node x if and only if the depth of x < max_depth AND the infoGain of the split $\geq$ min_gain.
- model_file shows the DT tree (cf. **model_ex**). Each line corresponds to a leaf node in the DT and it has the format: path training_instance_num c1 p1 c2 p2 ...
  Where path is the path from the root to the leaf node, training_instance_num is the number of the training examples that "reach" the leaf node, c_i is the class label, and p_i is the probability of c_i (i.e., the percentage of the training examples at the leaf node with the label c_i).
- sys_output is the classification result on the training and test data (cf. **sys_ex**). Each line has the following format:
  instanceName c1 p1 c2 p2 ..., where instanceName is just something like "array:0", "array:1".
- acc_file shows the confusion matrix and the accuracy for the training and the test data (cf. **acc_ex**). In the confusion matrix, a[i][j] is the number of instances where the truth is class i, and the system output is class j.
- As always, model_ex, sys_ex, and acc_ex are NOT gold standard. These files were created just to show you the format of the files.

Q5 (20 points): Write a condor submit script, build_dt.cmd, with **train.vectors.txt** as the training data and **test.vectors.txt** as the test data. Fill out Table 2 (where min_gain is set to 0) and Table 3 (where min_gain is set to 0.1).

[1]The depth of the root is 0, the depth of its children is 1, and so on.

Table 2: Your decision tree results when min_gain=0

Depth	Training accuracy	Test accuracy	Wall clock time (in minutes)
1			
2			
4			
10			
20			
50			

Table 3: Your decision tree results when min_gain=0.1

Depth	Training accuracy	Test accuracy	Wall clock time (in minutes)
1			
2			
4			
10			
20			
50			

Q6 (5 points): Slide #12 at http://courses.washington.edu/ling572/winter2016/teaching_slides/class3_DT.pdf shows a DT: f1 and f2 are two features; f1 is in [-20, 30]; f2 is in [-10, 30]. L_i (i=1, ..., 7) represents a leaf node. Each leaf node corresponds to a rectangle in a 2-dimension space, where f1 is the x-axis and f2 is the y-axis. Draw a graph that shows the boundary of the seven rectangles in this 2-dimension space.

Submission: Submit a tar file via CollectIt. The tar file should include the following.

- You can choose to form a team, a team can have at most two people. In that case, either student in the team can submit to CollectIt, but please submit only one copy. In your note file, please list the names of team members.
- In your note file hw2-notes.*, include your answers to Q1-Q3 and Q5-Q6, and any notes that you want the TA to read.
- Under a subdirectory q2/ includes related source/binary code for Q2.
- Under a subdirectory q4/ includes source/binary code and the shell script for Q4.
- Under a subdirectory q5/ includes the condor submit script and data files produced in Q5 when the min_gain is 0.1 and the max_depth is 50. For naming convention, acc_file for it can be called **acc_file.50_0.1**, and sys and model files can be named similarly.