ISLR Chapter 8 Exercises

Abhirup Sen

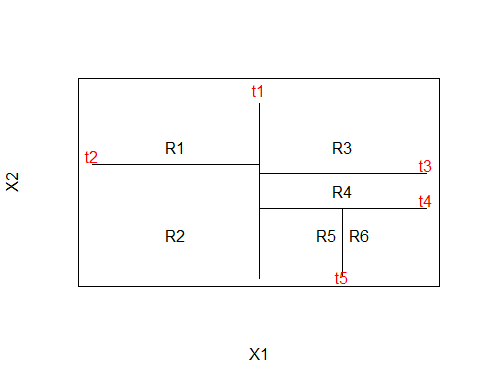
02/06/2021

**1:** Draw an example (of your own invention) of a partition of two-dimensional feature space that could result from recursive binary splitting. Your example should contain at least six regions. Draw a decision tree corresponding to this partition. Be sure to label all aspects of your figures, including the regions R1,R2,…, the cutpoints t1,t2,…, and so forth.

Hint: Your result should look something like Figures 8.1 and 8.2.

#library(rpart)  
#library(rpart.plot)

pts=data.frame(c(25,25,75,75,70,80),c(75,25,75,50,25,25))  
  
plot(pts,xlim=c(0,100),ylim=c(0,110),pch="",xlab="X1",ylab="X2",xaxt='n',yaxt='n')  
  
lines(x = c(50,50), y = c(0,100))  
lines(x = c(0,50), y = c(65,65))  
lines(x = c(50,100), y = c(60,60))  
lines(x = c(50,100), y = c(40,40))  
lines(x = c(75,75), y = c(1,40))  
  
text(x = 50, y = 108, labels = c("t1"), col = "red")  
text(x = 0, y = 70, labels = c("t2"), col = "red")  
text(x = 100, y = 65, labels = c("t3"), col = "red")  
text(x = 100, y = 45, labels = c("t4"), col = "red")  
text(x = 75, y = 1, labels = c("t5"), col = "red")  
  
text(pts,labels=paste("R",1:6,sep=""))

 **2:** It is mentioned in Section 8.2.3 that boosting using depth-one trees (or stumps) leads to an additive model: that is, a model of the form

f(X)=∑j=1pfj(Xj)

Explain why this is the case. You can begin with (8.12) in Algorithm 8.2.

Ans : If only stumps are considered, then for a predictor Xj the equation for its stump takes the form;

Now boosting considers several stumps on the set of predictors where the previously chosen predictor may be chosen or not. If another stump is created for a previously chosen predictor Xi, then we can define as follows;

which if Kj1<Kj2 is equivalent to the equation,

Which can be seen as adding a branch to the stump. Thus, since all functions solely depend on a single predictor the model takes the form of Where fj(Xj)=1λfj^(xj).

**3**: Consider the Gini index, classification error, and entropy in a simple classification setting with two classes.

A: We know **Classification Error:**

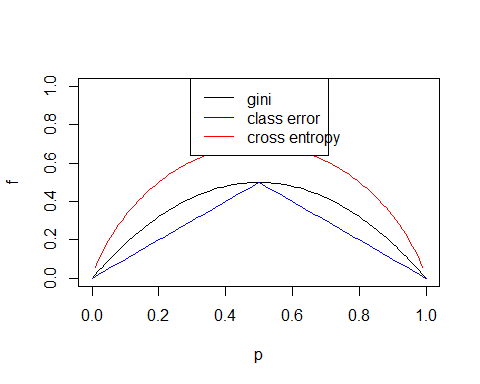
**Gini Index:**

**Entropy:**

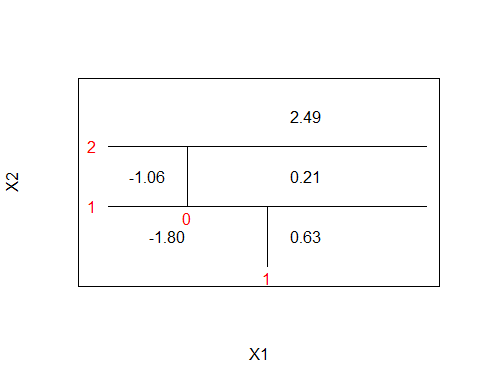
e.g. in R5 of question 1:

E=1−max{67,17}≈0.1429 G=67(1−67)+17(1−17)≈0.2449 D=−67log(67)−17log(17)≈0.4101

p=seq(0,1,0.01)  
  
gini= 2\*p\*(1-p)  
classerror= 1-pmax(p,1-p)  
crossentropy= -(p\*log(p)+(1-p)\*log(1-p))  
  
plot(NA,NA,xlim=c(0,1),ylim=c(0,1),xlab='p',ylab='f')  
  
lines(p,gini,type='l')  
lines(p,classerror,col='blue')  
lines(p,crossentropy,col='red')  
  
legend(x='top',legend=c('gini','class error','cross entropy'),  
 col=c('black','blue','red'),lty=1,text.width = 0.22)

 **4(a)** Sketch the Tree (given the feature space partition) Q: Sketch the tree corresponding to the partition of the predictor space illustrated in the left-hand panel of Figure 8.12. The numbers inside the boxes indicate the mean of Y within each region. **4(b)** Sketch the Feature Space Partition (given the tree) Q: Create a diagram similar to the left-hand panel of Figure 8.12, using the tree illustrated in the right-hand panel of the same figure. You should divide up the predictor space into the correct regions, and indicate the mean for each region.

X1=seq(0,3,0.1)  
X2=seq(0,3,0.1)  
  
plot(NA,NA,xlim=c(-1.2,3),ylim=c(-0.2,3),pch="",xlab="X1",ylab="X2",xaxt='n',yaxt='n')  
pts=data.frame(c(1.5,-0.5,1.5,-0.25,1.50),c(2.5,1.5,1.5,0.5,0.5))  
  
text(pts,labels=c('2.49','-1.06','0.21','-1.80','0.63'))  
text(x = 1, y = -0.2, labels = c("1"), col = "red")  
text(x = 0, y = 0.8, labels = c("0"), col = "red")  
text(x = -1.2, y = 1, labels = c("1"), col = "red")  
text(x = -1.2, y = 2, labels = c("2"), col = "red")  
   
lines(x = c(-1,3), y = c(1,1))  
lines(x = c(-1,3), y = c(2,2))  
lines(x = c(0,0), y = c(1,2))  
lines(x = c(1,1), y = c(0,1))

 **5. Bagged Probabilities to Class Predictions** Q: Suppose we produce ten bootstrapped samples from a data set containing red and green classes. We then apply a classification tree to each bootstrapped sample and, for a specific value of X, produce 10 estimates of P(Class is Red|X):

**0.1,0.15,0.2,0.2,0.55,0.6,0.6,0.65,0.7, and 0.75.**

There are two common ways to combine these results together into a single class prediction. One is the majority vote approach discussed in this chapter. The second approach is to classify based on the average probability. In this example, what is the final classification under each of these two approaches?

probs <- c(0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, 0.75)

In the **majority vote** approach we assign the observation to the class that occurs the most. Thus, we count the number of assignments done to each class by making use of a cutoff value.

sum(probs >= 0.5) # number of 'Red' predictions

## [1] 6

sum(probs < 0.5) # number of 'Green' predictions

## [1] 4

ifelse(sum(probs >= 0.5) > sum(probs < 0.5), "Red", "Green")

## [1] "Red"

As described in the question, when **average probability** is used the average is taken from the estimated probabilities that result from the bagging model. The average is 0.45, which determines that X does not belong to the Red class.

mean(probs) # average P(Red)

## [1] 0.45

ifelse(mean(probs) >= 0.5, "Red", "Green")

## [1] "Green"

**6. Regression Tree Algorithm** Q: Provide a detailed explanation of the algorithm that is used to fit a regression tree.

A regression tree performs subdivisions of the predictor space. The algorithm considers one split at a time in the set of predictors, where the split that is chosen is the one that achieves the most reduction in the RSS. This step will be repeated until threshold values are met (or cannot be met any longer such as minimum number of observations necessary in each node). When predictions are made, new observations travel down the branches of a tree, and upon reaching a leaf node, the average of the observations contained in that leaf is returned.