Stat 760 Final Project

Jakob Lovato

5/10/2022

This project considers data generated based on a linear separator. I discuss the relationship of the cut that defines the classes of the data with the average of many random linear cuts that perfectly classify the data, which comes very close to the maximum marginal classifier. I then consider linear classifiers that misclassify one or more of the data points, and how the pool of viable cuts relates to the number of misclassified points, and perhaps if there is a way to calculate a weighted average of the random cuts that takes into account the number of misclassified points.

Part 1

I start by generating a random cut in a unit square, then generate 50 data points and assign them to class +1 or -1 based on the generated cut. The plot below shows the "ground truth"; the linear cut that defines the class of the data.

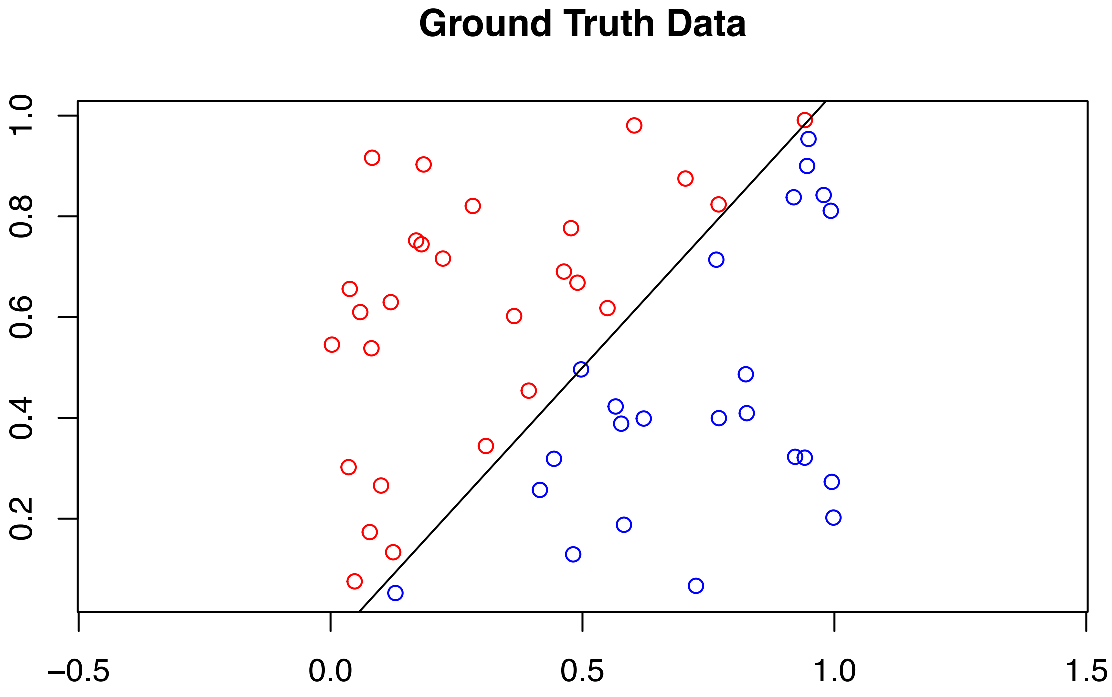
```r
#generate a random cut
generateInitialCut <- function(){
  #always split through (0.5, 0.5) to avoid extreme cuts with few or no points of one class
  X <- 0.5
  Y <- 0.5
  a <- runif(1, min = -1, max = 1)
  b <- runif(1, min = -1, max = 1)
  c <- a * X + b * Y
  return(data.frame(a = a, b = b, cut = c))
}

#similar to the last function, but this one isn't always anchored at (0.5, 0.5)
#to allow for more diverse cuts to be generated later on
#(X,Y) is restricted to being +/- 0.2 from 0.5 in order to speed up code
generateCut <- function(){
  X <- runif(1, min = 0.3, max = 0.7)
  Y <- runif(1, min = 0.3, max = 0.7)
  a <- runif(1, min = -1, max = 1)
  b <- runif(1, min = -1, max = 1)
  c <- a * X + b * Y
  return(data.frame(a = a, b = b, cut = c))
}

#create data in two classes based on that cut (within a unit square, for simplicity)
generateData <- function(n, cut){
  x <- c()
  y <- c()
  class <- c()
  for(i in 1:n){
    x <- c(x, runif(1, min = 0, max = 1))
    y <- c(y, runif(1, min = 0, max = 1))
    ifelse((cut$a * x[i]) + (cut$b * y[i]) >= cut$cut,
           class <- c(class, 1), class <- c(class, -1))
  }
  return(data.frame(x, y, class))
}

#number of points to generage
n <- 50
trueCut <- generateInitialCut()
data <- generateData(n, trueCut)

plot(data$x, data$y, col = c("red", "blue")[as.factor(data$class)], asp = 1,
     xlab = "", ylab = "", main = "Ground Truth Data")
curve((trueCut$cut - trueCut$a * x) / trueCut$b, add = TRUE)
```

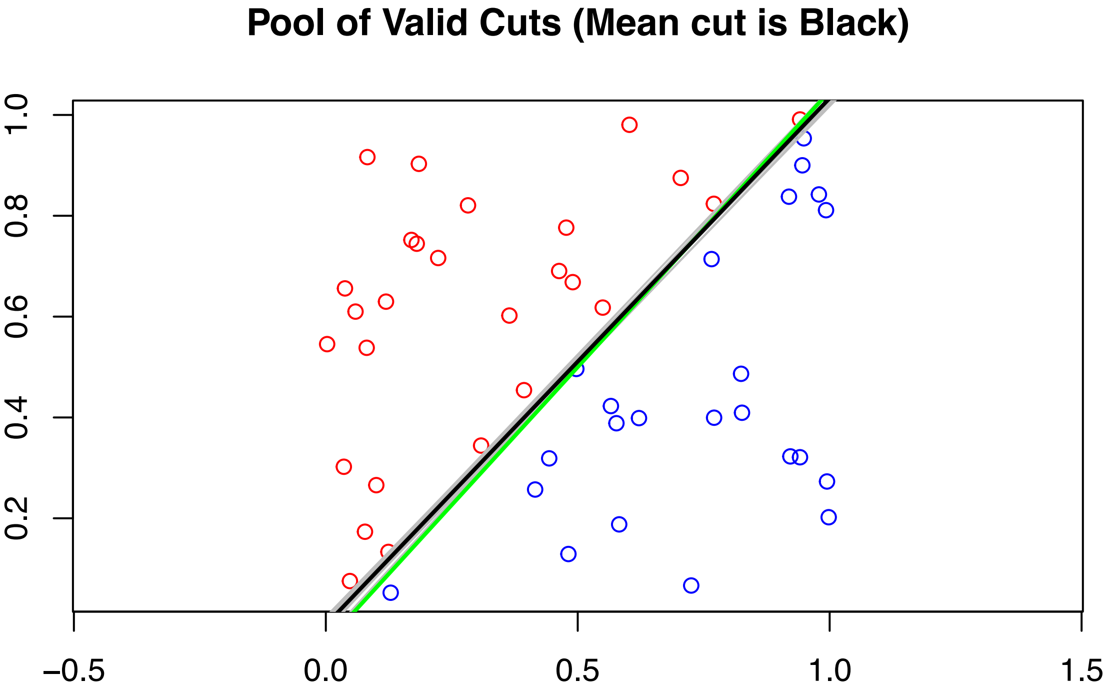


I now generate many random cuts through the data and keep only those that perfectly separate the two classes (valid cuts). The plot below shows the data with all of the valid cuts (drawn in light grey) as well as the mean of the valid cuts (drawn in black). The original cut used to generate the data is plotted in green.

```
#generate new cuts and keep the ones that perfectly separate the data (valid cuts)
generateValidCuts <- function(numCuts, data){
  #create blank data frame
  validCuts <- data.frame(a = numeric(), b = numeric(), cut = numeric())
  while(nrow(validCuts) < numCuts){
    cut <- generateCut()
    tempClass <- c()
    for(i in 1:nrow(data)){
      ifelse((cut$a * data[i,]$x) + (cut$b * data[i,]$y) >= cut$cut,
             tempClass <- c(tempClass, 1), tempClass <- c(tempClass, -1))
    }
    if(identical(data$class, tempClass)){
      #print("success")
      validCuts <- rbind(validCuts, data.frame(a = cut$a, b = cut$b, cut = cut$cut))
    }
    else{
      next
    }
  }
  return(validCuts)
}

#number of valid cuts to find
numCuts <- 10
validCuts <- generateValidCuts(numCuts, data)
```

```
plot(data$x, data$y, col = c("red", "blue")[as.factor(data$class)], asp = 1,
     main = "Pool of Valid Cuts (Mean cut is Black)", xlab = "", ylab = "")
for(i in 1:nrow(validCuts)){
  curve((validCuts[i,]$cut - validCuts[i,]$a * x) / validCuts[i,]$b,
        col = "grey", add = TRUE)
}
meanCut <- data.frame(a = mean(validCuts$a), b = mean(validCuts$b),
                      cut = mean(validCuts$cut))
#plot original true cut in green
curve((trueCut$cut - trueCut$a * x) / trueCut$b, add = TRUE, lwd = 2, col = "green")
#plot mean of valid cuts in black
curve((meanCut$cut - meanCut$a * x) / meanCut$b, lwd = 2, add = TRUE)
```



Now I compare the mean cut of all the valid cuts generated to the ground truth cut used to generate the data.

```
trueCut$cut
```

```
## [1] 0.01519766
```

```
meanCut$cut
```

```
## [1] 0.007359055
```

```
paste("The difference between the true cut and the mean of the valid cuts is",
      round(trueCut$cut - meanCut$cut, 5))
```

```
## [1] "The difference between the true cut and the mean of the valid cuts is 0.00784"
```

More meaningful for interpretation is to compare the slope and intercept of the two lines:

```
trueSlope <- -trueCut$a / trueCut$b
trueIntercept <- trueCut$cut / trueCut$b
meanSlope <- -meanCut$a / meanCut$b
meanIntercept <- meanCut$cut / meanCut$b

paste("The difference between slopes is", round(trueSlope - meanSlope, 5))
```

```
## [1] "The difference between slopes is 0.04948"
```

```
paste("The difference between the intercepts is",
      round(trueIntercept - meanIntercept, 5))
```

```
## [1] "The difference between the intercepts is -0.03562"
```

We can see that the difference between the true cut and the mean of the valid cuts is quite small, so the mean of all valid cuts is very similar to the ground truth.
I now repeat the process for multiple different ground truth cuts and compare them to the mean cuts.

```
n <- 50
table <- data.frame(trueCut = numeric(), meanCut = numeric())
for(iteration in 1:25){
  trueCut <- generateInitialCut()
  data <- generateData(n, trueCut)
  numCuts <- 10
  validCuts <- generateValidCuts(numCuts, data)
  meanCut <- data.frame(a = mean(validCuts$a), b = mean(validCuts$b),
                        cut = mean(validCuts$cut))

  table <- rbind(table, data.frame(trueCut = trueCut$cut, meanCut = meanCut$cut))
}

table
```

```
##        trueCut     meanCut
## 1  -0.39932981 -0.37265050
## 2  -0.83598441 -0.47236343
## 3  -0.34347154 -0.42488140
## 4  -0.26622230 -0.65627090
## 5   0.49160115  0.35381161
## 6   0.69287878  0.51193850
## 7   0.38779417  0.41037381
## 8   0.16972706  0.46728700
## 9   0.06870029  0.12776836
## 10 -0.43529650 -0.40070014
## 11 -0.11357478 -0.24704218
## 12 -0.06843669 -0.13419003
## 13  0.02258821  0.04237195
## 14  0.59242175  0.42358376
## 15  0.13844451  0.15072982
## 16  0.07409367  0.10568371
## 17 -0.03737212 -0.16746338
## 18  0.13798394  0.18138729
## 19 -0.06275997 -0.07761083
## 20 -0.11658933 -0.11131354
## 21 -0.62010019 -0.63712845
## 22  0.41186580  0.32347946
## 23 -0.11923329 -0.11150878
## 24 -0.52562873 -0.37959741
## 25 -0.32822366 -0.27853451
```

```
MSE <- mean((table$trueCut - table$meanCut)^2)
MSE
```

```
## [1] 0.02161424
```

The MSE (of difference from mean cuts with true cut) for 25 sets of data is 0.0216142, which is quite low, showing the mean of valid cuts is very close to the true cut.

Part 2

I now repeat the process but with 1, 2, 3… points misclassified to try and find how to compute a weighted average of the random cuts to take into account the number of misclassified points. I start by misclassifying one point:

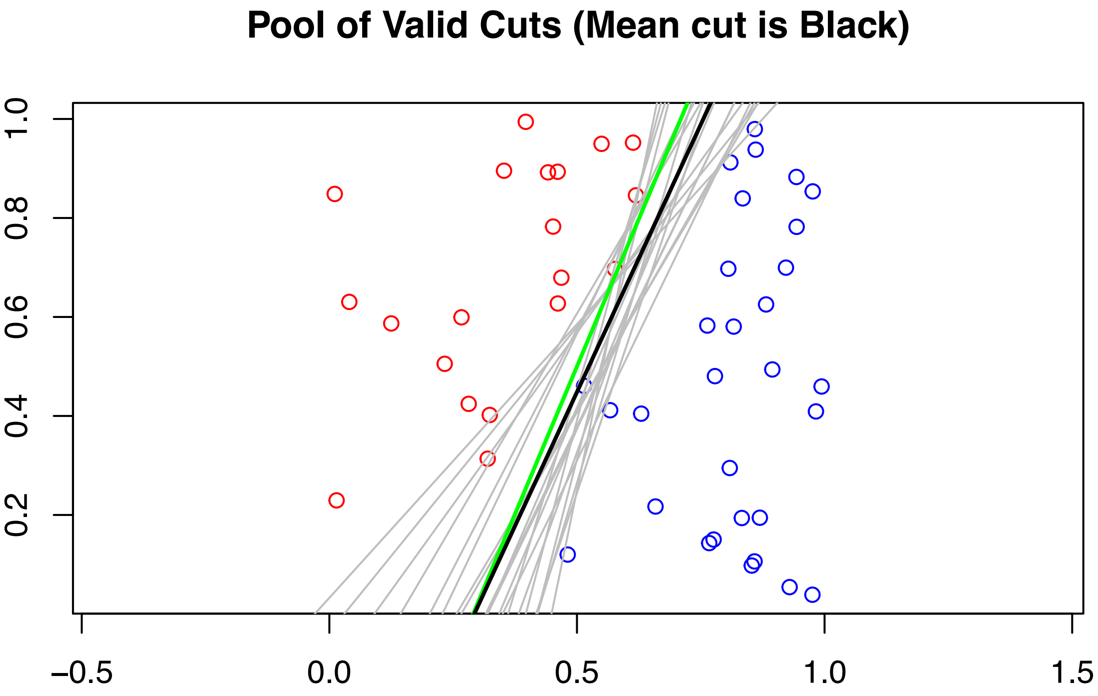
```
#modify previous function to misclassify one point
generateValidCuts1Mis <- function(numCuts, data){
  #create blank data frame
  validCuts <- data.frame(a = numeric(), b = numeric(), cut = numeric())
  while(nrow(validCuts) < numCuts){
    cut <- generateCut()
    tempClass <- c()
    for(i in 1:nrow(data)){
      ifelse((cut$a * data[i,]$x) + (cut$b * data[i,]$y) >= cut$cut,
             tempClass <- c(tempClass, 1), tempClass <- c(tempClass, -1))
    }
    if(sum(data$class != tempClass) == 1){
      #print("success")
      validCuts <- rbind(validCuts, data.frame(a = cut$a, b = cut$b, cut = cut$cut))
    }
    else{
      next
    }
  }
  return(validCuts)
}

trueCut1 <- generateInitialCut()
n <- 50
data <- generateData(n, trueCut1)
```

The plot below shows the more scattered pool of valid cuts, due to a wider variety of ways for the cut to be drawn since each cut will misclassify one point.

```
#generate more valid cuts to get better mean line since there is bound to be more variation in the line
numCuts <- 20
validCuts1Mis <- generateValidCuts1Mis(numCuts, data)

plot(data$x, data$y, col = c("red", "blue")[as.factor(data$class)], asp = 1,
     main = "Pool of Valid Cuts (Mean cut is Black)", xlab = "", ylab = "")
for(i in 1:nrow(validCuts1Mis)){
  curve((validCuts1Mis[i,]$cut - validCuts1Mis[i,]$a * x) / validCuts1Mis[i,]$b,
        col = "grey", add = TRUE)
}
meanCut <- data.frame(a = mean(validCuts1Mis$a), b = mean(validCuts1Mis$b),
                      cut = mean(validCuts1Mis$cut))
#plot original true cut in green
curve((trueCut1$cut - trueCut1$a * x) / trueCut1$b, add = TRUE, lwd = 2, col = "green")
#plot mean of valid cuts in black
curve((meanCut$cut - meanCut$a * x) / meanCut$b, lwd = 2, add = TRUE)
```



I now repeat the process for multiple different ground truth cuts and compare them to the mean cuts.

```
n <- 50
table <- data.frame(trueCut = numeric(), meanCut = numeric())
for(iteration in 1:25){
  trueCut <- generateInitialCut()
  data <- generateData(n, trueCut)
  numCuts <- 10
  validCuts1Mis <- generateValidCuts1Mis(numCuts, data)
  meanCut <- data.frame(a = mean(validCuts1Mis$a), b = mean(validCuts1Mis$b),
                        cut = mean(validCuts1Mis$cut))

  table <- rbind(table, data.frame(trueCut = trueCut$cut, meanCut = meanCut$cut))
}

table
```

```
##         trueCut     meanCut
## 1   0.093254213  0.12320570
## 2  -0.660581040 -0.56355812
## 3   0.212540681  0.25031329
## 4  -0.012799616  0.11260152
## 5  -0.049248442 -0.49739051
## 6   0.289821589  0.35014680
## 7  -0.440165229 -0.58697469
## 8   0.093395130  0.10882661
## 9  -0.005228939 -0.01852131
## 10  0.746582954  0.53074769
## 11 -0.240087217 -0.29628543
## 12  0.130220185  0.24985964
## 13 -0.232486902 -0.26651028
## 14 -0.945995435 -0.65855340
## 15  0.405507404  0.31317127
## 16 -0.627319595 -0.57986009
## 17  0.387699430  0.26436951
## 18 -0.766027279 -0.66544231
## 19  0.585635592  0.48526289
## 20  0.391953703  0.32845067
## 21 -0.590100736 -0.35763920
## 22  0.593697419  0.30539504
## 23 -0.914719050 -0.60645978
## 24 -0.365756096 -0.40409203
## 25 -0.194088347 -0.14584964
```

```
MSE <- mean((table$trueCut - table$meanCut)^2)
MSE
```

```
## [1] 0.02751714
```

Next, the same process is repeated, this time generating a pool of cuts that misclassify 2 points:

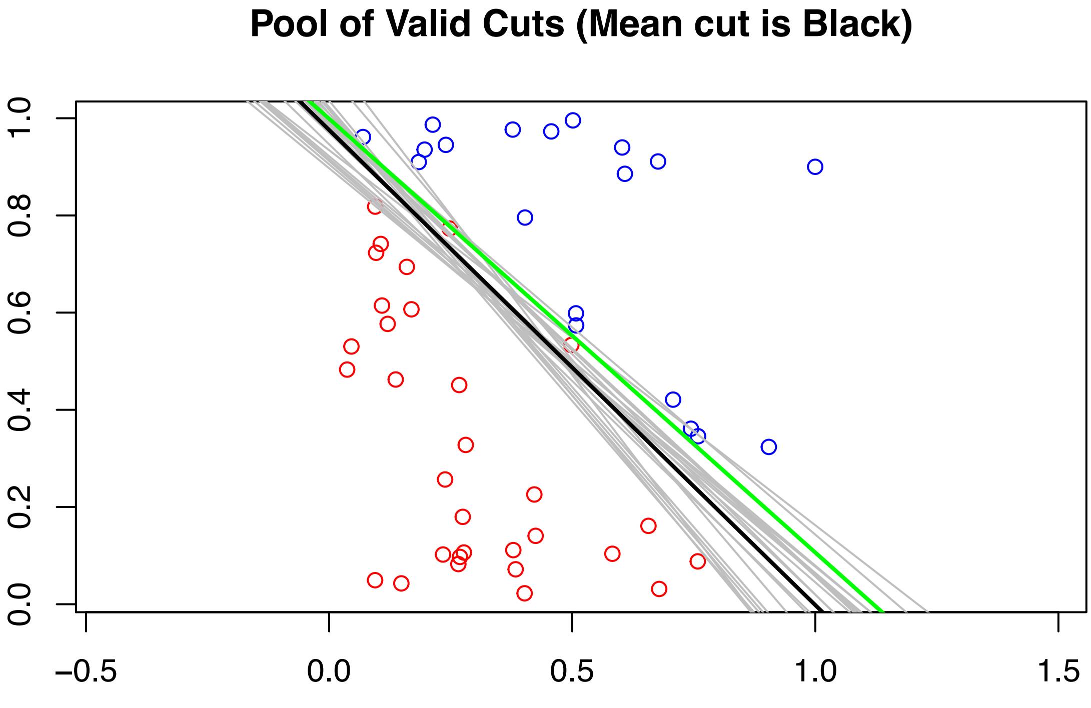
```
#modify previous function to misclassify two points
generateValidCuts2Mis <- function(numCuts, data){
  #create blank data frame
  validCuts <- data.frame(a = numeric(), b = numeric(), cut = numeric())
  while(nrow(validCuts) < numCuts){
    cut <- generateCut()
    tempClass <- c()
    for(i in 1:nrow(data)){
      ifelse((cut$a * data[i,]$x) + (cut$b * data[i,]$y) >= cut$cut,
             tempClass <- c(tempClass, 1), tempClass <- c(tempClass, -1))
    }
    if(sum(data$class != tempClass) == 2){
      #print("success")
      validCuts <- rbind(validCuts, data.frame(a = cut$a, b = cut$b, cut = cut$cut))
    }
    else{
      next
    }
  }
  return(validCuts)
}
```

```r
trueCut2 <- generateCut()
n <- 50
data <- generateData(n, trueCut2)
```

In the plot below, we see an even more scattered pool of valid cuts, since the constraints are becoming weaker; we now allow two points to be misclassified.

```r
#generate more valid cuts to get better mean line since there is bound to be more variation in the line
numCuts <- 20
validCuts2Mis <- generateValidCuts2Mis(numCuts, data)

plot(data$x, data$y, col = c("red", "blue")[as.factor(data$class)], asp = 1,
     main = "Pool of Valid Cuts (Mean cut is Black)", xlab = "", ylab = "")
for(i in 1:nrow(validCuts2Mis)){
  curve((validCuts2Mis[i,]$cut - validCuts2Mis[i,]$a * x) / validCuts2Mis[i,]$b,
        col = "grey", add = TRUE)
}
meanCut <- data.frame(a = mean(validCuts2Mis$a), b = mean(validCuts2Mis$b),
                      cut = mean(validCuts2Mis$cut))
#plot original true cut in green
curve((trueCut2$cut - trueCut2$a * x) / trueCut2$b, add = TRUE, lwd = 2, col = "green")
#plot mean of valid cuts in black
curve((meanCut$cut - meanCut$a * x) / meanCut$b, lwd = 2, add = TRUE)
```



I now repeat the process for multiple different ground truth cuts and compare them to the mean cuts.

```r
n <- 50
table <- data.frame(trueCut = numeric(), meanCut = numeric())
for(iteration in 1:25){
  trueCut <- generateInitialCut()
  data <- generateData(n, trueCut)
  numCuts <- 10
  validCuts2Mis <- generateValidCuts2Mis(numCuts, data)
  meanCut <- data.frame(a = mean(validCuts2Mis$a), b = mean(validCuts2Mis$b),
                        cut = mean(validCuts2Mis$cut))

  table <- rbind(table, data.frame(trueCut = trueCut$cut, meanCut = meanCut$cut))
}

table
```

```
##          trueCut      meanCut
## 1  -0.153844201 -0.355153235
## 2   0.651619976  0.439619154
## 3   0.102647814  0.008513561
## 4  -0.814106097 -0.513696983
## 5   0.701626615  0.486970580
## 6  -0.178652080 -0.217018610
## 7   0.145405910  0.191947619
## 8  -0.054152363 -0.102905573
## 9   0.212861586  0.133505398
## 10 -0.546098180 -0.652764705
## 11 -0.305275230 -0.354932452
## 12  0.055088322  0.105659926
## 13 -0.031103677 -0.049360840
## 14 -0.608949322 -0.452314615
## 15 -0.474220649 -0.428368044
## 16  0.775792773  0.487945258
## 17 -0.204096746 -0.092317533
## 18 -0.311029083 -0.569410628
## 19  0.118205487  0.283699686
## 20 -0.086339845  0.014668194
## 21 -0.006747227 -0.064603974
## 22 -0.226222132 -0.290144171
## 23  0.374015530  0.636944427
## 24  0.500197933  0.413205847
## 25  0.587488999  0.626492981
```

```
MSE <- mean((table$trueCut - table$meanCut)^2)
MSE
```

```
## [1] 0.02286777
```

The MSE now is not very significantly different than the MSE for mis-classifying one point. Finally, I will repeat the process once more, but this time I decide to misclassify five data points, to see if I can get a significantly different result from before. Up to now, the MSE of differences between the true cut and mean cuts has not changed much, and the mean cut tends to not differ from the true cut very much. I suspect that if I am to see any significantly different results, they will show up with five misclassified points.

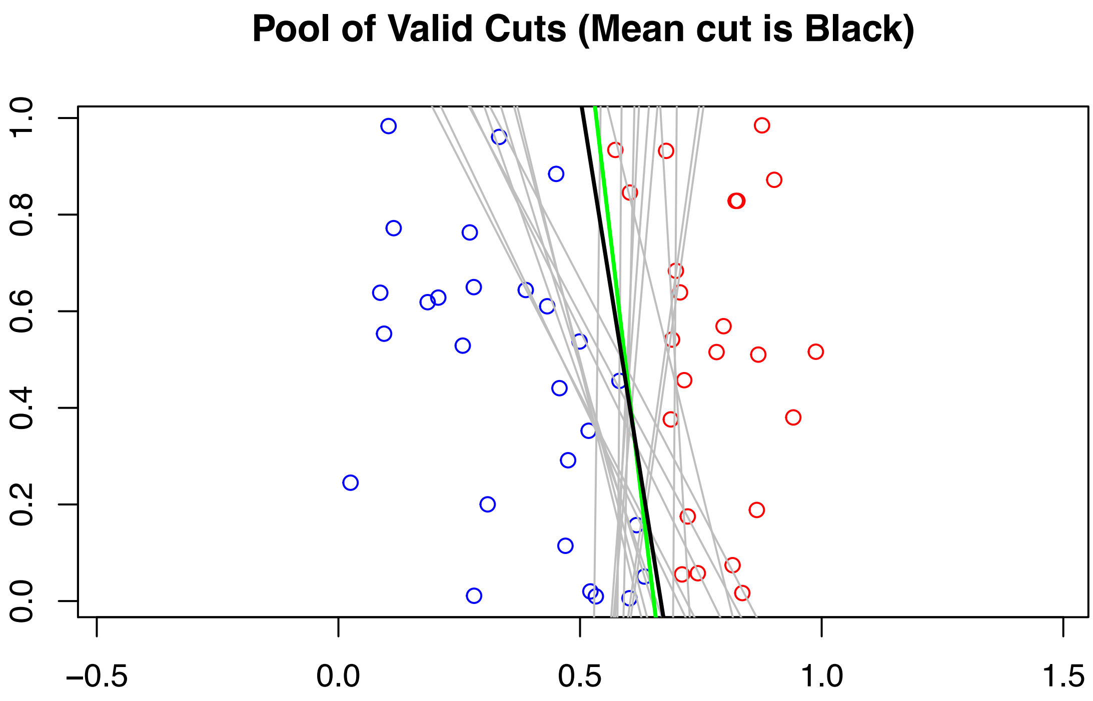
```
#modify previous function to misclassify five points
generateValidCuts5Mis <- function(numCuts, data){
  #create blank data frame
  validCuts <- data.frame(a = numeric(), b = numeric(), cut = numeric())
  while(nrow(validCuts) < numCuts){
    cut <- generateCut()
    tempClass <- c()
    for(i in 1:nrow(data)){
      ifelse((cut$a * data[i,]$x) + (cut$b * data[i,]$y) >= cut$cut,
             tempClass <- c(tempClass, 1), tempClass <- c(tempClass, -1))
    }
    if(sum(data$class != tempClass) == 5){
      #print("success")
      validCuts <- rbind(validCuts, data.frame(a = cut$a, b = cut$b, cut = cut$cut))
    }
    else{
      next
    }
  }
  return(validCuts)
}

trueCut5 <- generateCut()
n <- 50
data <- generateData(n, trueCut5)
```

As predicted, the plot below shows the most scattered pool of valid cuts yet. The constraints for valid cuts are now much weaker, allowing for five misclassified points.

```
numCuts <- 20
validCuts5Mis <- generateValidCuts5Mis(numCuts, data)

plot(data$x, data$y, col = c("red", "blue")[as.factor(data$class)], asp = 1,
     main = "Pool of Valid Cuts (Mean cut is Black)", xlab = "", ylab = "")
for(i in 1:nrow(validCuts5Mis)){
  curve((validCuts5Mis[i,]$cut - validCuts5Mis[i,]$a * x) / validCuts5Mis[i,]$b,
        col = "grey", add = TRUE)
}
meanCut <- data.frame(a = mean(validCuts5Mis$a), b = mean(validCuts5Mis$b),
                      cut = mean(validCuts5Mis$cut))
#plot original true cut in green
curve((trueCut5$cut - trueCut5$a * x) / trueCut5$b, add = TRUE, lwd = 2, col = "green")
#plot mean of valid cuts in black
curve((meanCut$cut - meanCut$a * x) / meanCut$b, lwd = 2, add = TRUE)
```



I now repeat the process for multiple different ground truth cuts and compare them to the mean cuts. The mean cut above does appear to be the most significantly different than the true cut compared to the 1 and 2 point misclassification models. So I expect to see a significant change in MSE now.

```
n <- 50
table <- data.frame(trueCut = numeric(), meanCut = numeric())
for(iteration in 1:25){
  trueCut <- generateInitialCut()
  data <- generateData(n, trueCut)
  numCuts <- 10
  validCuts5Mis <- generateValidCuts5Mis(numCuts, data)
  meanCut <- data.frame(a = mean(validCuts5Mis$a), b = mean(validCuts5Mis$b),
                        cut = mean(validCuts5Mis$cut))

  table <- rbind(table, data.frame(trueCut = trueCut$cut, meanCut = meanCut$cut))
}

table
```

```
##        trueCut      meanCut
## 1  -0.45991886 -0.388809341
## 2   0.66692670  0.474048229
## 3  -0.22678310 -0.236097701
## 4   0.11603109  0.128337984
## 5   0.18498794  0.306311505
## 6  -0.10457996 -0.107755915
## 7  -0.16836070 -0.147484935
## 8  -0.69428479 -0.584912187
## 9  -0.51182630 -0.318567525
## 10 -0.13487855 -0.299724571
## 11  0.29464817  0.147204077
## 12  0.37144445  0.158618095
## 13 -0.38127463 -0.484276024
## 14  0.07448808  0.455136420
## 15 -0.22135726 -0.381573843
## 16 -0.64958623 -0.474717278
## 17 -0.79307515 -0.556607719
## 18 -0.01799622 -0.172411535
## 19 -0.36125569 -0.385679426
## 20  0.65299143  0.633412167
## 21 -0.11721529 -0.285958974
## 22 -0.11013213 -0.017291010
## 23 -0.70272135 -0.427128746
## 24  0.01555437  0.006952669
## 25 -0.16952211 -0.290504708
```

```
MSE <- mean((table$trueCut - table$meanCut)^2)
MSE
```

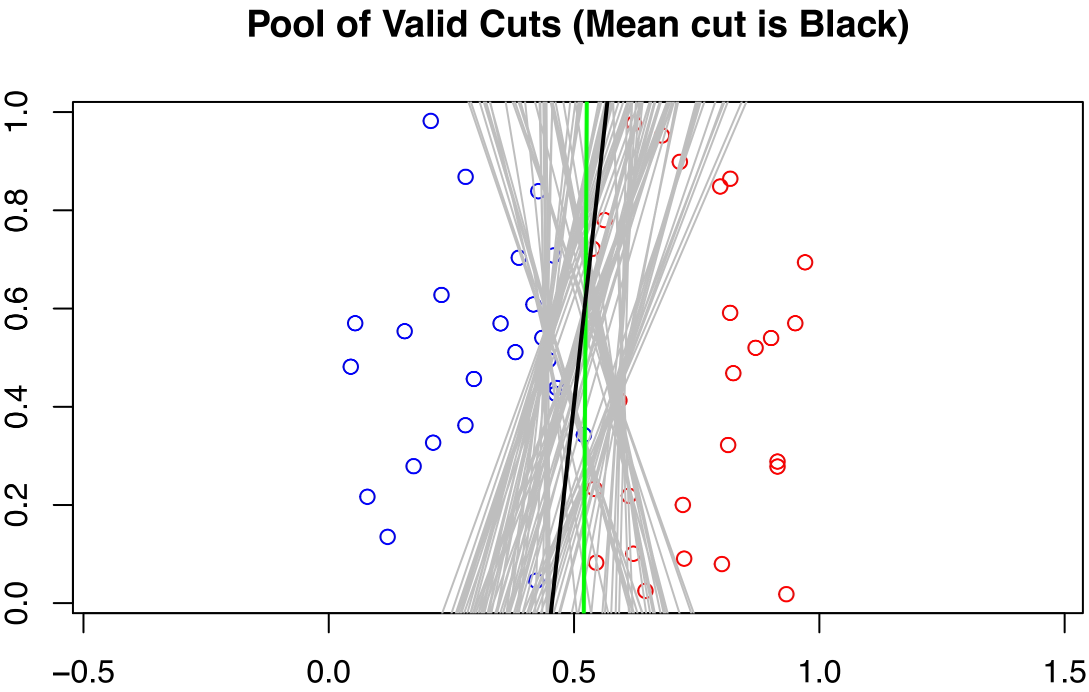
```
## [1] 0.02485721
```

Surprisingly, the MSE is still quite low, and not significantly different from the MSE for the 1 and 2 point misclassification models. In some way, this makes sense. Individual cuts misclassifying n = 1, 2, 3, . . . points will definitely differ from the true cut that perfectly separates the data. However, there should be about equal ways (assuming uniformly random data) to misclassify n points with a slope steeper than the true cut and a slope more shallow than the true cut. Or, shifted up or down from the true cut. So the average of the valid cuts should be about close to the true cut each time. While the mean of valid cuts (black) looked different from the true cut (green) on some of the plots, once the process is reapeated many times, the differences become less aparent when we look at the MSE.

I decided to, one more time, get the mean of many valid cuts misclassifying five points. But this time, I wanted many more, so I plotted 100 cuts with the mean cut and true cut to visually analyze what happens with more and more cuts. In the plot below, we can see that again, even with a pool of 100 valid cuts, the mean cut is not much different than the true cut.

```
trueCut5 <- generateCut()
n <- 50
data <- generateData(n, trueCut5)
numCuts <- 100
validCuts5Mis <- generateValidCuts5Mis(numCuts, data)

plot(data$x, data$y, col = c("red", "blue")[as.factor(data$class)], asp = 1,
     main = "Pool of Valid Cuts (Mean cut is Black)", xlab = "", ylab = "")
for(i in 1:nrow(validCuts5Mis)){
  curve((validCuts5Mis[i,]$cut - validCuts5Mis[i,]$a * x) / validCuts5Mis[i,]$b,
        col = "grey", add = TRUE)
}
meanCut <- data.frame(a = mean(validCuts5Mis$a), b = mean(validCuts5Mis$b),
                      cut = mean(validCuts5Mis$cut))
#plot original true cut in green
curve((trueCut5$cut - trueCut5$a * x) / trueCut5$b, add = TRUE, lwd = 2, col = "green")
#plot mean of valid cuts in black
curve((meanCut$cut - meanCut$a * x) / meanCut$b, lwd = 2, add = TRUE)
```



At the end of the day, this process is quite similar to bootstrapping. Just instead of using subsets of the data, we intentionally misclassify a certain number of points. Either way, the mean of all the valid cuts (given a large enough pool of valid cuts) will look close to the true cut. So it seems that my goal of finding a weighted average of the random cuts to take into account the number of misclassified points is not a meaningful task.

Eventually, if we tried to calculate the average of cuts that misclassify more and more data, the individual cuts would become further and further from the true cut. At a certain point, I think we would be only able to truly misclassify half of the data, and at this point, I would expect the average cut to be nearly perpendicular to the true cut. If we were to misclassify 100% of the data, then the cut would once again look very close to the true cut (only here, the sign $>$ or $<$ would be flipped from when we generated the data classes).