# CMSC 422 – Assignment 3: Learning Decision Trees – Spring 2020

Induction of decision trees has been a long-standing topic of interest in machine learning. The purpose of this assignment is to give you experience with basic tree induction methods.

### 1. Basic Decision Tree Induction

Suppose that a decision tree is desired that will classify a person's credit risk as high, medium or low based on the person's credit history, debt, collateral and income. The training data D is:

| CREDIT HISTORY | DEBT | COLLATERAL | INCOME | RISK |
|---|---|---|---|---|
| bad | high | none | low | high |
| unknown | high | none | medium | high |
| unknown | low | none | medium | moderate |
| unknown | low | none | low | high |
| unknown | low | none | high | low |
| unknown | low | adequate | high | low |
| bad | low | none | low | high |
| bad | low | adequate | high | moderate |
| good | low | none | high | low |
| good | high | adequate | high | low |
| good | high | none | low | high |
| good | high | none | medium | moderate |
| good | high | none | high | low |
| bad | high | none | medium | high |

In this table, RISK is the class to be predicted. Assume all possible attribute values are shown.

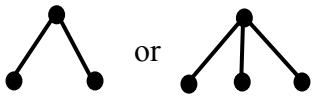
**a)** For this domain, what is the theoretical maximum of the entropy of *any* original set of data (like that in the table above) that you might receive?

**b)** What is the entropy H(D) of the complete set D of data given in the table above?

**c)** Considered as a possible root node of a decision tree being built by ID3, what is the expected entropy associated with the attribute INCOME?

**d)** Given that the information *gain* (not the entropy) of the other attributes is gain(credit history) = 0.27, gain(debt) = 0.06, and gain(collateral) = 0.21, which of the four possible attributes would ID3 select for the root node of the decision tree it is building?

**e)** Given your choice for the root node in (d), show the nodes ID3 would add immediately below each branch of the root node. In other words, you should have a diagram like one of those at the right, where each node and link has an appropriate adjacent label. It's not necessary to show any further calculations, but you must give a one sentence explanation for why ID3 would select each specific attribute or class label that it does for each non-root node.




**f)** Suppose that C4.5 is being used instead of ID3. Compute the gain-ratio for attribute INCOME considered as a possible root node.

## 2. Tree Induction Using an ID3-Like System

Marsland's ML textbook web site (https://homepages.ecs.vuw.ac.nz/~marslast/MLbook.html ) provides Python code (specifically *dtree.py*) that can be used to induce decision trees. You will be given a modified version of Marsland's *dtree.py* in the file *DT.py* that contains a number of changes (discussed in class) to use with this problem. Modify the code (specifically function *printTree()*) so that it saves its output directly in a file named *ResultsID3.txt* that is to be turned in, and so that it not only outputs the induced trees (indented format) as it does now, but also the total number of nodes in the tree (including leaves) and the number of leaves in the tree. Your results file should also contain the number of training examples used and the number of these training examples that were subsequently classified correctly by the induced tree. To keep life simple, i.e., to minimize code that needs to be written, *in this problem only*, the complete set of provided data is to be used for both training and testing (no need to divide it into training, validation and testing portions). Obviously, this is not acceptable in general. Further, the textbook's code is in Python 2; you can use it that way or convert it to Python 3 as you prefer.

You will be provided with a data set *carData.txt* where each of the 1728 examples specifies the values of six features of a car

buying: whether the purchase price is very high, high, medium or low;
maint: whether the maintenance costs are very high, high, medium or low;
doors: number of doors;
persons: number of people car can carry;
lugBoot: size of the luggage boot (boot = British English for trunk/compartment);
safety: estimate of car's safety

and the value of the car's acceptability (the output class). Write a script file *cars.py* that uses this car data set and *DT.py* to generate a decision tree that predicts whether a person will find a car to be unacceptable, acceptable or good. This tree and the other information described above should be saved in the *ResultsID3.txt* file. Separately answer the following questions.

**a**. Make a picture of just the top two levels of the decision tree that was output by your code. A hand-drawn sketch of the labeled nodes and labeled branches is fine; i.e., don't use the indented format that *printTree()* uses. The "top two levels" are the root node, the nodes directly under the root node, plus the nodes directly under those, along with the labeled branches.

**b**. According to the results here, which of the six input features, in isolation, provides the most information in terms of determining whether a car is acceptable, unacceptable, or good? How many nodes total are in the induced tree, and how many of these are leaf nodes?

**c**. How would this induced decision tree classify a car with the following properties?

buying: med; maint: med; doors: four; persons: four; lugBoot: med; and safety: high.

**d**. It is often said that induced decision trees provide simple representations of the data on which they are trained. Is that the case here for the cars decision tree? Why is this an important issue?

**e**. Learning decision trees can be viewed as one approach to learning a set of rules. Show how that is true in this case by *manually* generating a single example of an if-then rule (a production) from the cars decision tree you induced. How many rules total would one generate in producing a rule set equivalent to this induced tree?

**f**. Unlike with typical textbook examples, many real-world decision trees induced by ID3 do not provide a conceptually simple and readily understandable representation of the data on which ID3 was trained. What post-processing step is often used to generate a more understandable version of such complicated decision trees?

### 3. Tree Induction Using C4.5-like System

There are several existing machine learning toolboxes in addition to scikit-learn, many of which include modules for inducing decision trees using more contemporary methods than ID3. One of these is Weka's J48, a version of C4.5. To use Weka, go to http://www.cs.waikato.ac.nz/ml/weka/ and download it to your computer, following the documentation there for Windows, Mac OS X, or Linux as is appropriate. Select the latest stable version to use, and review the instructions. For example, with OSX on iMac's, double clicking on the file *weka.jar* (or on the Weka bird icon) will start Weka, perhaps after a small delay. Save the decision tree that you generate below using Weka in the file *ResultsJ48.txt*.

Use Weka's J48 module to induce a decision tree for the car data in *carData.arff*. The data in file *carData.arff* is the same as that in *carData.txt* used in Problem 2, but the file suffix has been changed (".arff") to conform to Weka's expectations, and required header information has been added to the beginning of the file. The header information defines the name of the data set, the input features (or attributes) and their values, and the output classes. Use the existing default settings and parameter values of J48 (e.g., 10-fold cross validation). Copy and paste the contents of the output window into the results file *ResultsJ48.txt*, thereby saving not only the induced tree but also the performance information (error rate, confusion matrix, etc.).

**a** . How does the decision tree for the car data produced by Weka differ from that that produced by ID3 in Problem 2 above? What is the total number of nodes in this tree, and how many of them are leaf nodes? Explain why these differences in ID3 and J48 results occurred.

**b.** How does Weka's decision tree classify the example car given in Problem 2c?

**c**. Looking at the results given in Weka's output, what is the reported error rate for the generated decision tree, and what is the most common specific error made by the decision tree on the data?

**d**. If additional future car examples are found that were not included in the training data used here, which of the two trees generated, the one using ID3 in Problem 2 or the one using J48 in this problem, would be most likely to generalize best to this new data? Why?

**e.** Explain how the original C4.5 (not J48) software pruned the rules that it generated. No need to do any rule pruning here, just explain how C4.5 did it.

### What should I turn in?

*The hardcopy and electronic submissions are due at different times*. The hard copy portion is due at the start of class Thurs. March 5; the electronic submission is due earlier at 11:30 pm Weds. March 4.

*Hardcopy*: Your answers to the questions in problems 1a-f, 2a-f, and 3a-e.
*Electronic submission*: For problems 2 and 3, turn in a single zip file that includes the result files *ResultsID3.txt* and *ResultsJ48.txt*, your modified version of *dtree.py*, and the script file *cars.py* that you wrote to do Problem 2. Be sure that you include any files you wrote that are needed to make your code run. As was done with the previous assignments, use the Computer Science Department project submission server that is located at https://submit.cs.umd.edu to submit this zip file.