CS475 - Assignment 8

(Basic Categorical Decision Trees)

REWARD: 250 points

DUE: Tue May 3 at 11pm PT

TEST DATA: testDataA8.tar or testDataA8.zip
LIBRARY: Matrix Version 7.51 mat.tar, mat.zip
Tree Version 2.0 tree.tar, tree.zip

1 the task

Write a C++ program called **id7** that builds a decision tree and then queries the tree to find expected categories. Use the id7 starter code to get a strong start on the problem. I have given you all the entropy and gain calculations and lots of other goodies. This algorithm is based on the id3 algorithm in the book, but is slightly different in that it has default values and makes some other choices, see the starter code. Learn how things are called and data structures arranged by looking at the starter code. You still need to create some tricky code so plan your time. I put in diagnostic print statements to show what it is doing as works so you can create those diagnostics. You must match the diagnostics as well as the tree and computed categories!

I will invoke id7 to read data from standard in as done by the algorithm demonstrated in class (see Input Format below). It takes no arguments on the command line. The algorithm is simple and recursive, but it takes some thought. Important: your algorithm must compute information gain in the order the features were given so you always generate the same tree given the same data. The order makes a difference when you maximize the information gain by saving the first occurrence of maximum gain. I will test this in the tests.

2 Input format

The program takes a data file from standard input in the form of three matrices that match our matrix library input. The matrix library and a simple tree object can be found in these archives:

Here is the layout:

```
numfeatures maxnumfeaturevalues
feature1 numvalues value1 value2 value3 ...
feature2 numvalues value1 value2 ...
feature3 numvalues value1 value2 value3 value4 ...
...
ans ans1 ans2 ans3
numtrainingexamples numfeatures
f1 f2 f3... ans
f1 f2 f3... ans
f1 f2 f3... ans
f1 f2 f3... ans
f1 f2 f3... ans
f1 f2 f3... ans
f1 f2 f3... ans
numqueryexamples numfeatures
q1 q2 q3... ans
q1 q2 q3... ans
q1 q2 q3... ans
q1 q2 q3... ans
```

Be sure to examine the test examples to be sure you see what is going on. Numfeatures is the number of features including the answer which must be labeled ”ans”. Then comes numfeatures lines of features value descriptions the last of which must be “ans”. Each line is a feature name followed by a list of strings that represent the possible values for the feature.

Example Input File:

```
5 4
Outlook  Sunny Overcast Rain
Temp     Cool  Warm     Hot
Humidity Dry   Humid    -
Wind     Calm  Breezy   Windy
Ans      Play  Noplay   -
6 5
Sunny    Hot  Humid Calm  Play
Sunny    Hot  Humid Windy Noplay
```

```
Overcast Hot  Humid Calm  Play
Rain     Warm Humid Calm  Play
Rain     Cool Humid Calm  Play
Overcast Cool Dry   Windy Play
3 4
Sunny    Hot  Humid Calm
Sunny    Hot  Humid Windy
Overcast Hot  Humid Calm
```

The number of feature values for a given feature may be variable. The matrix is filled out the the maximum number of feature values across all the rows with missing feature values replaced with the string ”-”. See examples. This matrix can be read in with the readstrings method in matrix.

The next matrix is data used to construct the tree and can be thought of as training data.

Finally, the third and last matrix is the query data. These are rows that are to be classified using the tree you built.

For turning in your program you must generate the exact tree and indenting character by character. I have supplied a helpful tree object in with the matrix download. As always if you get the exact right answer you will get a congratulations message. If not you will get the sdiff with the expected output.

3 Submission

Homework will be submitted as an **uncompressed** tar file to the homework submission page linked from the class web page. Be sure to include a make file to build your code as well as the matrix and tree libraries. You can submit as many times as you like. **The LAST file you submit BEFORE the deadline will be the one graded.** For all submissions you will receive email at your uidaho.Edu mail address giving you some automated feedback on the unpacking and compiling and running of code and possibly some other things that can be autotested.

Have fun.