

CS 410/510
Machine Learning
Winter, 2006

Homework 2: Decision Trees

Due Tuesday, January 24.

In this homework assignment you will experiment with C4.5, an implementation of ID3 written by Ross Quinlan and available on the Internet.

You will be working with your training examples of spam and non-spam e-mail messages, and the features you extracted from these examples in Homework 1, as well as with a larger database of feature values from a spam dataset given in the UCI Machine Learning Repository.

For this homework, you need to perform the following steps:

1. Download C4.5: Download C4.5 and documentation from the following web site:

<http://www2.cs.uregina.ca/~hamilton/courses/831/notes/ml/dtrees/c4.5/tutorial.html>

This version of C4.5 works on Linux systems. If you have any trouble getting it to compile and work, let me know.

For this assignment, you will need only the manual pages for “c4.5” and for “verbose c4.5”. For the format of the input files, see one of the sets of example files given on this web page (e.g., “Golf”).

2. Create training and test data sets for your spam data: Your training data should be the set of feature values from your spam and non-spam data from Homework 1. Create a test set of spam and non-spam messages by running your feature extraction program on the spam test examples and non-spam test examples that are now on the class web page. Put these files in the format required by C4.5. Your files should be named something like *spam.data* and *spam.test*. You also need to create a *spam.names* file that gives the names and types of your attributes (see the examples on the C4.5 web page).

3. Run C4.5 on your spam data: (a) Run C4.5 on your training set, and test the resulting trees on your test set. This can be done with the command:

```
c4.5 -f spam -u
```

Put the following information in your writeup:

- A hand drawing (as a tree) of the unpruned tree (and the pruned tree if it differs from the unpruned tree)
- The training error and test error from both the unpruned and pruned tree (if different).

(b) Now run c4.5 using information gain instead of gain ratio:

```
c4.5 -f spam -u
```

and give the same information as above in your writeup.

4. Calculate the information gain and gain ratio of root: Calculate by hand the information gain and gain ratio of the root node in the unpruned tree from step 3. Show your work in doing these calculations. If the root attribute is continuous, use the threshold found by C4.5 in your calculation of the gain and gain ratio. For example, if the root is the continuous attribute *message_length*, and c4.5 used $message_length > 400$ or $message_length \leq 400$ as the split for this attribute, consider these two ranges to be the two values of this attribute.

5. Add an attribute: What additional attribute do you think would improve the performance of C4.5 on the test data? Add it to your set of attributes, and repeat steps 3(a) and (b).

6. Download UCI spam data: Now download the files “UCI-spam.names”, “UCI-spam.data”, and “UCI-spam.test” from the class web page. These files give a large set of examples of spam and non-spam, using 57 different features, from the UCI Machine Learning Repository. I have divided it into 3681 training examples (1454 spam and 2227 non-spam examples) and 920 test examples (359 spam and 561 non-spam). The file “UCI-spam.names”

gives the names of the attributes, which are (hopefully) self-explanatory. *word_freq_[word]* gives the frequency of the given word in the message. *char_freq_[char]* gives the frequency of the given character in the message. *capital_run_length_[statistic]* gives a statistic concerning runs of capital letters in the message. (Note that we don't have the actual messages these values are taken from, just the values.)

7. Run C4.5 on the UCI spam data: Repeat steps 3(a) and (b), but with the UCI spam data (except you don't have to draw the trees for this or any of the following steps).

8. Effect of training-set size: (a) Using examples from UCI-spam.data, create a new training set that is one-half the size of UCI-spam.data, but has approximately the same ratio of spam to non-spam as in UCI-spam.data. Repeat steps 3(a) and (b) using this new training set, and UCI-spam.test as the test set. (b) Do the same, but this time with a training set that is one-fourth the size of UCI-spam.data, still using the entire UCI-spam.test as the test set.

9. Effect of noise: (a) Take your one-fourth-size training set from step 8b, and add 5% noise, by choosing 5% of the spam and non-spam examples at random and changing their class value to the opposite class. Repeat steps 3(a) and (b) using this new noisy training set (still use the entire UCI-spam.test as your test set). (b) Repeat step 9(a) but with 10% noise. (c) Repeat step 9(a) but with 20% noise.

10. Summarize your results: Write a summary of your results from the steps above. In particular, what are the effects of training set size and noise on the accuracy of decision trees? Explain why you think these effects occurred. What are the effects of using information gain versus gain ratio in the various experiments you did? Why do you think you got these effects? Include anything else that you found noteworthy in your results.

Grading

You will be graded on the completeness and clarity with which you report and explain your results on the steps listed above.