CSCI 6350 Spring 2006
Project 2 Classification – part 1(Due: beginning of class, March 14th)

In this project, we experiment with two classification schemes: decision tree classification and Naive Bayes classification. Different classification methods are typically evaluated in terms of how accurate a classifier built based on the training data predict the class labels for the testing data, which is disjoint from the training data. 10-fold cross validation is used for this purpose.

For decision tree classification, we will use a standard package, called C4.5. Detailed description of how to download, compile, and run the program is given below. For Naïve Bayes classification, you are required to write your own program for the experiment (part 2). A detailed description of this part of the project will be given later. Your program can assume that the data is described by all discrete-valued attributes. It should be able to handle missing attribute values in data. You need to think of a strategy for dealing with missing attribute value, and write your program based on your chosen strategy.

Part 1 : Classification with Decision Tree

1. Data

You need to find two data sets for this experiment. A place to look for data is the machine learning repository at ftp://ftp.ics.uci.edu/pub/ml-repos/machine-learning-databases/. There are about 100 data sets in the repository. The "SUMMARY TABLE" file contains a summary of all data sets in the repository. Descriptions of individual data sets are given in "data.names" file in each data's directory. It describes the documented past usage of the data, the attributes and the valid values of the attributes used in data, and the characteristics of the data, e.g., number of examples, whether there are missing attribute values, etc.. two data sets that you are interested in, and download the data to your own directory.

- The first data you select should contain *both discrete and continuous* valued attributes.
- The second data you select should contain only discrete valued attributes (may also contain missing attribute value). You can always convert a complete data set to one that contains missing attribute values by removing the attribute values for some data objects.

2. Program: Decision tree classification -- C4.5

Copy the C4.5 package from the course web page into your own directory.

Unzip and unpack the files by:
Frank% gunzip c4.5r8.tar.gz
Frank% tar -xvf c4.5r8.tar

Compile the program by :
Frank% cd R8/Src
Frank% make all
This generates the four executable files: c4.5, c4.5rules, consult, and consultr:

c4.5	generates a decision tree from a file of examples
c4.5rules	generates production rules from unpruned decision trees
consult	classifies items using a decision tree
consultr	classifies items using a rule set

Create your own directory for experiments:
Frank% cd ..
Frank% mkdir my-exps

Move all executable files into that directory:
Frank% mv c4.5 c4.5rules consult consultr xval-prep average xval.sh ../my-exps

C4.5 file format

Each data set has to have at least two files created: **filestem.names** and **filestem.data**.
filestem is the name used for all files related to a data set, including: training data, testing data, decision trees generated, decision rules generated, cross validation results, etc..

Names file : "filestem.names"

It contains a series of entries defining the names of the features, feature values, and classes used in the data. The file is free-format with the exception that the vertical bar `|' causes the remainder of that line to be ignored. Each entry is terminated by a period which may be omitted if it is the last character of a line.

The file commences with the names of the classes, separated by commas and terminated with a period. Each name consists of a string of characters that does not include comma, question mark or colon (unless preceded by a backslash). A period may be embedded in a name provided it is not followed by a space. Embedded spaces are also permitted but multiple whitespace is replaced by a single space. The rest of the file consists of a single entry for each attribute. An attribute entry begins with the attribute name followed by a colon, and then either the word `ignore' (indicating that this attribute should not be used), the word `continuous' (indicating that the attribute has real values), the word `discrete' followed by an integer n (indicating that the program should assemble a list of up to n possible values), or a list of all possible discrete values separated by commas. (The latter form for discrete attributes is recommended as it enables input to be checked.) Each entry is terminated with a period.

Example (golf.names)

```
Play, Don't Play.

outlook: sunny, overcast, rain.
temperature: continuous.
humidity: continuous.
windy: true, false.
```

Data file : "filestem.data"

It contains one line per example. Each line contains the values of the attributes in order followed by the example's class, with all entries separated by commas. The rules for valid names in the names file also hold for the names in the data file. An unknown value of an attribute is indicated by a question mark `?'.

Example (golf.data)

```
sunny, 85, 85, false, Don't Play
sunny, 80, 90, true, Don't Play
overcast, 83, ?, false, Play
rain, 70, 96, false, Play
```

3. Experiments

3.1 For each data, build the decision tree and rules with these steps.

Build the decision tree based on the training data by :
Frank% ./c4.5 -f filestem

Extract the set of rules by:
Frank% ./c4.5rules -f filestem

Analyze the experimental results:

- What is the decision tree derived?
- What are the decision rules derived?
- Was pruning performed?
- Do the rules obtained agree with or disagree with what you think about the domain?
  If disagree, what is your hypotheses for the disagreement?
  If agree, what are the rules that agree with you?
- Are there any "interesting"(or unexpected) rules discovered?

3.2 Classify new data using decision tree and rules:

Frank% ./consult –f filestem
Then type in the values of the attributes when prompted.(repeat with multiple data)

Repeat the classification task by classifying new data using decision rules extracted.

- For the testing examples, do the classification predicated by the decision tree agree with your classification of the examples? Explain.

3.3 Compute the performance of decision tree algorithm on these data using N-fold cross validation (N=10)

Frank% csh xval.sh filestem 10
Examine the results stored in filestem.rres.

4. Things to turn in:

- Write a mini-report on this experiment. It should contain:
  - a paragraph describing the classification task
  - a paragraph describing the decision tree classification method.
  - A description of the two data sets used in the experiment.
  - A description of how the experiment will be carried out, include:
    - 10-fold cross validation procedure
  - Experimental results and analysis of the these results, include:
    - the rules derived
    - Classification results generated with new data
    - 10 fold cross validation result
    - analysis