CS565: Data Mining
Programming Assignment 2

Due Date: 20th November, 2007 at 11:59 PM.

Aim of the assignment:
The aim of this assignment is to implement the ID3 and Naïve Bayesian Classification methods and evaluate their performance over a dataset that we will provide to you.

Implementation Guidelines:
Implementation platform: Your program should be compile-able and executable in Unix.
Implementation language: It is highly recommended that you use C, C++ and Java. The grading of this assignment will be based on the correctness of the algorithms and not on the performance.

Dataset:
You will use a dataset that has two classes and 21 categorical attributes. The dataset is split into training and test sets. You will use the training set to build your classifier and the test set to evaluate its accuracy. You can download the dataset that will be used to train and evaluate your methods from there:
http://www.cs.bu.edu/faculty/gkollios/dm07/P2
For more information about the dataset please use the "Readme" file.

A. Basic versions of ID3 and NB (80%)

Implement ID3, exactly as described in Figure 6.3 of your textbook, at page 293. Note that ID3 uses multi-way splits for categorical attributes. Implement Naïve Bayes (NB) as explained in Section 6.4.2 of your textbook.

1. ID3: Your program should have options to print the classification rules that can be generated by the decision tree. AN example output is the following:

```
Total number of rules: 950
Rules:
IF odor=a THEN class=e
IF odor=l THEN class=e
…
IF odor=n AND habitat=g AND gill-color=k AND cap_shape=b THEN class=e
…
```

2. Compare the accuracy of ID3 and NB using the test dataset. The accuracy is defined by the number of correctly classified samples divided by total the number of samples.

B. ID3 with prepruning (20%)

Modify your ID3 program to take a *minimum information gain threshold* parameter *t*, which can be used to control the size of the tree as follows. After the line:
(6) select *test-attribute,* the attribute in *attribute-list* with the highest information gain;
add a line
(6b) if *igain*(*test-attribute*)<*t* then goto line (5); //return a leaf node using majority voting

Basically, if the highest information gain is not at least *t* we want to terminate the branching there and do not expand the tree further. A leaf is placed using majority voting to determine the class label.
Call this algorithm ID3_prepruning. Create a table, like the one below, which shows the classification accuracy and the number of generated rules for several values of *t*:

t	0	0.005	0.01	0.04	0.05	0.1	1
Accuracy	0.944	…	…	…	…	…	…
No. of rules	950	…	…	…	…	…	…

Explain your results. Does, in general, accuracy increases when *t* increases? Why? Why the number of rules decreases when *t* increases?

What to hand-in:

1. You source code and a readme file that explains how to compile and run your code.
2. A report that presents and explains your results. Also, you should discuss any assumption that you make or any optimization that you used (if any) to improve the performance of the classifiers.