

# CIS 572 Assignment 1

Steven Walton

April 8, 2019

## Problem 1

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

### Part a)

$$A \wedge \neg B$$

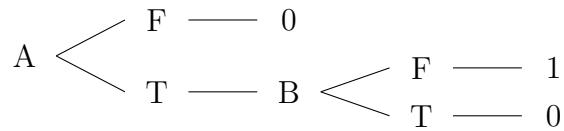


Figure 1:  $A \wedge \neg B$

### Part b)

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

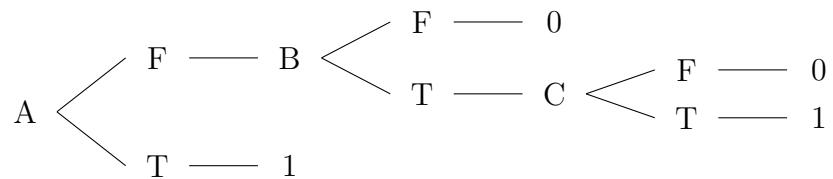


Figure 2:  $A \vee (B \wedge C)$

### Part c)

$A \text{ XOR } B$

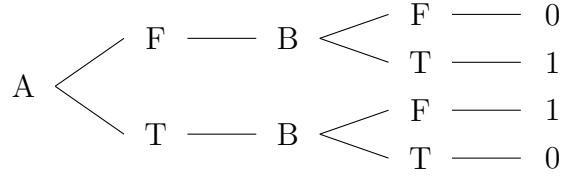


Figure 3:  $A \text{ XOR } B$

### Part d)

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

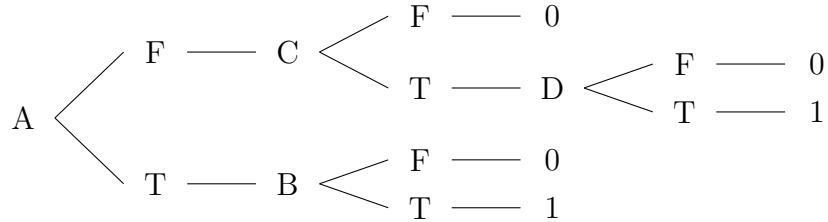


Figure 4:  $(A \wedge B) \vee (C \wedge D)$

## 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  $[+yes, -no]$

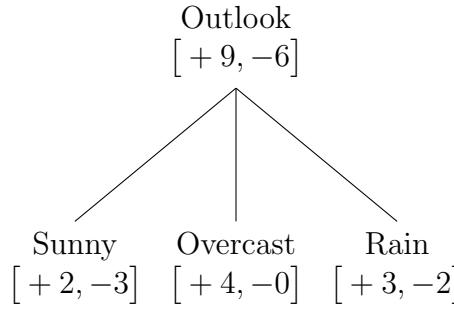


Figure 5: Labels of new tree branches below root

## Part b)

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  $[ +1, -2]$

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

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

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

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

From here we can see that the best thing to split on is humidity and the worst is wind. That gives us the following update to our tree.

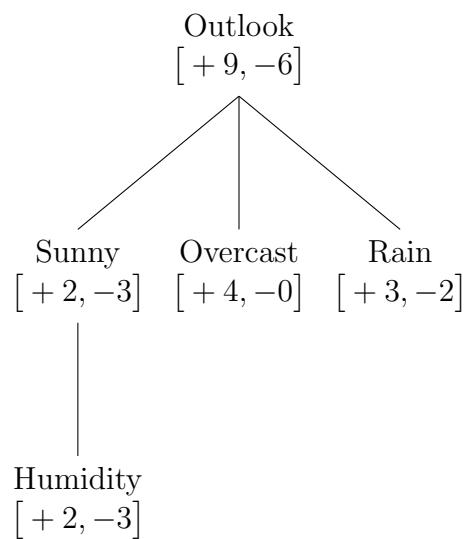


Figure 6: Partition of data

### Part c)

Continuing with this process we get the following tree

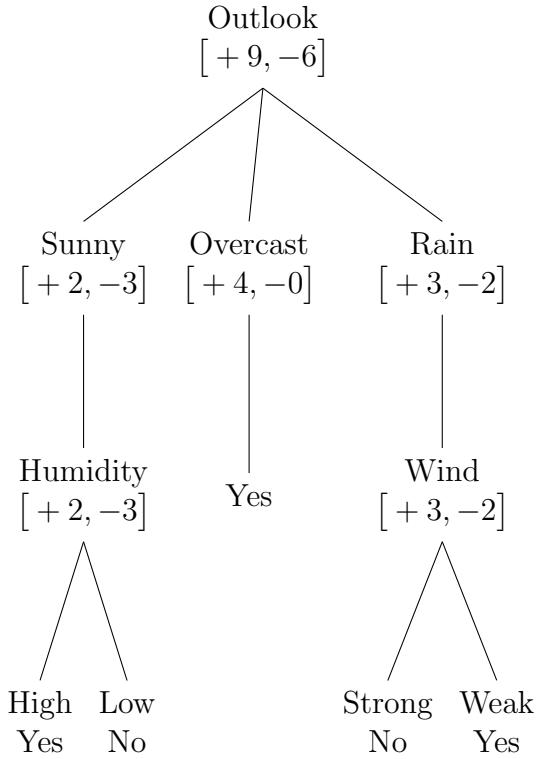


Figure 7: Full tree

## Problem 3

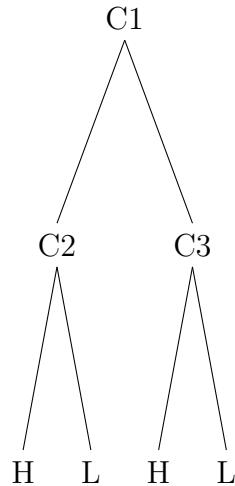
Suppose a bank makes loan decisions using two decision trees, one that uses attributes related to credit history and one that uses other demographic attributes. Each decision tree separately classifies a loan application as “High Risk” or “Low Risk”. The bank only offers a loan when both decision trees 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  $n_2$ .

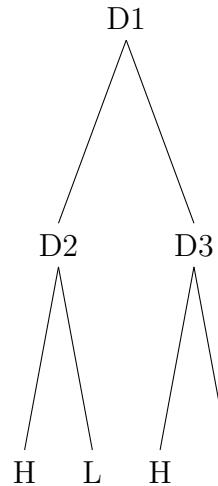
### Part 1)

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 5 and Figure 6.

### Part 2)



(a) Credit History



(b) Demographic Attributes

Figure 8: Example trees of Credit History and Demographic Attributes

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 L<sup>A</sup>T<sub>E</sub>X. 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 7. 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<sub>2</sub>, which has  $n_2$  leaves. Thus we have the tightest bounds.

### **Part fun observation)**

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  $\frac{1}{2}n_1 + \frac{1}{2}n_1 n_2$ . 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.

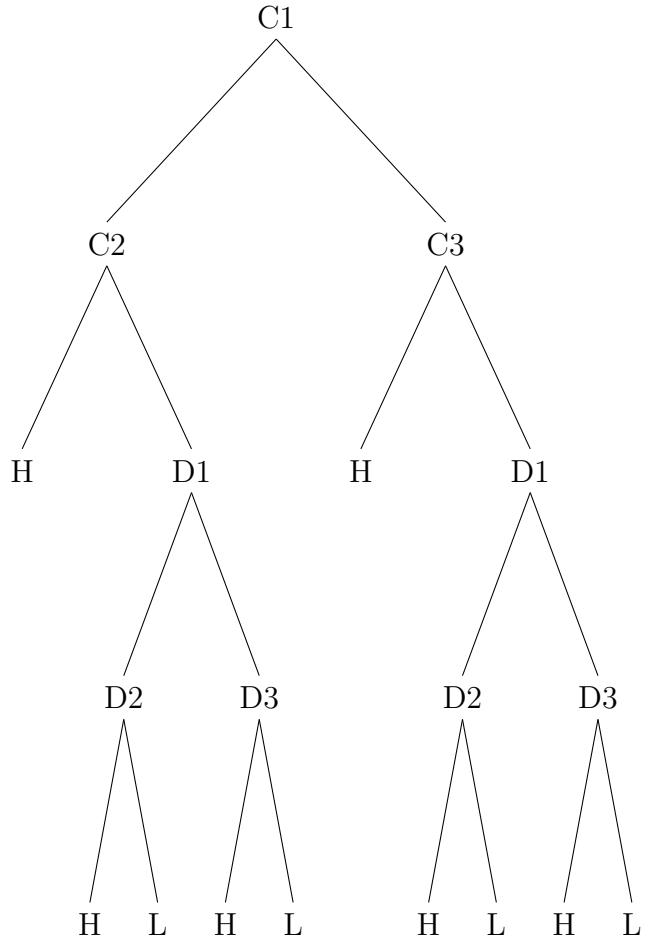


Figure 9: Combined Credit History and Demographic Attributes Tree

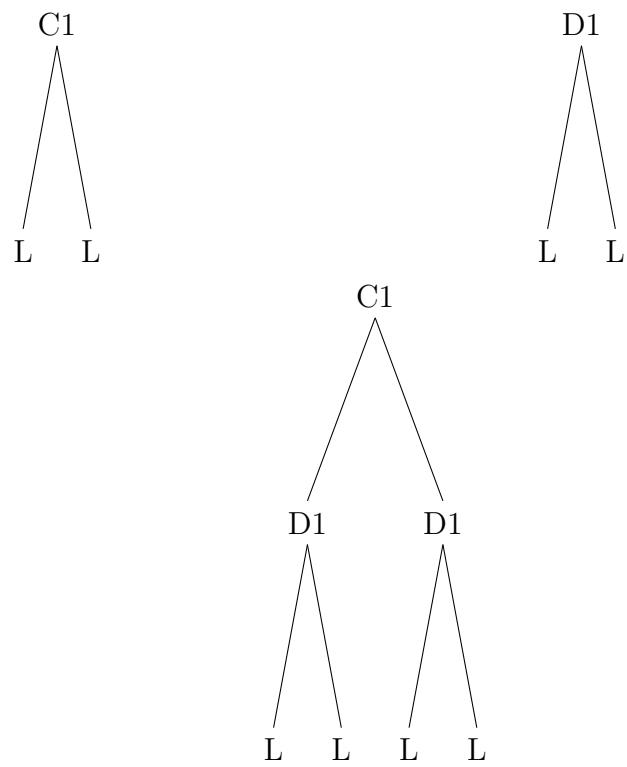


Figure 10: Worst case of tree additions