Seminar Homework

Rishabh

Thursday, January 29, 2015

Contents

1. Wrapper Functions

The idea is to encapsulate frequently used segments of codes into logical functions. Most of these are taken from the sample codes provided in the first seminar

```
#Wrapper functions

#Create variance covariance matrix
sigmaXY <- function(rho, sdX, sdY) {
  covTerm <- rho * sdX * sdY
  VCmatrix <- matrix(c(sdX^2, covTerm, covTerm, sdY^2),
                     2, 2, byrow = TRUE)
  return(VCmatrix)
}

#Create a BI Variate NOrmal distribution
genBVN <- function(n = 1, seed = NA, muXY=c(0,1), sigmaXY=diag(2)) {
  if(!is.na(seed)) set.seed(seed)
  rdraws <- rmvnorm(n, mean = muXY, sigma = sigmaXY)
  return(rdraws)
}

#Create a dataset for classification
animals <- function(rho, sdXY, muXY, noSim, seed = 5341) {
  catsVC     <- sigmaXY(rho[1], sdXY[[1]][1], sdXY[[1]][2])
  dogsVC     <- sigmaXY(rho[2], sdXY[[2]][1], sdXY[[2]][2])
  maggotsVC    <- sigmaXY(rho[3], sdXY[[3]][1], sdXY[[3]][2])

  catsBVN    <- genBVN(noSim[1], seed = 5341, muXY[[1]], catsVC)
  dogsBVN    <- genBVN(noSim[2], seed = 5342, muXY[[2]], dogsVC)
  maggotsBVN <- genBVN(noSim[3], seed = 5343, muXY[[3]], maggotsVC)

  animalsDf <- as.data.frame(rbind(catsBVN,dogsBVN, maggotsBVN))
  Animal <- c(rep("Cats", noSim[1]), rep("Dogs", noSim[2]), rep("Maggots", noSim[3]))
  animalsDf <- cbind(animalsDf, Animal)
  colnames(animalsDf) <- c("weight", "height", "Animal")

  #Provide 1 0 labels to relevant class
  animalsDf <- cbind(animalsDf, labCats = c(rep(1, noAnimals[1]),
                                            rep(0, noAnimals[2]),
                                            rep(0, noAnimals[3])))
  animalsDf <- cbind(animalsDf, labDogs = c(rep(0, noAnimals[1]),
                                            rep(1, noAnimals[2]),
                                            rep(0, noAnimals[3])))
```

```r
  animalsDf <- cbind(animalsDf, labMaggots = c(rep(0, noAnimals[1]),
                                               rep(0, noAnimals[2]),
                                               rep(1, noAnimals[3])))

  return(animalsDf)
}

#maps numerical categories to character labels
numToChar <- function(x) {
  if(which.max(x) ==1) {return ("Cats")}
  else if (which.max(x) == 2){ return ("Dogs")}
  else {return ("Maggots")}}


#Plot simulated values of Animals
plotAnimals <- function(animalDF) {
    p <- ggplot(data = animalsDF, aes(x = height, y = weight,
           colour=Animal, fill=Animal)) +
    geom_point() +
    xlab("Height") +
    ylab("Weight") +
    theme_bw(base_size = 14, base_family = "Helvetica") +
    scale_color_manual("Animal",
       values = c("Boundary" = "grey", "Cats" = "blue", "Dogs" = "red",
                  "Maggots" = "Orange"))

    p
}

#Perform K regressions and classify labels and return coefficiemts and labels
getPrediction <- function(animalsDF){
areYouACat <- lm(labCats ~ weight + height, data = animalsDF)
areYouADog <- lm(labDogs ~ weight + height, data = animalsDF)
areYouAMaggot <- lm(labMaggots ~ weight + height, data = animalsDF)

predict <- cbind(predict(areYouACat), predict(areYouADog), predict(areYouAMaggot))
predictLabel <- apply(predict, 1, numToChar)
return(list(predictLabel, areYouACat, areYouADog, areYouAMaggot))
}

#Plot the decision boundaries
plotBoundary <- function(wCat, wDog, wMag, animalsDF){
  #Calculating points corresponding to boundary lines
  x <- seq(min(animalsDF$height), max(animalsDF$height), 0.01)
  y1 <- -((wCat[3] - wDog[3]) / (wCat[2] - wDog[2])) * x +
    ((wDog[1] - wCat[1]) / (wCat[2] - wDog[2]))
  y2 <- -((wMag[3] - wCat[3]) / (wMag[2] - wCat[2])) * x +
    ((wCat[1] - wMag[1]) / (wMag[2] - wCat[2]))
  y3 <- -((wDog[3] - wMag[3]) / (wDog[2] - wMag[2])) * x +
    ((wMag[1] - wDog[1]) / (wDog[2] - wMag[2]))


  boundDF1 <- data.frame(height = x , weight = y1, Animal=rep("Bound1", length(x)))
```

```r
  boundDF2 <- data.frame(height = x , weight = y2, Animal=rep("Bound2", length(x)))
  boundDF3 <- data.frame(height = x , weight = y3, Animal=rep("Bound3", length(x)))

  plotAnimals(animalsDF) +
    geom_line(data = boundDF1) +
    geom_line(data = boundDF2) +
    geom_line(data = boundDF3) +
    scale_color_manual("Animal",
     values = c("Bound1" = "black", "Boundary" = "grey",
                "Cats" = "blue", "Dogs" = "red", "Maggots" = "orange",
                "Bound2" = "black", "Bound3" = "black")) +
    ggtitle("Decison Boundaries")
  }
```

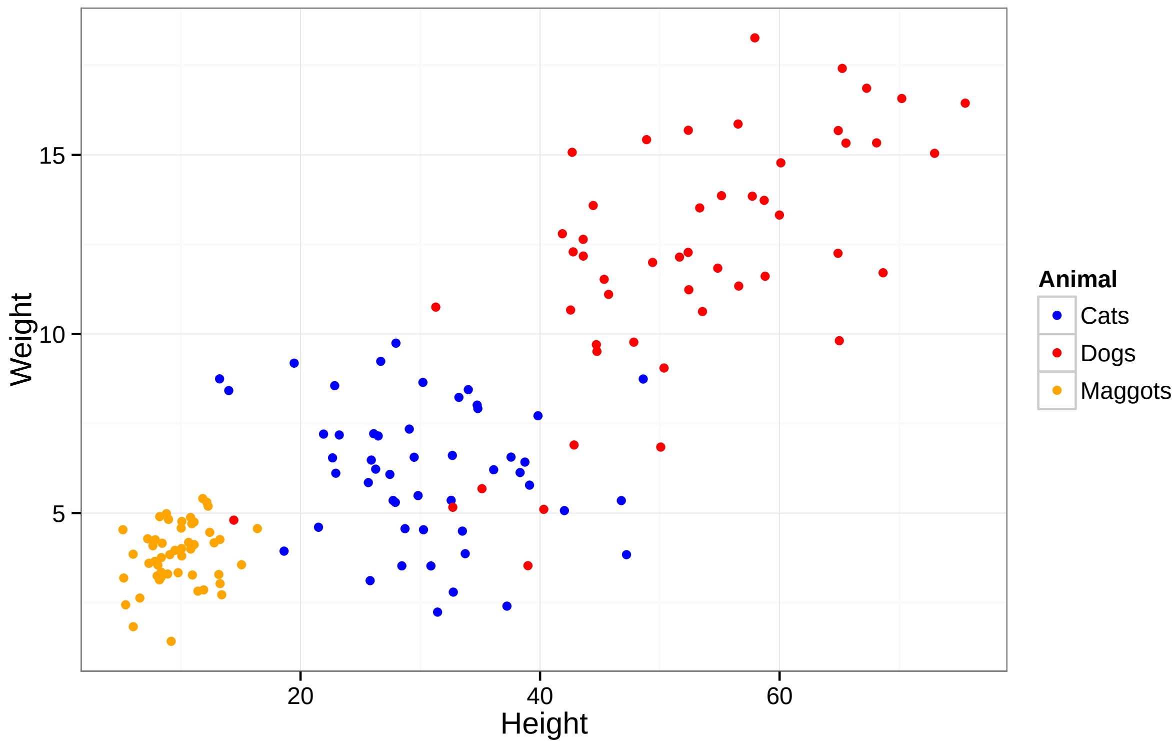
2. This sections deals with simulating heights and weights of 3 animal types.

The animals are cats dogs and maggots. We also display these simulations graphically

```r
#Simulate three variables for categories Cats Dogs and Maggots
noAnimals <- c(50, 50, 50)
rho  <- c(-0.2, 0.8, 0.02)
sdXY <- list(c(2, 8), c(3.5, 12), c(1, 3))
muXY <- list(c(6, 30), c(12, 50), c(4,10))

animalsDF <- animals(rho, sdXY, muXY, noAnimals)

# illustrating the data
plotAnimals(animalsDF)
```



3. K = 3 class classification

1.We construct K classifiers corresponding to K classes and predict the type of animal

```
predictionList <- getPrediction(animalsDF)
predictLabel <- predictionList[[1]]
head(predictLabel)
```

```
##         1         2         3         4         5         6
## "Maggots" "Maggots"    "Cats"    "Cats"    "Cats"    "Dogs"
```

4. Results and interpretation

The table below shows that our classifier works well for Dogs and Maggots but performs very badly for Cats. This is a known issue of using linear regression classification. When K >= 3 masking of classes may occur.

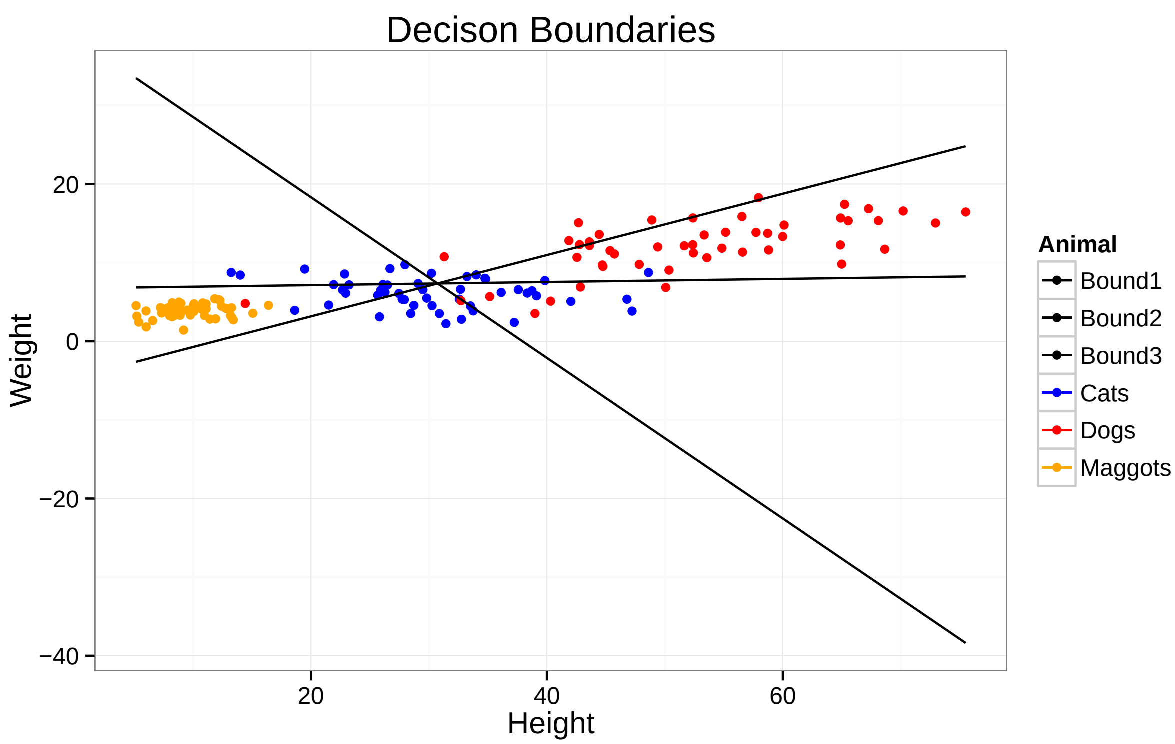
```
#Performance of our classifier
table(animalsDF$Animal, predictLabel)
```

```
##          predictLabel
##           Cats Dogs Maggots
##   Cats      24    7      19
##   Dogs       6   43       1
##   Maggots    0    0      50
```

```r
#Collect regression coefficients of our discriminants
#Will serve as inputs to calculate and plot decison boundaries

wCat <- coef(predictionList[[2]])
wDog <- coef(predictionList[[3]])
wMag <- coef(predictionList[[4]])

plotBoundary(wCat, wDog, wMag, animalsDF)
```



We resolve the boundary line by the following rule:

$$y_k(x) \neq y_j(x) where k \neq j$$

This translates to three sets of two equations whose intersection is the region for a class. Example of a boundary line is with the following structure:

$$weight = -\frac{w_{cat,height} - w_{dog,height}}{w_{cat,weight} - w_{dog,weight}} height + \frac{w_{dog,bias} - w_{cat,bias}}{w_{cat,weight} - w_{dog,weight}}$$

We will have another two equations which will divide our space into there singly connected convex hyperplanes.