10-601 Machine Learning: Homework Assignment 1

Professor Tom Mitchell
Carnegie Mellon University
Updated on January 13, 2009 (originally posted Jan 12)

- The assignment is due at 1:30pm (beginning of class) on **Wednesday, January 21, 2009**.
- Write your name at the top right-hand corner of each page submitted.
- Each student must hand in their own written answers to the following questions (in hardcopy), and their own code (electronically). To submit your code, please send it as an attachment via email to `acarlson` AT `cs.cmu.edu`. Package your code as a gzipped TAR file or a ZIP file with the prefix `601hw1-jdoe` where you substitute in your first initial and last name into the filename in place of '`jdoe`'. See the course webpage for the collaboration policies.
- See the course Homework 1 page to download the data

Decision Trees

The data: In this assignment you will train a decision tree to predict votes of US Congressmen based on their political party and on other votes they have made in the past. This is based on a data set from the UC Irvine data repository. The data contains a total of 435 examples, one for each member of the US House of Representatives. Each example is described in terms of 17 attributes including "party" (democrat or republican), and 16 votes made by this congressperson, such as "antiSatelliteTestBan" (with values "y", "n" and "noVote"). These are in comma-separated-file format, with the first line in the file listing the attribute names (separated by commas) and each remaining line in the file giving the values of these attributes for a single congressperson.

You must train your decision tree classifier for two different classification problems (you are free, of course, to try others and will receive extra credit if you answer the questions below using more than two problems). The two decision trees you must train are:

1. a tree to predict "party" (democrat or republican) based on the 16 votes of the congressperson
2. a tree to predict the vote on "physicianFeeFreeze" based on the other votes and party.

1 Implement the Basic Decision Tree Learning Algorithm

Implement the ID3 decision tree learning algorithm described in class (without post-pruning). You may use any programming language you like. To make life easy, your program can assume that all features/attributes take on only discrete values (no real-valued attributes), that the data contains no missing attributes, and that legal values for each attribute are known in advance. Of course your code should assume the target attribute to be predicted can take on more than two possible values (e.g., "y","n" and "noVote").

2 Train and Test Your Basic Algorithm

Run your algorithm on the training data to learn a decision tree for both "party" and "physicianFeeFreeze". *Answer this and all of the remaining questions, reporting and analyzing data for*

both of these classification tasks. For each classification task, what is your learned decision tree's accuracy over the training set? Over the test set? Suggest why these accuracies might differ in the ways you observe.

3 Implement and Test Post-Pruning

Implement a post-pruning procedure for your learned decision tree. You may choose reduced error pruning, rule post-pruning, or any other method you choose so long as you describe it precisely. Train your tree on the *train.csv* training data, then prune it using the *prune.csv* pruning data.

1. What are the sizes of your original tree and your pruned tree (Note: if you use rule post-pruning, make up a reasonable definition of "size")?
2. What are the accuracies of your unpruned and pruned trees over the training set? Over the pruning set? Over the test set?
3. Which of these trees (pruned or unpruned) would you recommend using to classify future data (justify your answer in terms of your actual observed accuracies).

4 Why Use a Separate Pruning Set?

1. In the above pruning step you used an independent set of pruning data rather than the test data. Explain why this is better than using the test data for pruning if we want the best possible estimate of the accuracy of our final decision tree. (Feel free to conduct the experiment to find out what happens if you use the test data for pruning.)
2. Suppose we want to optimize the accuracy of the learned decision tree, and are willing to forgo having a precise estimate of its actual accuracy. How would you recommend splitting the available examples into training, pruning and testing sets in this case?

5 OPTIONAL Extra Credit

(This question is optional, and you need not answer it.) In the above analysis, we took the available examples and allocated some for training and some for pruning. Among those used for training and pruning, there are many different ways to split up the data. Can you find a better way to utilize this set examples for training and pruning (i.e., can you train a tree that achieves higher accuracy over the test examples)?