CIS 572 Assignment 1

Steven Walton

April 11, 2019

Problem 1

Answer Exercise 3.1 from Chapter 3 of Mitchell's machine learning book.

Part a)

$$A \wedge \neg B$$

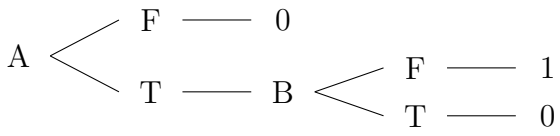


Figure 1: $A \wedge \neg B$

Part b)

$$A \vee (B \wedge C)$$

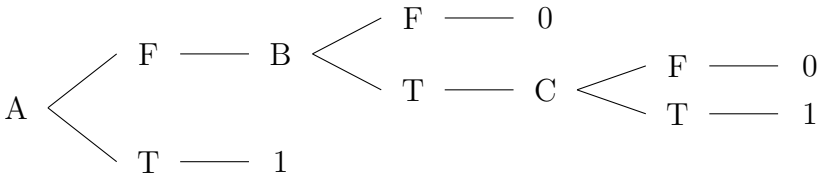


Figure 2: $A \vee (B \wedge C)$

Part c)

A XOR B

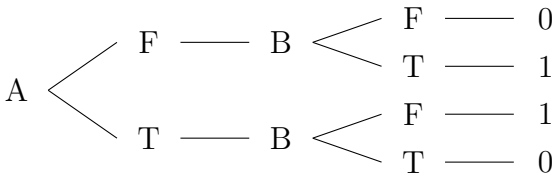


Figure 3: A XOR B

Part d)

$$(A \wedge B) \vee (C \wedge D)$$

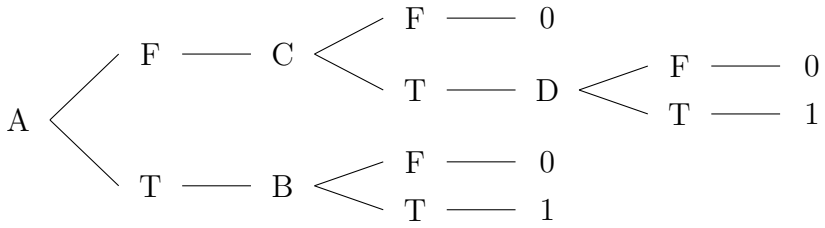


Figure 4: $(A \wedge B) \vee (C \wedge D)$

Problem 2

Consider the samples in the Play-tennis dataset from Table 3.2 in Mitchell's textbook. If you calculate the information-gain for all of the attributes of this set, you will observe that the attribute "Outlook" has the largest information- gain, which is equal to 0.246. Therefore, the attribute "Outlook" is the best heuristic choice for the root node.
(a) List the labels of the new tree branches below the root node.
(b) Which partition of the data will be assigned to each branch by ID3? Please list the sample IDs that will be assigned to each branch.
(c) Calculate the information gain for the remaining attributes in each branch, and determine which attribute will be chosen as the root of the sub-tree in each branch.

Part a)

Now that we know that Overcast has the highest information gain, of 0.246, we will use it as the root node. We will then create the next nodes and show their corresponding values for playing tennis or not, in the form of $\big[+yes, -no\big]$

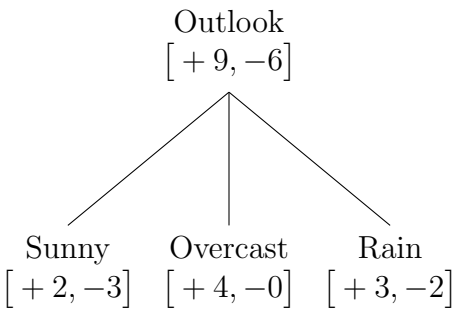


Figure 5: Labels of new tree branches below root

Part b)

We'll redraw the same tree but now include what days we partition.

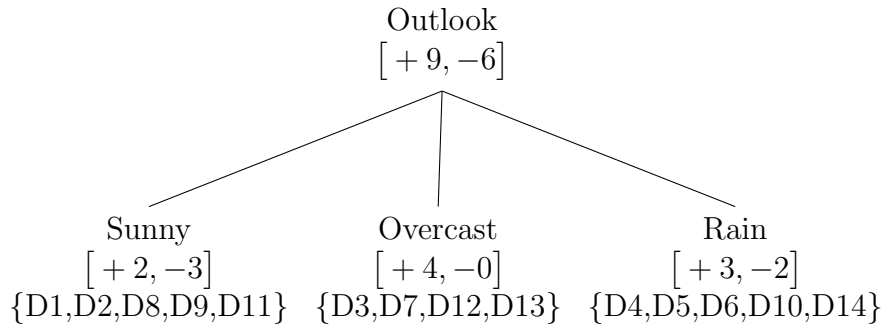


Figure 6: Label of tree with purity and associated days (D)

Note that Overcast is pure, so we are actually done with that decision process.

Part c)

To determine how we will partition the data we will look at the entropy of each node.
Given sunny, we have:
Humidity: High $[+0, -3]$, Normal $[+2, -0]$
Temperature: High $[+0, -2]$, Mild $[+1, -1]$, Cold $[+1, -0]$
Wind: Strong $[+1, -1]$, Weak $[+1, -2]$

$$\begin{aligned} S(Sunny, Humidity) &= S_{sunny} - S_{sunny,high} - S_{sunny,low} \\ &= S_{sunny} - \frac{3}{5}(-1\log_2 1) - \frac{2}{5}(-1\log_2 1) \\ &= 0.97 - \frac{3}{5}0 - \frac{2}{5}0 \\ &= 0.97 \end{aligned}$$

We'll shorten steps from now on. We note that any $\big[+a,-a\big] = 1$ and $\big[+a,0\big] = \big[0,-a\big] = 0$

$$\begin{aligned} S(Sunny, Temperature) &= S_{sunny} - S_{sunny,hot} - S_{sunny,mild} - S_{sunny,cold} \\ &= 0.97 - \frac{2}{5}0 - \frac{2}{5}1 - \frac{2}{5}0 \\ &= 0.57 \end{aligned}$$

$$\begin{aligned} S(Sunny, Wind) &= S_{sunny} - S_{sunny,strong} - S_{sunny,weak} \\ &= 0.97 - \frac{2}{5}1 - \frac{3}{5}\left(-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}\right) \\ &= 0.019 \end{aligned}$$

From here we can see that the best thing to split on is humidity and the worst is wind. We're actually done on *Sunny* because we always make a decision to play tennis or not given our current decision process. This results in the following

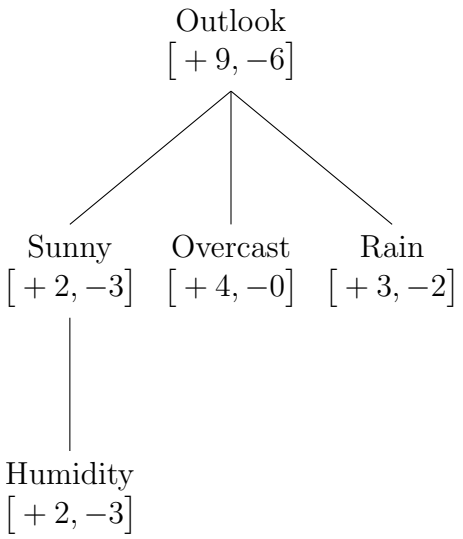


Figure 7: Partition of data

We now need to do the same thing with Rain. We can skip overcast because it is pure. Our only decisions left are "Wind" and "Temperature". We don't actually need to know S_{Rain} because we can just figure out which term is the smallest given Rain.

$$\begin{aligned} S(Rain, Wind) &= S_{Rain} - S_{Rain,strong} - S_{Rain,weak} \\ &= S_{Rain} - 0 - 0 \\ &= S_{Rain} \end{aligned}$$

This is because we have pure outcomes ($S_{Rain,Strong} = [+0, -2]$, $S_{Rain,Weak} = [+3, -0]$) We should actually know here that $Wind$ is the variable that we want, but we'll do $Temperature$ just to check that it isn't equally as good. We know it isn't because $S_{Rain,Hot} = [+0, -0]$, $S_{Rain,Mild} = [+2, -1]$, and $S_{Rain,Cold} = [+1, -1]$, but we'll do it anyway.

$$\begin{aligned} S(Rain, Temperature) &= S_{Rain} - S_{Rain,Hot} - S_{Rain,Mild} - S_{Rain,Cold} \\ &= S_{Rain} - 0 - \frac{3}{5}(-\frac{2}{3}\log_2(\frac{2}{3}) - \frac{1}{3}\log_2(\frac{1}{3})) - \frac{2}{5} \\ &= S_{Rain} - 0.551 - \frac{2}{5} \\ &= S_{Rain} - 0.151 \end{aligned}$$

It is trivial to see that $S_{Rain} > S_{Rain} - 0.151$, so we pick $Wind$. We can now complete our tree.

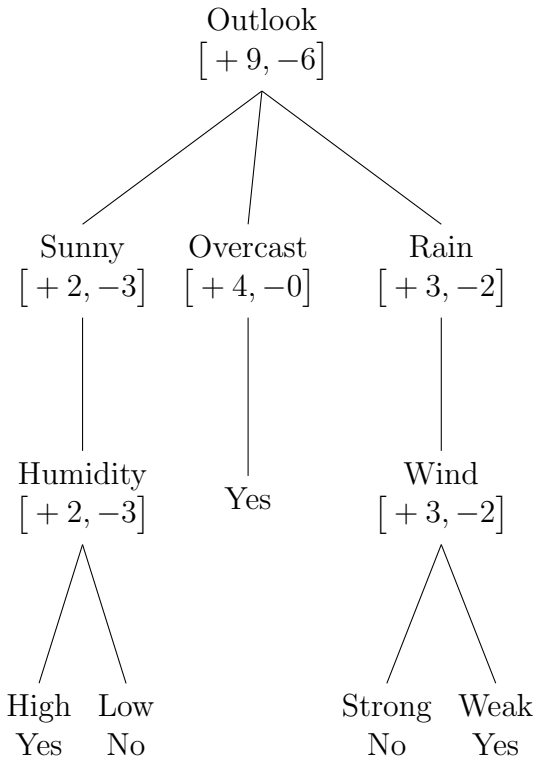


Figure 8: Full tree

We can throw out $Temperature$ because our current decision tree always results in an outcome of playing tennis or not.

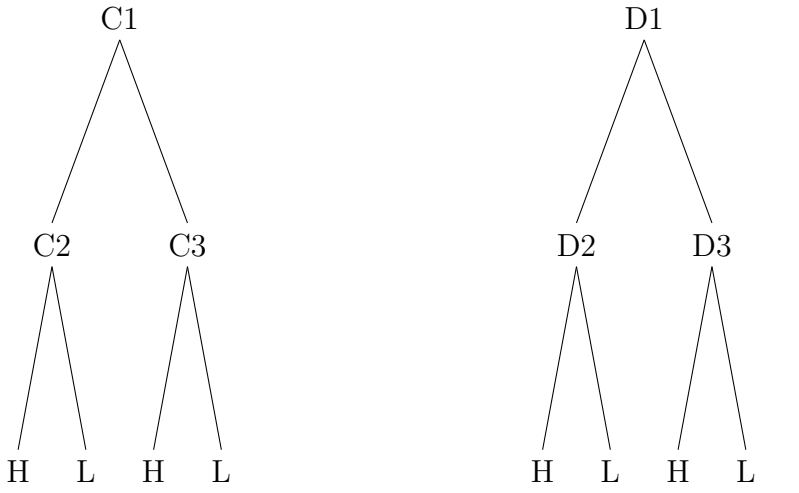
Problem 3

Suppose a bank makes loan decisions using two decision trees, one that uses attributes related to credit history and one that uses other demographic attributes. Each

decision tree separately classifies a loan application as "High Risk" or "Low Risk". The bank only offers a loan when both decision trees predict "Low Risk"
(a) Describe an algorithm for converting this pair of decision trees into a single decision tree that makes the same predictions (that is, it predicts non-risky only when both of the original decision trees would have predicted non-risky).
(b) Let n_1 and n_2 be the number of leaves in the first and second decision trees, respectively. Provide an upper bound on n, the number of leaves in the single equivalent decision tree, expressed as a function of n_1 and n_2.

Part 1)

A simple method to combine the two trees is to just attach one of the trees to all "Low Risk" leaves of the other tree. We only need to check the "Low Risk" leaves because a "High Risk" leaf is already ruled out. A simplified example is shown below in Figure 9 and Figure 10.



(a) Credit History (b) Demographic Attributes

Figure 9: Example trees of Credit History and Demographic Attributes

Part 2)

The best way to do this is to draw a bunch of graphs and test different. I won't draw there here because it takes a lot in LaTeX. But testing we can easily see that $n_1 n_2$ is a tight upper bound. I will draw our worst case tree to demonstrate, seen in Figure 11. Note that our best case would be all leaves being "High Risk" (which results in a bound of n_1, but that's boring). We can reason this is the worst because every instance of "Low Risk" generates an instance of Tree_2, which has n_2 leaves. Thus we have the tightest bounds.

Part fun observation)

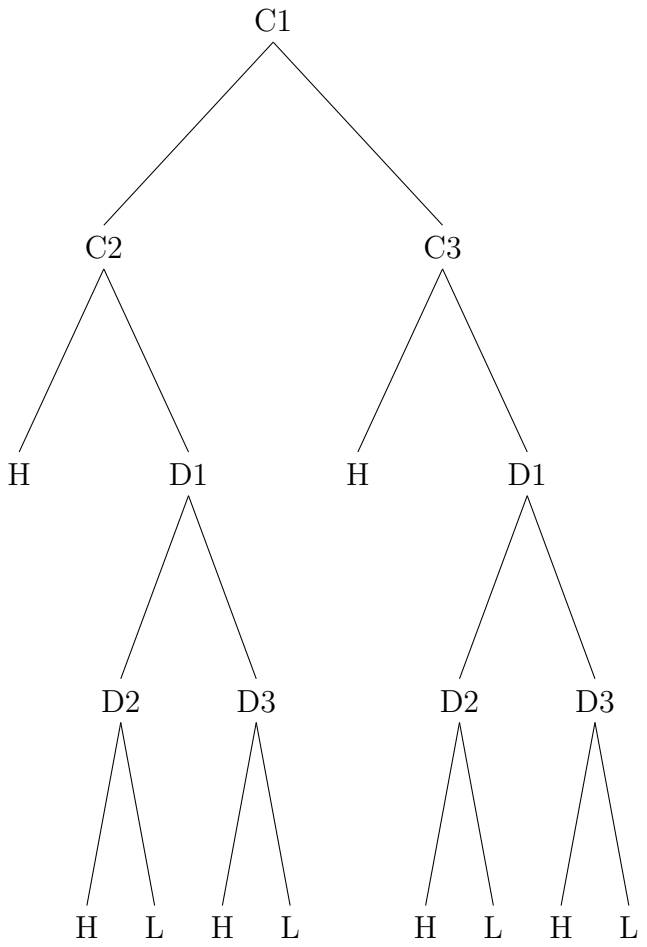


Figure 10: Combined Credit History and Demographic Attributes Tree

Playing around with trees I found a tighter bound if we have stricter definitions of a decision tree. If every node has either "High Risk" or another node then we can actually create a tighter bound of $\frac{1}{2}n_1 + \frac{1}{2}n_1n_2$. But with our less strict definition of a decision tree we can trivially show that this will not provide an upper bound to our worst case example.

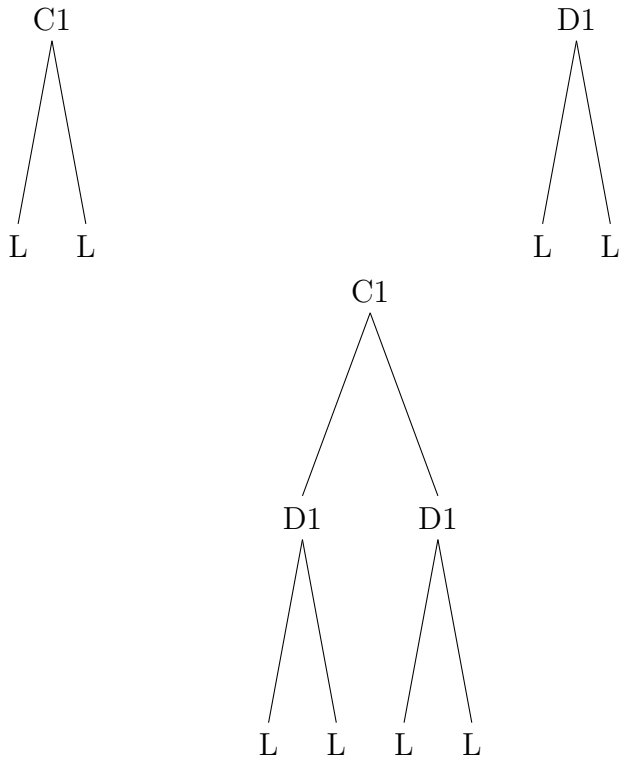


Figure 11: Worst case of tree additions