Math 656 - HW3

Jeff Gould

9/15/2020

Exercise 3

a) What is the entropy of this collection of training examples with respect to the class attribute?

```
Table3.6 <- data.frame(
  Instance = seq(1,9,1),
  a1 = c(T, T, T, F, F, F, F, T, F),
  a2 = c(T, T, F, F, T, T, F, F, T),
  a3 = c(1, 6, 5, 4, 7, 3, 8, 7, 5),
  TargetClass = c(1,1,0,1,0,0,0,1,0)
)
```

Entropy is defined as $H(N) = -\sum P(i|N)\log_2(P(i|N))$. We have two classes, + (denoted with 1's) and -, denoted with 0's.

There are 4 instances of +, and 5 instances of -, and $N = 9$. So the entropy is given by:

$H(N) = -\sum_{i=1}^{9} P(i|9)\log_2(P(i|9)) = -P(+|N)\log_2(P(+|N)) - P(-|N)\log_2(P(-|N)) = -\frac{4}{9}\log_2(\frac{4}{9}) - -\frac{5}{9}\log_2(\frac{5}{9}) = 0.9910761$

b) What are the information gains of a_1 and a_2 relative to these training examples?

Information gain at a node is equal to the entropy of the class attribute minue the entropy at the node, ie at node a_1 the information gain is $H(N) - H(a_1)$

$H(a_1) = P(a_1 = F)H(a_1 = F) + P(a_1 = T)H(a_1 = T)$

$H(a_1 = F) = -P(-|a_1 = F)\log_2(P(-|a_1 = F)) - P(+|a_1 = F)\log_2(P(+|a_1 = F))$

$H(a_1 = T) = -P(-|a_1 = T)\log_2(P(-|a_1 = T) - P(+|a_1 = T)\log_2(P(+|a_1 = T))$

```
ftable(Table3.6[,c("a1", "TargetClass")])
```

```
##       TargetClass 0 1
## a1
## FALSE             4 1
## TRUE              1 3
```

$H(a_1 = F) = -\frac{4}{5}\log_2\left(\frac{4}{5}\right) - \frac{1}{5}\log_2\left(\frac{1}{5}\right)$

$P(a_1 = F) = 5/9$

$H(a_1 = T) = -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)$

$P(a_1 = T) = 4/9$

$H(a_1) = -\frac{5}{9}\left[\frac{4}{5}\log_2\left(\frac{4}{5}\right) + \frac{1}{5}\log_2\left(\frac{1}{5}\right)\right] - \frac{4}{9}\left[\frac{3}{4}\log_2\left(\frac{3}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right] = 0.7616392$

Subtract this from original entropy to get information gain:

```
a1_entropy <- -5/9 * (4/5*log2(4/5) + 1/5*log2(1/5)) + -4/9 * (3/4*log2(3/4) + 1/4*log2(1/4))
entropy <- -4/9 * log2(4/9) - 5/9 * log2(5/9)

entropy - a1_entropy
```

```
## [1] 0.2294368
```

Follow the same process for a_2:

$H(a_2) = P(a_2 = F)H(a_2 = F) + P(a_2 = T)H(a_2 = T)$

$H(a_2 = F) = -P(-|a_2 = F)\log_2(P(-|a_2 = F)) - P(+|a_2 = F)\log_2(P(+|a_2 = F))$

$H(a_2 = T) = -P(-|a_2 = T)\log_2(P(-|a_2 = T) - P(+|a_2 = T)\log_2(P(+|a_2 = T))$

```
ftable(Table3.6[,c("a2", "TargetClass")])
```

```
##       TargetClass 0 1
## a2
## FALSE             2 2
## TRUE              3 2
```

$H(a_2 = F) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)$

$P(a_2 = F) = 4/9$

$H(a_2 = T) = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)$

$P(a_2 = T) = 5/9$

$H(a_2) = -\frac{4}{9}\left[\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right] - \frac{5}{9}\left[\frac{3}{5}\log_2\left(\frac{3}{5}\right) + \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right] = 0.9838614$

Subtract from entropy to get an information gain of:

```
a2_entropy <- -4/9 * (1/2*log2(1/2) + 1/2*log2(1/2)) + -5/9 * (3/5*log2(3/5) + 2/5*log2(2/5))
entropy - a2_entropy
```

```
## [1] 0.007214618
```

So we find much greater information gain from a_1 than a_2, which suggests splitting and classifying the data on a_1 is better.

e) What is the best split, between a_1 and a_2, according to the misclassification error rate?

Misclassification error rate is simply the number of incorrect classifications divided by $N = 9$

Revisiting the table for a_1:

```
ftable(Table3.6[,c("a1", "TargetClass")])
```

```
##       TargetClass 0 1
## a1
## FALSE             4 1
## TRUE              1 3
```

If our classification rule is - if $a_1 = F$ and + if $a_1 = T$, , then we would incorrectly classify one item as - because $a_1 = F$, and incorrectly classify one item as + with $a_1 = T$. This leads to a misclassification rate of 2/9

Table for a_2:

```
ftable(Table3.6[,c("a2", "TargetClass")])
```

```
##       TargetClass 0 1
## a2
## FALSE             2 2
## TRUE              3 2
```

For classifying on $a_2 = F$, it is split 2:2 on whether or not to classify as a + or -. Either way, we have two misclassifications. Classifying for $a_2 = T$, we will classify as a -, with accuracy 3/5, which gives us 2 misclassifications. The total misclassification rate on $a_2 = 4/9$

Thus, using misclassification rate, the best split is again on a_1

f) What is the best split, between a_1 and a_2, according to the Gini index?

$Gini(N) = 1 - \sum[p(i|N)]^2$

For a_1:

$G(a_1 = F) = 1 - P(-|a_1 = F)^2 - P(+|a_1 = F)^2 = 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2$

$G(a_1 = T) = 1 - P(-|a_1 = T)^2 - P(+|a_1 = T)^2 = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2$

$P(a_1 = F) = 5/9,\ P(a_1 = T) = 4/9$

$G(a_1) = \frac{5}{9}\left[1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2\right] + \frac{4}{9}\left[1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2\right] = 0.3444444$

For a_2:

$G(a_2 = F) = 1 - P(-|a_2 = F)^2 - P(+|a_2 = F)^2 = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2$

$G(a_2 = T) = 1 - P(-|a_2 = T)^2 - P(+|a_2 = T)^2 = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2$

$P(a_2 = F) = 5/9,\ P(a_2 = T) = 4/9$

$G(a_1) = \frac{4}{9}\left[1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2\right] + \frac{5}{9}\left[1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2\right] = 0.4888889$

Since a smaller Gini is better, we again split the data based on a_2

7)

```
Table7 <- data.frame(
  X = c(0,0,0,0,1,1,1,1),
  Y = c(0,0,1,1,0,0,1,1),
  Z = c(0,1,0,1,0,1,0,1),
```

```
  C1 = c(5,0,10,45,10,25,5,0),
  C2 = c(40,15,5,0,5,0,20,15) )

Table7_long <- data.frame(
  X = c(rep(0,120), rep(1, 80)),
  Y = c(rep(0,60), rep(1,60), rep(0, 40), rep(1, 40)),
  Z = c(rep(0, 45), rep(1, 15), rep(0,15), rep(1, 45), rep(0, 15), rep(1, 25), rep(0, 25), rep(1, 15)),
  Class = c(rep("C1", 5), rep("C2", 40), rep("C2", 15), rep("C1", 10), rep("C2", 5), rep("C1", 45),
            rep("C1", 10), rep("C2", 5), rep("C1", 25), rep("C1", 5), rep("C2", 20), rep("C2", 15))
)
```

a) Compute a two-level decision tree using the greedy approach described in this chapter. Use the classification error rate as the criterion for splitting. What is the overall error rate of the induced tree?

The greedy approach facilitates that at each step we select the node that leads to the lowest mis-classification error

For attribute X:

```
table <- Table7 %>%
  group_by(X) %>%
  summarise(C1 = sum(C1),
            C2 = sum(C2)) %>%
  mutate(
    misclassified = pmin(C1, C2),
    MisclassificationRate = pmin(C1, C2) / (C1+C2))

kable(table, escape = FALSE, format = "latex") %>%
  kable_minimal(full_width = F)
```

X	C1	C2	misclassified	MisclassificationRate
0	60	60	60	0.5
1	40	40	40	0.5

For X we have a mis-classification rate of $\frac{100}{200} = 0.5$

Y:

```
table <- Table7 %>%
  group_by(Y) %>%
  summarise(C1 = sum(C1),
            C2 = sum(C2)) %>%
  mutate(
    misclassified = pmin(C1, C2),
    MisclassificationRate = pmin(C1, C2) / (C1+C2))

kable(table, escape = FALSE, format = "latex") %>%
  kable_minimal(full_width = F)
```

Y	C1	C2	misclassified	MisclassificationRate
0	40	60	40	0.4
1	60	40	40	0.4

mis-classification rate of $\frac{80}{200} = 0.4$

Z:

```
table <- Table7 %>%
  group_by(Z) %>%
  summarise(C1 = sum(C1),
            C2 = sum(C2)) %>%
  mutate(
    misclassified = pmin(C1, C2),
    MisclassificationRate = pmin(C1, C2) / (C1+C2))

kable(table, escape = FALSE, format = "latex") %>%
  kable_minimal(full_width = F)
```

Z	C1	C2	misclassified	MisclassificationRate
0	30	70	30	0.3
1	70	30	30	0.3

mis-classification rate of $\frac{60}{200} = 0.3$

So the first node will consist of classifiyng based on node Z

Node 2:

```
table <- Table7 %>%
  group_by(Z, X) %>%
  summarise(C1 = sum(C1),
            C2 = sum(C2)) %>%
  mutate(
    misclassified = pmin(C1, C2),
    MisclassificationRate = pmin(C1, C2) / (C1+C2))

kable(table, escape = FALSE, format = "latex") %>%
  kable_minimal(full_width = F)
```

Z	X	C1	C2	misclassified	MisclassificationRate
0	0	15	45	15	0.250
0	1	15	25	15	0.375
1	0	45	15	15	0.250
1	1	25	15	15	0.375

```
table <- Table7 %>%
  group_by(Z, Y) %>%
  summarise(C1 = sum(C1),
            C2 = sum(C2)) %>%
  mutate(
    misclassified = pmin(C1, C2),
    MisclassificationRate = pmin(C1, C2) / (C1+C2))

kable(table, escape = FALSE, format = "latex") %>%
  kable_minimal(full_width = F)
```

Z	Y	C1	C2	misclassified	MisclassificationRate
0	0	15	45	15	0.250
0	1	15	25	15	0.375
1	0	25	15	15	0.375
1	1	45	15	15	0.250

We discover that we gain no additional information at the second nodes with either X or Y.

```
# tree <- rpart(Class ~ Z + X, data = Table7_long, method = "class")
# rpart.plot(tree)
Table7_long <- Table7_long %>%
  mutate(X = as.factor(X),
         Y = as.factor(Y),
         Z = as.factor(Z))

sZ <- partysplit(which(names(Table7_long) == "Z"), index = 1:2)
sY <- partysplit(which(names(Table7_long) == "Y"), index = 1:2)

test <- partynode(id = 1L, split = sZ, kids = list(
  partynode(id = 2L, split = sY, kids = list(
    partynode(3L, info = "C2"),
    partynode(4L, info = "C1"))),
  partynode(5L, split = sY, kids = list(
    partynode(6L, info = "C1"),
    partynode(7L, info = "C2")
  ))
  ))
py <- party(test, Table7_long)

ggparty(py) +
  geom_edge() +
  geom_edge_label() +
  geom_node_label(aes(label = splitvar), ids = "inner") +
  geom_node_label(aes(label = info), ids = "terminal")
```

b) Repeat part (a) using X as the first splitting attribute and then choose the best remaining attribute for splitting at each of the two successor nodes. What is the error rate of the inudced tree?

```
table <- Table7 %>%
  group_by(X, Y) %>%
  summarise(C1 = sum(C1),
            C2 = sum(C2)) %>%
  mutate(
    misclassified = pmin(C1, C2),
    MisclassificationRate = pmin(C1, C2) / (C1+C2))

kable(table) %>%
  kable_minimal(full_width = F)
```

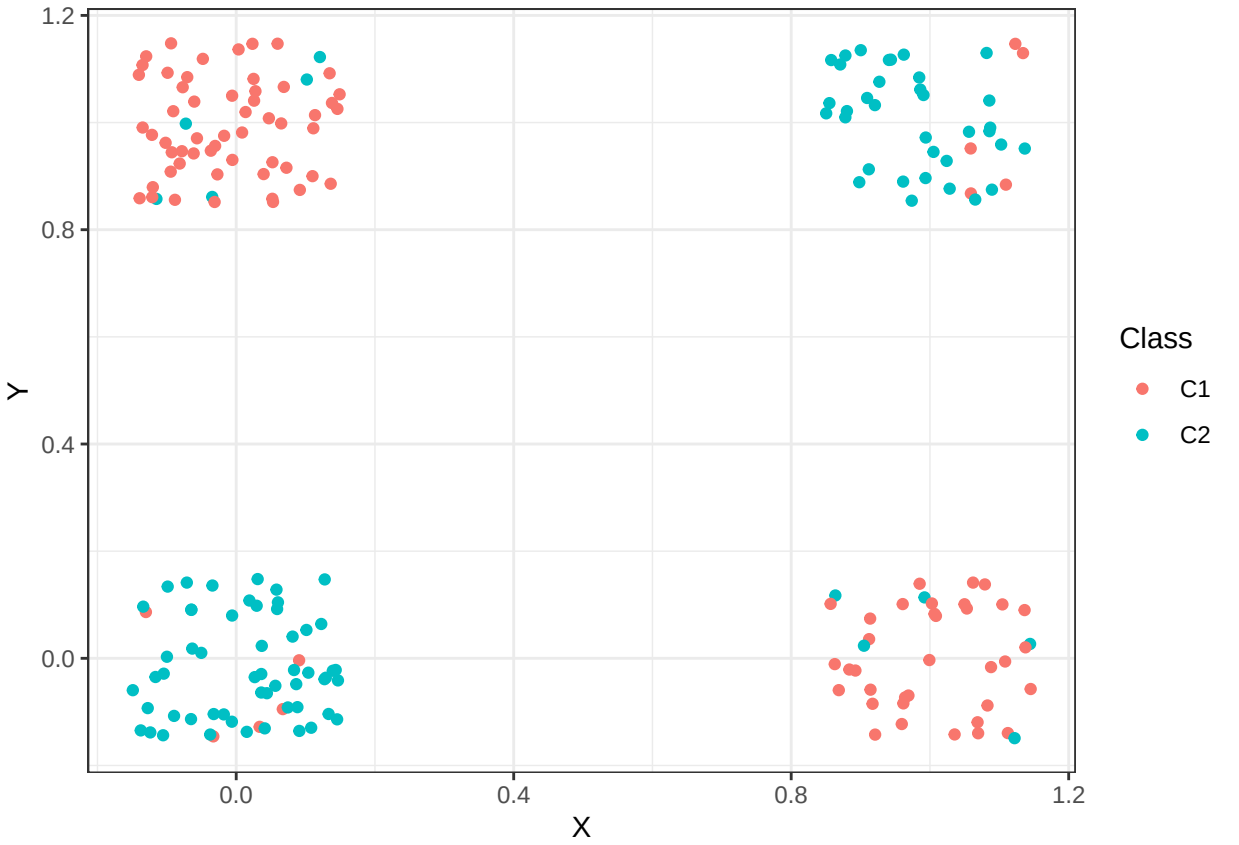
X	Y	C1	C2	misclassified	MisclassificationRate
0	0	5	55	5	0.0833333
0	1	55	5	5	0.0833333
1	0	35	5	5	0.1250000
1	1	5	35	5	0.1250000

```
table <- Table7 %>%
  group_by(X, Z) %>%
  summarise(C1 = sum(C1),
            C2 = sum(C2)) %>%
  mutate(
    misclassified = pmin(C1, C2),
    MisclassificationRate = pmin(C1, C2) / (C1+C2))

kable(table) %>%
  kable_minimal(full_width = F)
```

X	Z	C1	C2	misclassified	MisclassificationRate
0	0	15	45	15	0.250
0	1	45	15	15	0.250
1	0	15	25	15	0.375
1	1	25	15	15	0.375

```
ggplot(data = Table7_long, aes(x = X, y = Y, color = Class)) +
  geom_jitter(width = 0.15, height = 0.15) +
  theme_bw()
```




```
Table7_long <- Table7_long %>%
  mutate(X = as.factor(X),
         Y = as.factor(Y))

sX <- partysplit(which(names(Table7_long) == "X"), index = 1:2)
sY <- partysplit(which(names(Table7_long) == "Y"), index = 1:2)

test <- partynode(id = 1L, split = sX, kids = list(
  partynode(id = 2L, split = sY, kids = list(
    partynode(3L, info = "C2"),
    partynode(4L, info = "C1"))),
  partynode(5L, split = sY, kids = list(
    partynode(6L, info = "C1"),
    partynode(7L, info = "C2")
  ))
  ))
py <- party(test, Table7_long)

ggparty(py) +
  geom_edge() +
  geom_edge_label() +
  geom_node_label(aes(label = splitvar), ids = "inner") +
  geom_node_label(aes(label = info), ids = "terminal")
```

c) Compare the results of parts (a) and (b). Comment on the suitability of the greedy heuristic used for splitting attribute selection.

We see that using the greedy heuristic is no suitable for this data, as we end up with a 30% mis-classification rate compared to a 10% mis-classification rate when taking an initial step that may atr first seem "sun-opmtimal"