CSE 546 Machine Learning Winter 2012
Assignment #1 – Decision Trees
Due: Friday January 20, 2012 by 5pm.

What you'll need:

- Code files:
    - decisionTree.py
    - ./www/ {many files for visualizing the tree}

- Data files:
    - republican.py
    - cars.py

What to turn in:

- Your code and electronic copy of your write up of the questions for each step in this document to Dropbox.
    - Your code can be in any language you want. We give you some python code framework, but it is fairly minimal.
- A write up, in hard copy, which you can bring to Lydia Chilton's office – CSE 605.

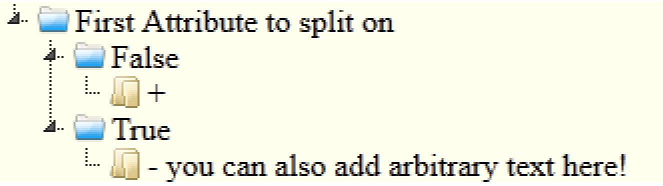
Instructions:

Goal: In this assignment you will implement the ID3 algorithm for creating decision trees. You will use information gain to decide the "best attribute" and chi-squared metric for deciding whether a split has statistical significance or not. You will evaluate your tree on test data with the misclassification impurity metric. There are two datasets to run and evaluate your code on: one to predict whether somebody is a republican or not, and one to predict whether a car on craigslist is "acceptable" or not. For this assignment, you may assume all attributes are binary (represented as 0 or 1).

Step 1. (Warm up)

Run the python script **decisionTrees.py**
This should create a file called **./www/decisionTree.js**
If you open **./www/decisionTree.html** in a web browser, you will see a (admittedly non-optimal) visualization of a tree.

Cross your fingers, say a prayer to the Gods of cross-browser compatibility and hope your tree looks like this:



First Attribute to split on
False
+
True
- you can also add arbitrary text here!

This is a data format called JSON which we will be using. There is a short reference at the beginning of the file **decisionTrees.py** .

You do not need to put any of this is your write up.

Step 2. Implement decisionTrees.py

Fill in holes in the code in decisionTrees.py (or start from scratch in a language of your choice)

- id3(examples, targetAttribute, attributes)
- findBestAttribute(examples, targetAttribute, attributes)

Follow the ID3 algorithm pseudo code on p.56 of Mitchell's Machine Learning book (see handout). Remember, for this assignment, you may assume all attributes are binary. (You’re welcome.)

Implement information gain to decide the “best attribute.” This is described in the Mitchell handout as well.

In this step, you do not need to implement split stopping; build the full tree.

When you have done this, you should be able to run the following code:

```
tree = id3(republican.trainingData1, "republican", ["salary more than $100,000", "owns
your own business","listenes to NPR","owns a truck","lives in a red state","watches
The Daily Show"])

writeHierarchyToJSFile(tree, 'www/decisionTree.js')
```

You can visualize the tree in your browser (see Step 1).

A note about the dataset republicans: all datapoints appearing in this dataset are fictitious. Any resemblance to real persons, living or dead, is about as coincidental as resemblances to real people on shows such as Law & Order.

<u>For your write up</u>

A. Include a picture of the tree (it should be modestly sized)

B. Answer the following questions in whatever style you like: pictures, experiments, poems (rhyme of prose), or just plain English:

   a. Consider data from the following distribution: Every data point has 50% probability for being positive for each of the explanatory attributes. Any person who is positive for “salary over $100,000” is also positive for the target attribute, “republican,” and any person who is negative for “salary over $100,000” is also negative for the target attribute, “republican.” (See the function `generateDataPartA` in decisionTrees.py for clarification. You are not expected to run this code.)
   Could this tree have come from data produced with this distribution? Why or Why not?

   b. Consider a slightly different dataset from the following distribution: Every data point has 50% probability for being positive for each of the explanatory attributes. Additionally, any person who is positive for “salary over $100,000” is also positive for the target attribute, “republican,” with 95% probability and any person who negative for “salary more than $100,000” is also negative for the target attribute, “republican” with 95% probability. (See the function `generateDataPartB` in decisionTrees.py for clarification. You are not expected to run this code.)
   Could this tree have come from data produced with this distribution? Why or Why not?

   c. If you trained a decision tree on data from `generateDataPartA,` what effect would it have on the resulting tree if .5 were changed to .3?

   d. If you trained a decision tree on data from `generateDataPartB,` what effect would it have on the resulting tree if .05 were changed to .3?

Step 3. Implement Misclassification Impurity evaluation

If it’s not clear from the title, misclassification impurity is the percentage of errors you get when you use the tree your training data produced to classify the test data.

<u>For your write up:</u>

A. Train a tree using republican.trainingData1. Report the misclassification impurity on:

   a. republican.trainingData1
   b. republican.testingData1

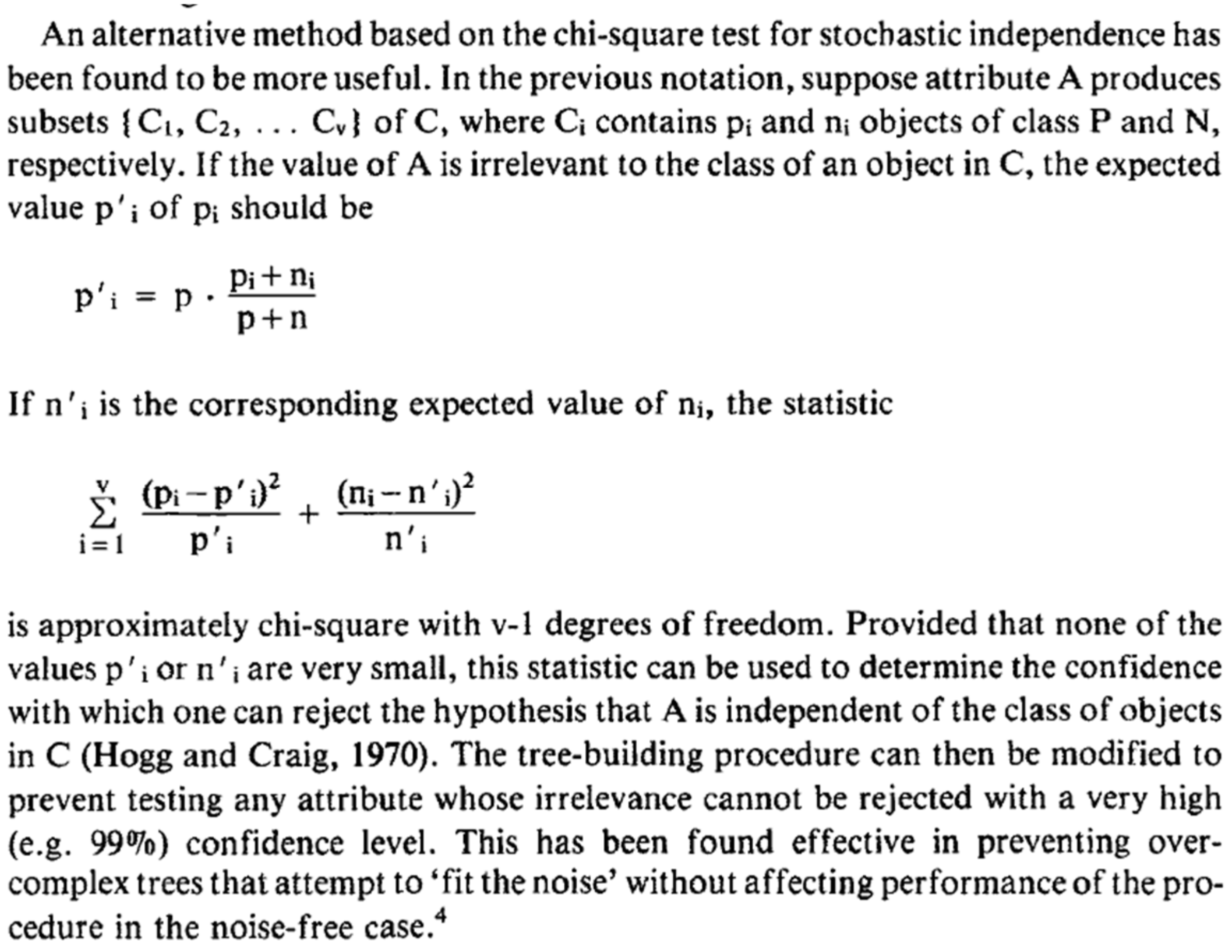
A. Train a tree using republicans.trainingData2. Report the misclassification impurity on:

   a. republican.trainingData2
   b. republican.testingData2

Step 4. Chi Squared Pruning

Chi Squared pruning is described in the original ID3 paper:
http://www.dmi.unict.it/~apulvirenti/agd/Qui86.pdf

Here is the relevant section of the paper:



An alternative method based on the chi-square test for stochastic independence has been found to be more useful. In the previous notation, suppose attribute A produces subsets $\{C_1, C_2, \ldots C_v\}$ of C, where C_i contains p_i and n_i objects of class P and N, respectively. If the value of A is irrelevant to the class of an object in C, the expected value p'_i of p_i should be

$$p'_i = p \cdot \frac{p_i + n_i}{p + n}$$

If n'_i is the corresponding expected value of n_i, the statistic

$$\sum_{i=1}^{v} \frac{(p_i - p'_i)^2}{p'_i} + \frac{(n_i - n'_i)^2}{n'_i}$$

is approximately chi-square with v-1 degrees of freedom. Provided that none of the values p'_i or n'_i are very small, this statistic can be used to determine the confidence with which one can reject the hypothesis that A is independent of the class of objects in C (Hogg and Craig, 1970). The tree-building procedure can then be modified to prevent testing any attribute whose irrelevance cannot be rejected with a very high (e.g. 99%) confidence level. This has been found effective in preventing over-complex trees that attempt to 'fit the noise' without affecting performance of the procedure in the noise-free case.[4]

Where p_i are the number of examples that are positive for the *target attribute*, and C_1 and C_2 are the set of examples that are positive and negative for the *selected attribute*.

Implement chi squared pruning in the ID3 algorithm.

You will make a decision tree to predict whether cars seen on craigslist in the cars.training dataset are acceptable. There are 1728 observations in the dataset. It has features such as “new paint job”, “high price”, “high maintainance,” etc. The attribute we are trying to predict is “acceptable.” All variables are binary.

This time, use the data set cars.training dataset to create the following three trees:

```
tree1 = id3(cars.trainingData,"acceptable", ["new paint job","many previous
owners","recent oil change","has vanity plate","very high price","high price","mid
price","low price","very high maintainance","high maintainance","mid
maintainance","low maintainance","2 doors","3 doors","4 doors","5+ doors","2
passengers","4 passengers","more passengers","small boot","med boot","large boot"])
```

```
tree2 = id3(cars.trainingData,"acceptable", ["very high price","high price","mid
price","low price","very high maintainance","high maintainance","mid
maintainance","low maintainance","2 doors","3 doors","4 doors","5+ doors","2
passengers","4 passengers","more passengers","small boot","med boot","large boot"])
```

```
tree3 = id3(cars.trainingData,"acceptable", ["high safety","med safety","low
safety","very high price","high price","mid price","low price","very high
maintainance","high maintainance","mid maintainance","low maintainance","2 doors","3
doors","4 doors","5+ doors","2 passengers","4 passengers","more passengers","small
boot","med boot","large boot"])
```

Make a plot of the misclassification impurity for:

cars.trainingData
cars.testingData

Using the following 10 chi squared values:

[0.0, 1.0, 2.0, 3.0, 4.0, 10.0, 20.0, 50.0, 100.0, 200.0]

For your write up

A. Include the 3 plots

B. Include images (or partial images, if the trees are big) of the three trees

C. List the depth of each tree (I recommend counting this by hand)

D. Which of these 30 trees (3 trees, 10 chi-squares) is best? Why? For the types of trees that are generally “good,” what trade-offs are there in choosing one? Provide images where it would help your argument.

E. Based on the performance of each tree, which attributes is/are the most powerful explanatory features. Which attributes is/are the least explanatory features. Give a reason.

F. Describe the signs of overfitting you see and speculate as to what explains it. (It’s fine if this answer is a bit redundant with parts of your previous answers.)

G. What does a chi-squared cutoff of 200 mean? How would you describe the decision tree it produces?