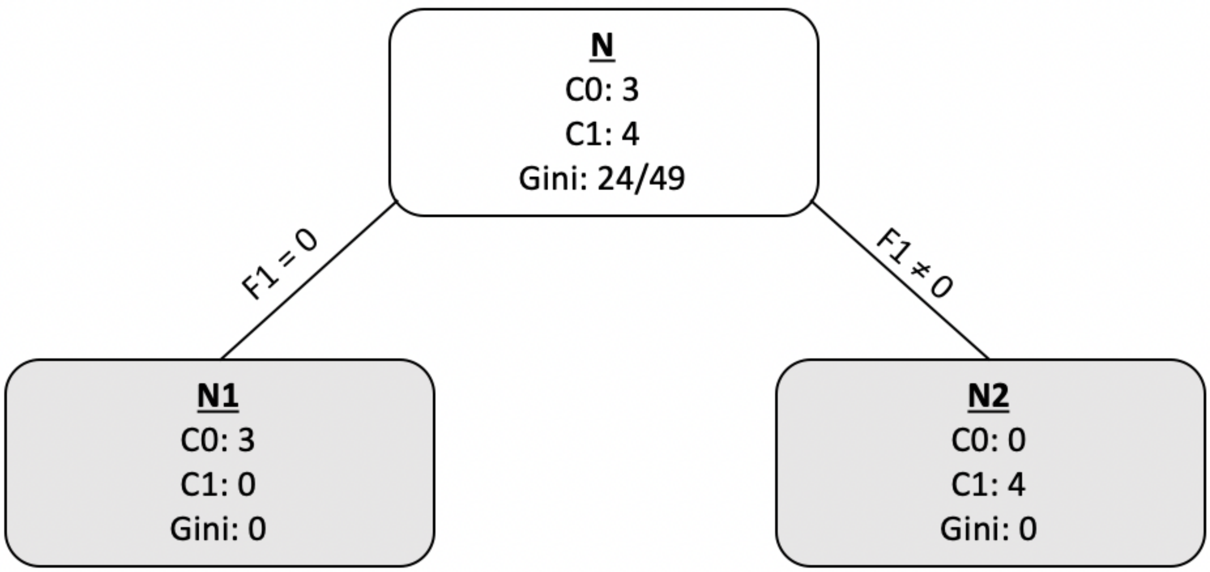
CS 74: Homework 2

Thomas Monfre

January 28, 2019

Problem 1

Using just equality tests, draw a decision tree containing a max of 3 levels (root level, children of root level, grandchildren of root level). Use Gini scores to generate the decision tree and clearly show the Gini values for each node as well as the Gini value of the splits generated.



Listed below are the Gini values for each node:

$$\begin{aligned} Gini(N) &= 1 - (\frac{3}{7})^2 - (\frac{4}{7})^2 \\ &= 1 - \frac{9}{49} - \frac{16}{49} \\ &= \frac{49}{49} - \frac{25}{49} \\ &= \frac{24}{49} \end{aligned} \tag{1}$$

$$
\begin{aligned}
Gini(N1) &= 1 - (\frac{3}{3})^2 - (\frac{0}{3})^2 \\
&= 1 - 1 - 0 \\
&= 0
\end{aligned} \tag{2}
$$

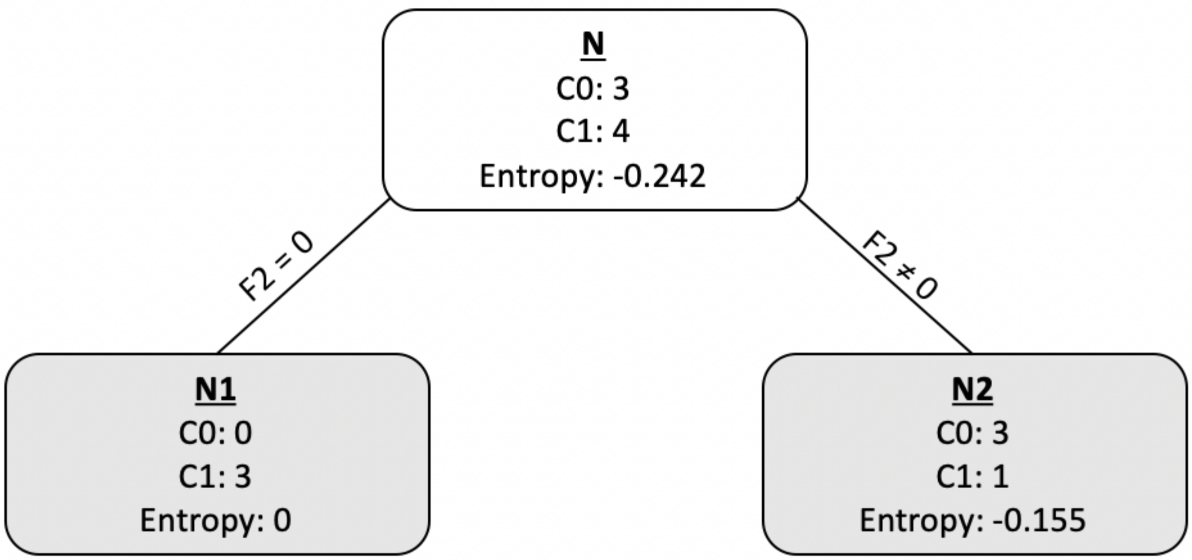
$$
\begin{aligned}
Gini(N2) &= 1 - (\frac{0}{4})^2 - (\frac{4}{4})^2 \\
&= 1 - 0 - 1 \\
&= 0
\end{aligned} \tag{3}
$$

Listed below is the Gini value for the split:

$$
\begin{aligned}
Gini(N, N1, N2) &= \left(\frac{|Sat(N1)|}{|Sat(N)|} \times Gini(N1) \right) + \left(\frac{|Sat(N2)|}{|Sat(N)|} \times Gini(N2) \right) \\
&= \frac{3}{7} \times 0 + \frac{4}{7} \times 0 \\
&= 0
\end{aligned} \tag{4}
$$

Problem 2

Suppose the definition of entropy of a node in a decision tree is changed from the definition given in the slides to that shown in the problem assignment. In other words, the summation in the definition given in the slides is replaced by a product. What decision tree would you get if you used this definition of entropy on the data given in Problem 1? Compare the result you got in Problem 1 with the result in Problem 2.



Listed below are the Entropy values for each node:

$$\begin{aligned} Entropy(N) &= -\left(\frac{3}{7} \times \lg\frac{3}{7} \times \frac{4}{7} \times \lg\frac{4}{7}\right) \\ &= -(\frac{3}{7} \times -1.222 \times \frac{4}{7} \times -0.807) \\ &= -(-0.524 \times -0.461) \\ &= -0.242 \end{aligned} \tag{5}$$

$$\begin{aligned} Entropy(N1) &= -\left(\frac{0}{3} \times \lg\frac{0}{3} \times \frac{3}{3} \times \lg\frac{3}{3}\right) \\ &= -(0 \times \lg 0 \times 1 \times \lg 1) \\ &= 0 \end{aligned} \tag{6}$$

$$
\begin{aligned}
Entropy(N2) &= -\left(\frac{3}{4} \times \lg\frac{3}{4} \times \frac{1}{4} \times \lg\frac{1}{4}\right) \\
&= -(0.75 \times \lg 0.75 \times 0.25 \times \lg 0.25) \\
&= -(0.75 \times -0.415 \times 0.25 \times -2) \\
&= -0.155
\end{aligned} \tag{7}
$$

Listed below is the Entropy value for the split:

$$
\begin{aligned}
Entropy(N, N1, N2) &= \left(\frac{|Sat(N1)|}{|Sat(N)|} \times Entropy(N1)\right) + \left(\frac{|Sat(N2)|}{|Sat(N)|} \times Entropy(N2)\right) \\
&= \frac{3}{7} \times 0 + \frac{4}{7} \times -0.155 \\
&= -0.088
\end{aligned} \tag{8}
$$

Since we have defined a new version of entropy, the decision tree has changed from splitting on Feature 1 to splitting on Feature 2.

In Problem 1, we computed the Gini values for each possible split (i.e. either Feature 1 or Feature 2). Using these calculations, we found that splitting on Feature 1 created a Gini value $Gini(N, N1, N2) = 0$ and splitting on Feature 2 created a Gini value $Gini(N, N1, N2) = 0.214$. Since we seek to find the minimal Gini value, we split on Feature 1. This created an optimal split since we achieved complete homogeneity at the leaves.

In Problem 2, however, we computed the modified Entropy values for each possible split. Using these calculations, we found that splitting on Feature 1 created a modified Entropy value $Entropy(N, N1, N2) = 0$ and splitting on Feature 2 created a modified Entropy value of $Entropy(N, N1, N2) = -0.088$. Since we seek to find the minimal modified Entropy value, we split on Feature 2. This created a less optimal split since we do not have complete homogeneity, but it is the split we must take since we are constructing the tree by minimizing Entropy.

We then must consider further splits in the tree. Since $N1$ is completely homogenous and each value for Feature 1 is identical, we leave it alone since we cannot get a lower Entropy. Since $N2$ has values for Feature 1 that are different and separable, however, we must determine if we should split again or leave it as is. Intuition would tell us to split again on Feature 1 since we can achieve complete homogeneity in the subsequent children $N3$ and $N4$. If we were to do this, we would have a modified Entropy value of $Entropy(N2, N3, N4) = 0$ since each child of $N2$ is completely homogenous. If we were to leave $N2$ alone, we would have the same modified Entropy value of $Entropy(N2) = -0.155$ as before. Since $-0.155 < 0$, we choose to leave $N2$ as is. This gives us the tree shown in the diagram above.

As a whole, the decision tree in Problem 2 is different from that of Problem 1 because the new metric we defined for an optimal split (the modified Entropy) creates negative values when we do not have a homogenous split and 0 when we do. In this way, an optimal split is defined as one that does not produce complete homogeneity, thus creating a different tree.

Problem 3 (i)

How would you define support, confidence, and lift in this case?

Support $= P(A_1...A_n \wedge B_1) + P(A_1...A_n \wedge B_2) - P(A_1...A_n \wedge B_1 \wedge B_2)$

Confidence $= P(B_1|A_1...A_n) + P(B_2|A_1...A_n) - P(B_1 \wedge B_2|A_1...A_n)$

Lift $= \frac{P(B_1|A_1...A_n)+P(B_2|A_1...A_n)-P(B_1 \wedge B_2|A_1...A_n)}{P(B_1)+P(B_2)-P(B_1 \wedge B_2)}$

Problem 3 (ii)

Can you suggest an adaptation of the Apriori algorithm to extract such rules from a body of data similar to those in the slides for association rules?

The traditional Apriori algorithm is split into two phases. The first finds all frequent itemsets that meet the support threshold. This guarantees that when testing rules against the confidence threshold, we are only considering itemsets that are serious contenders to generate a valid rule. The second phase takes the frequent itemsets that were generated and tests all possible rules of the correct form that itemset can generate. Each tested rule that passes the confidence threshold is then declared to be a valid rule.

In adapting the Apriori algorithm, let us first observe that the nature of a frequent itemset has changed since we've redefined the structure of our desired rules. Specifically, for a rule of the form $(A_1, A_2, ..., A_n) \implies B_1 \vee B_2$, in order for a rule to pass the support and confidence thresholds, $(A_1, A_2, ...A_n)$ must be frequent, but only one of B_1 or B_2 must be frequent. Both B_1 and B_2 could be frequent, but in order to generate a logically valid rule, only one ***must*** be frequent. For example, if each of $(A_1, A_2, ..., A_n)$ are frequent, B_1 is frequent, and B_2 is not, the rule $(A_1, A_2, ..., A_n) \implies B_1 \vee B_2$ would be logically valid and would meet the support threshold (assuming $(A_1, A_2, ..., A_n, B_1)$ meet the threshold) since our redefined support sums up the individual probabilities and subtracts their joint probability.

Observing this, let us then define our new phase one of the modified Apriori algorithm. We will first find all items that are themselves frequent (defined as meeting the support threshold). From those items, we will then find all pairs of items that are frequent. When finding just these frequent pairs, we will declare a pair frequent if the probability of each value in the pair occuring in the set is greater than the support threshold.

In more specific terms, let us first generate set L_1 that contains only elements from the universal set U that meet the support threshold. Specifically, for each element $x \in U$, $P(x) > s \implies x \in L_1$. Once we've constructed L_1, we then repeat the same process to generate L_2. For each element $x \in L_1$, we take some other element $y \in L_1$ and test (x, y) against s: $P((x, y)) > s \implies (x, y) \in L_2$. We now have L_2 which is a set of pairs of elements in which the probability of each item in the pair occuring together is greater than the support threshold.

Now, we handle the modified form of the desired rule. We will first take all pairs in L_2 and add a third element from U. Once we've added the third element, we will test the set against the support threshold s using our modified support metric defined above. Since this new metric defines a specific choice of B_1 and B_2, when determining whether or not an itemset I meets s, let us try all possible choices of B_1 and B_2 and consider I as passing the support threshold if at least one choice of B_1 and B_2 generates a support greater than s. Adding elements from the universal set (which includes elements that might not individually be frequent), allows us to observe the fact shown above that a set of items can pass the support threshold for a chosen B_1 and B_2 if one of B_1 or B_2 is infrequent and the other is frequent as well as when both B_1 and B_2 are frequent.

In more specific terms, for each set $X \in L_2$, let us choose some element $y \in U$. Since we are considering all elements in U, we therefore will try combinations that are entirely frequent as well as combinations that contain all but one frequent element. Then, let us take the union of X and y to produce a set Z. If Z is of size 3, then we test it against the support threshold. As discussed above, we will try all choices of B_1 and B_2 in Z (where the order of B_1 and B_2 does not matter). For each choice, if Z passes the support threshold defined above, we consider Z a member of L_3. We repeat this process for each set in L_2 and each element in U. When we are finished, we will have constructed the set L_3 that contains all sets that are contenders for producing a valid rule.

From here, we perform a process similar to that of the traditional Apriori algorithm to find sets L_4 and so on. For each set $X \in L_3$ we take the union of X and some other set $Y \in L_3$. If $Z = X \cup Y$ is of size 4, we then perform the same frequency check as before. We try each possible combination of B_1 and B_2 and promote Z to L_4 if at least one combination of B_1 and B_2 meets the support threshold. We repeat this process until our next produced set L_j is of size 0.

After finishing this process, we have then found all itemsets that pass the frequency threshold, keeping in mind the fact that not all elements in a set must be frequent in order for them to be contenders for producing a valid rule. Therefore, sets L_3 through L_{j-1} contain itemsets in which all or all but one element in the set is itself frequent. This considers all possible itemsets that meet the support threshold. By testing combinations of elements in L_3 with all members of U, we've captured all itemsets that are contenders for generating a valid rule. Since we build from L_3 and from then only take the union of sets previously determined frequent, we prune the collection of frequent itemsets.

Now that phase one is complete, phase two may begin. Phase two of the modified Apriori algorithm is similar to that of the traditional. For each frequent itemset X, we try all possible permutations of X and test it against the confidence threshold. Specifically, we try all rules of the desired form. Therefore, similar to above when testing against the support threshold, for each frequent itemset, we try all possible combinations of B_1 and B_2 (where order does not matter). For each choice of B_1 and B_2, if the set generated meets the confidence threshold, we declare it a rule. We use the equation for confidence defined above in the previous problem.

Therefore, in testing all possible combinations of B_1 and B_2, we've considered rules in which both B_1 and B_2 are frequent as well as rules in which only one of B_1 and B_2 are frequent. Thus, this modified Apriori algorithm extracts all possible rules of the desired form from the body of data.