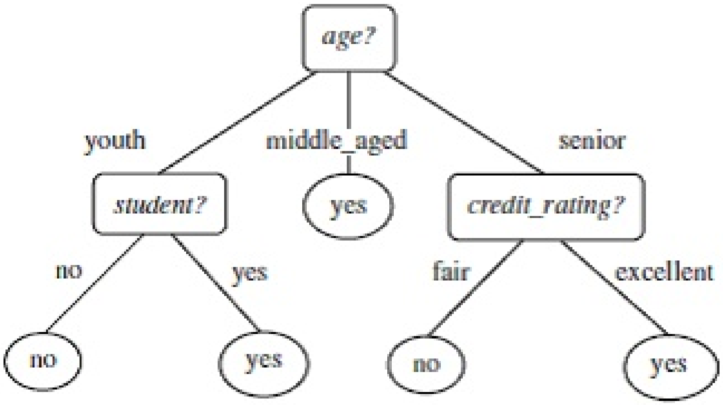
Decision Trees Tutorial
COMS3007

Benjamin Rosman, Devon Jarvis

Instructions: Use your notes and any resources you find online to answer the following questions. You need to submit a PDF of your answers to questions 1-4 on Moodle. This can be done in groups of *up to four*. Make sure all your names and student numbers appear on the document!

1. Consider the following decision tree on whether or not a customer should be sold an expensive phone:



   (a) What are the three *feature* variables we expect to see in each data point in the data set? What values can they each take on?

   (b) What is the *target/class* variable in the data? What values can it take on?

   (c) Classify the following customers according to the decision tree:

      i. A senior with a fair credit rating.

      ii. A middle aged student with an excellent credit rating.

      iii. A youth who isn't a student but has an excellent credit rating.

   (d) We can describe a decision tree as a rule for making a decision. This would be of the form: "Classify *yes* if (some variables have specific values) or (some other variables have specific values) or (some other variables have specific values) or ..."

      i. Write down the rule (in natural language, as above) that this tree describes.

      ii. Try and formulate this same rule in terms of formal logic, i.e. use $\vee$ in place of *or*, and $\wedge$ in place of *and*. We may expect a result such as

$$Class = (var1 = value \wedge var2 = value) \vee (var1 = value \wedge var3 = value) \vee (var1 = value)$$

2. When learning the tree from training data, we need some way of deciding where to put each decision in the tree. Why? What difference would this make?

3. A common approach is to consider the *information entropy* (uncertainty) of the distribution of the *class labels*. We would ideally like each leaf node in the tree to have no uncertainty, so every data point under that node has the same label (entropy = 0).

   Entropy $H(p)$ of a distribution p is computed as

$$H(p) \quad = \quad -\sum_{i=1}^{n} p_i \log_2(p_i)$$

   where p_i is the probability of class i. For example, if the two classes are Y and N, then $H(p) = -(p_Y \log_2(p_Y) + p_N \log_2(p_N))$.

   Now, compute the entropy $H(p)$ of the following distributions:

   (a) $p = \{0.5, 0.5\}$
   (b) $p = \{\frac{1}{3}, \frac{2}{3}\}$
   (c) $p = \{0.25, 0.75\}$
   (d) $p = \{0, 1\}$
   (e) $p = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$
   (f) $p = \{0.25, 0.25, 0.5\}$
   (g) The probability of the outcomes in $\{Y, N, N, N, Y, Y, N, Y\}$
   (h) The probability of the outcomes in $\{Y, Y, N, Y, Y, Y, Y, Y\}$

4. To construct a decision tree using the entropy, we want to iteratively add new nodes to a tree, starting with an empty tree. At each step, we want to add the node (a variable) that gives us the maximal reduction in entropy over the leaves below it. This is the node that gives us the largest *gain* in information.

   A dataset D has an entropy of $H(D)$. This is the entropy of the distribution of the target/class variable in D. We then want to split the dataset into several branches based on the values of a feature F. If f is a value that F can take, we denote the entropy of the subset of D where $F = f$ as D_f (i.e. this is the rows of D where $F = f$). For a feature F we define the *gain* as:

$$Gain(D, F) \quad = \quad H(D) - \frac{1}{|D|} \sum_{f \in valuesofF} |D_f| H(D_f).$$

   Note that we sum over every possible value f of F as we want the total entropy across all branches. $|X|$ is the number of elements in set X, and in the sum we weight the entropy of each branch by the number of data points in that branch. Why?

   Now, consider the data D of COMS3 performance in table 1 below, and answer the following questions.

   (a) What is $H(D)$? (Remember: the target here is the variable *passing?*)
   (b) What is $Gain(D, COMS2mark)$? (Here the target is still *passing?*, but you will need to consider the three branches where $F = A, B, C$, and compute their entropies individually)
   (c) What is $Gain(D, doinglabs?)$? (Here the target is still *passing?*, but you will need to consider the two branches where $F = Y, N$)
   (d) What is $Gain(D, doingtuts?)$? (Here the target is still *passing?*, but you will need to consider the two branches where $F = Y, N$)
   (e) Now, we add the (root) node the tree as the variable with the maximum gain. Which variable is this? How many branches originate at this node?

Table 1: COMS3 performance dataset

COMS2 mark	**doing labs?**	**doing tuts?**	**passing? (target)**
A	N	Y	Pass
C	Y	N	Fail
C	N	Y	Pass
B	Y	Y	Pass
B	N	N	Fail
C	Y	N	Pass
A	N	N	Fail
B	Y	N	Pass

(f) Then we repeat the process, treating each branch separately (i.e. the next variable on each branch may be different). Why?

5. Download the zip folder "Decision_Tree_Lab.zip" from Moodle. Inside is a Jupyter Notebook, "Decision_Trees.ipynb" and two text files "card_categories.txt" and "card_data.txt". The Jupyter Notebook can be opened on the lab machines by running "/usr/local/anaconda3/bin/jupyter notebook". Further instructions and context can be found in the Notebook.

   (a) Complete the Notebook by filling in all points marked by a "TODO" comment.

   (b) What do you notice about the architecture of the tree when the individual words are no longer used as a feature to learn the Decision Tree, compared to when the words are used as a feature?

   (c) What do you notice about the depth of the tree as the tasks get more complicated?

   (d) What do you notice about the Training and Test accuracies when noise is present in the data? How similar is the model's accuarcy for Training and Test data? Why do you think one accuracy is slightly higher than the other, and will this relation between the accuracies usually be like this?

   (e) Find another dataset online to test your algorithm. A good repo for datasets is the UCI Machine Learning repository (https://archive.ics.uci.edu/ml/index.php). As an example, you could use the Soybean Data Set: (https://archive.ics.uci.edu/ml/datasets/Soybean+%28Small%29).