CS478 Machine Learning
Spring 2000
Assignment 2

Due electronically by Tuesday, March 7, 11 a.m.

Problem Set Portion (45 pts)

1. Exercise 3.1 in *Mitchell* (10 pts).
2. Exercise 3.4 in *Mitchell* (25 pts).
3. (10 pts) Discuss the merits and drawbacks of each of the following two schemes for handling multiclass data sets:
   (a) Form a single decision tree for all classes.
   (b) Form one decision tree for each class. To train a decision tree for class C_i convert the data into a two-class problem using the following rule to relabel each instance: if the class is equal to C_i then set label to +, else set label to -. To classify an instance using this scheme you use each class's tree to classify it and then output all class names whose corresponding tree predicted a +.

Programming Portion (55 pts)

1. Modify your ID3 code from assignment one to allow it to be run in the following manner. The program should request three items from the user:

   DATA_FILE**:** This will contain a set of instances of the same format as assignment one.

   TRAINING_PERCENTAGE**:** This will be an integer between 1 and 100 that indicates the percentage of instances to use as training.

   TESTING_PERCENTAGE**:** This will be an integer between 0 and 100 that indicates the percentage of instances to use as testing.

   The program should then read in the instances from the DATA_FILE, randomly select TRAINING_PERCENTAGE instances from it, and use those instances to train the decision tree. Augment the trace information from assignment one to report **the number of nodes in the tree** and to include at the top of the trace **the values for the three arguments to the program**.

   If TESTING_PERCENTAGE is greater than zero, then test the learned decision tree on TESTING_PERCENTAGE randomly selected instances from DATA_FILE. Note that there is no separate test file. Ensure that there is **no overlap** in the training and test sets. Assume that the user will request only reasonable training/test set breakdowns, i.e. the sum of TRAINING_PERCENTAGE and TEST_PERCENTAGE will be less than or equal to 100. It is possible that during testing, the ID3 code will encounter attribute values that did not occur during training. You do not need to handle this and can abort any runs that encounter the problem. You do not need to turn anything in for this part.

2. (15 pts) *****IMPORTANT***: This question has been promoted to an EXTRA CREDIT question. It is not part of the main assignment.** This question lets you experiment with your new version of ID3 using the sample *PlayTennis* learning data from Table 3.2 in *Mitchell*. The PLAYTENNIS.DATA data set is available from the course web page. Run ID3 on this data set to create a decision tree for all of the examples. There will be no test data. Thus, TRAINING_PERCENTAGE is 100 and TESTING_PERCENTAGE is 0.

   Answer each of the following statements as true or false. If the statement is true, augment the PLAYTENNIS.DATA data set with an instance that demonstrates it. Label your new data set as PLAYTENNIS-A.DATA, PLAYTENNIS-B.DATA, or PLAYTENNIS-C.DATA accordingly. Also save the corresponding trace file, e.g. PLAYTENNIS-A.TRACE. If the statement is false, you should explain why (and not turn in a data set or trace file).

   Assume that (1) the target concept is the concept described by the decision tree of Figure 3.1 of the text, and (2) all training examples you add *must be consistent* with this target concept.

   (a) **True or false:** It is possible to get ID3 to further elaborate the tree below the rightmost leaf in Figure 3.1 (and make no other changes to the tree), by adding a single new (correct) training example to the original fourteen examples.

   (b) **True or false:** It is possible to get ID3 to learn an *incorrect* tree (i.e. a tree that is not equivalent to the target concept of Figure 3.1) by adding new *correct* training examples to the original fourteen.

   (c) **True or false:** It is possible to get ID3 to include the attribute *Temperature* in the learned tree, even though the true target concept is independent of *Temperature*.

3. (20 pts) This part part of the assignment will let you experiment with a larger set of data describing voting records of congressmen/women from the U.S. House of Representatives. This data set, VOTING.DATA is available from the course home page and is in the usual format. The attributes of each example indicate the member's yes/no/absent vote for each bill considered by congress. The attribute to predict is the political party (Democrat, Republican) of the representative, based on his/her voting record.

   (a) **Variance due to different training sets:** Use the voting data to train a decision tree, with a TRAINING_PERCENTAGE of 25 and a TESTING_PERCENTAGE of 75. Rerun this experiment several times and notice the impact of different random splits of the data into training and test sets. Report the sizes and accuracies of these trees over **five** distinct runs.

   (b) **Effect of training size:** Measure the impact of training set size on the accuracy and the size of the learned tree. Use 30% of the data for testing. Consider training set sizes in the range 0-40% (include at least the values 2, 10, 20, 30, and 40 for training percentages). Because of the high variance due to random splits, repeat the experiment **ten** times for each training set size. Determine the mean, the maximum, and the minimum accuracies at each training set size. Also measure the mean, the maximum, and the minimum tree size at each training set size. Turn in two plots that indicate:

      i. How accuracy varies with training set size

      ii. How the number of nodes in the final tree varies with training set size

4. (20 pts) ID3 uses information gain as its splitting criterion. Modify your ID3 code to instead select attributes at **random** and study the effect of this change. In each of ten runs, use 30%

of the data for training and 30% for testing. Compare the results you saw in steps 3a and 3b above. What is the impact of learning with randomly selected attributes?

Electronic Submission

All of the files for this assignment should be submitted electronically as a single **zip archive** or **tar** file. Electronic submission is done through an interface available at the CS478 home page. Any additional instructions for electronic submission will be posted there.

Files to Turn In

1. The **source code** and **executable(s)** for the decision tree program. Include all files that go with it, such as .h files. If you are using C or C++, please include a **Makefile**. The extension of the source files should indicate the programming language in which the code was written (e.g. dtree.c or dtree.scheme).

2. A **README** file, which includes information on how to run your program.

3. **(For the extra credit question only.)** A trace of your program for the **true** statements in problem 2, when run on your new data sets. Trace files should be named with a ".trace" extension.

4. **(For the extra credit question only.)** The new data sets you created for the **true** statements in problem 2.

5. Your answers to the written problem set (1, 2, and 3). The answers to the questions may be turned in as an ASCII text file (**assign2-written.txt**), a postscript file (**assign2-written.ps**), or a pdf file (**assign2-written.pdf**). Other file formats will not be accepted.

6. Your answers to the programming problem set (3, 4, and 2-**extra credit**). The answers to the questions may be turned in as an ASCII text file (**assign2-prog.txt**), a postscript file (**assign2-prog.ps**), or a pdf file (**assign2-prog.pdf**). You may submit the plots in separate files; if so, name them **voting-treesize.ps** and **voting-accuracy.ps** (or **.pdf**).