Applied Data Science, Assignment 2

Submitted by: Divi Khanna (dk2745)

February 21th, 2014

Contents

1 Question 1

The data displayed below consists of categorical variables which must be converted to binary dummy variables for sparse matrix representation.

```
data <- read.csv("columbia_data_set.csv.csv.csv", stringsAsFactors = TRUE)
```

site.id has a large number of categories which are reduced by selecting 99% of data points and classifying the remaining data points in "*Other*" class.

```
a <- sort(table(data$site.id), decreasing = TRUE)/nrow(data)
a <- subset(a, cumsum(a) < 0.99)
data$site.id[!(is.element(data$site.id, names(a)))] <- "Other"
data$site.id <- as.factor(data$site.id)
str(data$site.id)

##  Factor w/ 137 levels "1","101","102",..: 98 98 3 137 46 58 120 120 68 10 ...
```

A model matrix is created with binary sparse notation, using dummy variables for all categorical data.

```
require(Matrix)

## Loading required package:  Matrix
## Loading required package:  lattice

data$hour <- as.factor(data$hour)
data$browser.id <- as.factor(data$browser.id)
x <- model.matrix(~-1 + impression.id + user.id + day.of.week + hour + site.id +
    ad.size + browser.id + state, data, contrasts.arg = list(day.of.week = contrasts(data$day.of.week,
    contrasts = F), hour = contrasts(data$hour, contrasts = F), site.id = contrasts(data$site.id,
    contrasts = F), ad.size = contrasts(data$ad.size, contrasts = F), browser.id = contrasts(data$browser.id
    contrasts = F), state = contrasts(data$state, contrasts = F)))
x <- as.data.frame(x)
dim(x)

## [1] 100000    236
```

2 Question 2

2.1 Linear Regression

```
x1 <- c(rnorm(100, 5, 2.5), rnorm(100, 5, 2), rnorm(100, 5, 3))
x2 <- c(rnorm(100, 5, 2), rnorm(100, 10, 2), rnorm(100, 15, 2))
y <- c(rep(0, 100), rep(1, 100), rep(2, 100))
Rreg <- lm(y ~ x1 + x2)
summary(Rreg)

##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8666 -0.2291 -0.0023  0.2570  0.8913
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.54830    0.06008   -9.13   <2e-16 ***
## x1          -0.02196    0.00751   -2.93   0.0037 **
## x2           0.16618    0.00437   38.06   <2e-16 ***
## ---
## Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
##
## Residual standard error: 0.338 on 297 degrees of freedom
## Multiple R-squared:  0.83,Adjusted R-squared:  0.829
## F-statistic:  727 on 2 and 297 DF,  p-value: <2e-16
```

```
plot(c(0, 10), c(0, 20), type = "n", bg = "red", xlab = "X1", ylab = "X2")
rect(0, 0, 10, 20, col = "skyblue")
points(x1[1:100], x2[1:100], col = "BLUE", pch = 20)
points(x1[101:200], x2[101:200], col = "RED", pch = 20)
points(x1[201:300], x2[201:300], col = "black", pch = 20)
x <- c(1:20000)/1000
betas <- Rreg$coefficients
y <- (0.5 - betas[1] - betas[2] * x)/betas[3]
lines(x, y, lwd = 2, col = "green")
z <- (1.5 - betas[1] - betas[2] * x)/betas[3]
lines(x, z, lwd = 2, col = "green")
```

2.2 K nearest neighbor

```
x1 <- c(rnorm(100, 5, 2.5), rnorm(100, 10, 2), rnorm(100, 15, 3))
x2 <- c(rnorm(100, 5, 2), rnorm(100, 10, 2), rnorm(100, 5, 2))
mygrid <- expand.grid(X1 = seq(0, 20, by = 0.15), X2 = seq(0, 15, by = 0.15))

NN <- function() {
    Neighbors <<- rep(0, nrow(mygrid))
    for (i in c(1:nrow(mygrid))) {
```

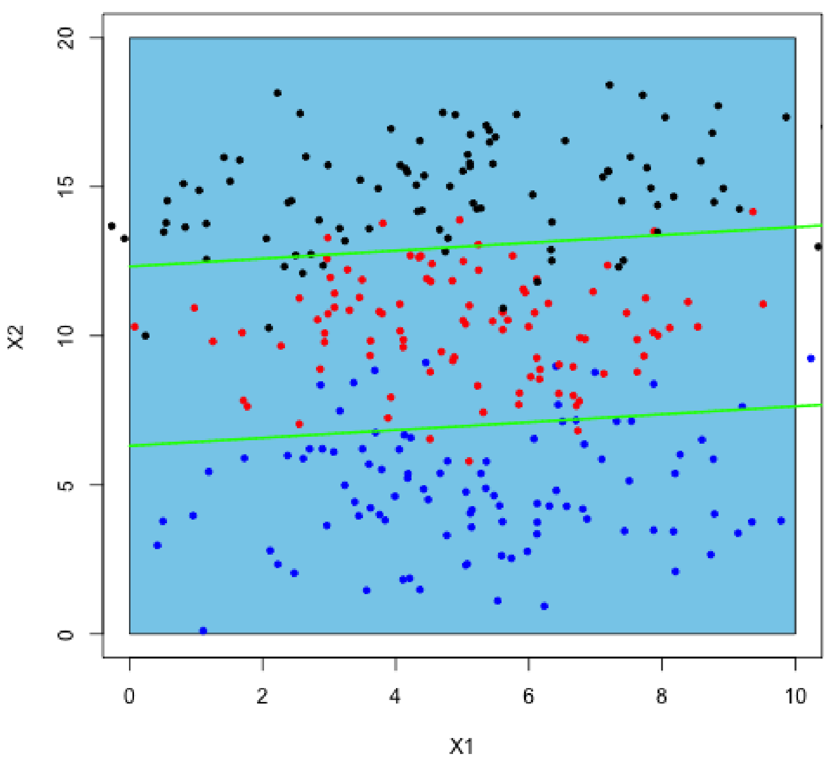

Figure 1: Classification boundaries for three classes distinguished by color using linear regression

```
        distances <- (mygrid$X1[i] - x1)^2 + (mygrid$X2[i] - x2)^2
        sort.distances <- sort.int(distances, index.return = TRUE)
        sort.indexes <- sort.distances$ix[1:15]  #knn=15
        sort.indexes[sort.indexes <= 100] = 0
        sort.indexes[sort.indexes <= 200 & sort.indexes > 100] = 1
        sort.indexes[sort.indexes <= 300 & sort.indexes > 200] = 2
        Neighbors[i] <- names(sort((table(sort.indexes)), decreasing = TRUE)[1])
    }
    Neighbors <<- Neighbors
}

myplotNN <- function() {
    plot(c(0, 20), c(0, 15), type = "n", xlab = "X1", ylab = "X2")
    points(mygrid$X1[Neighbors == 0], mygrid$X2[Neighbors == 0], col = "paleturquoise1")
    points(mygrid$X1[Neighbors == 1], mygrid$X2[Neighbors == 1], col = "rosybrown1")
    points(mygrid$X1[Neighbors == 2], mygrid$X2[Neighbors == 2], col = "palegreen")
    points(x1[201:300], x2[201:300], col = "black", pch = 20)
    points(x1[101:200], x2[101:200], col = "violetred3", pch = 20)
    points(x1[1:100], x2[1:100], col = "mediumblue", pch = 20)
}
NN()
```

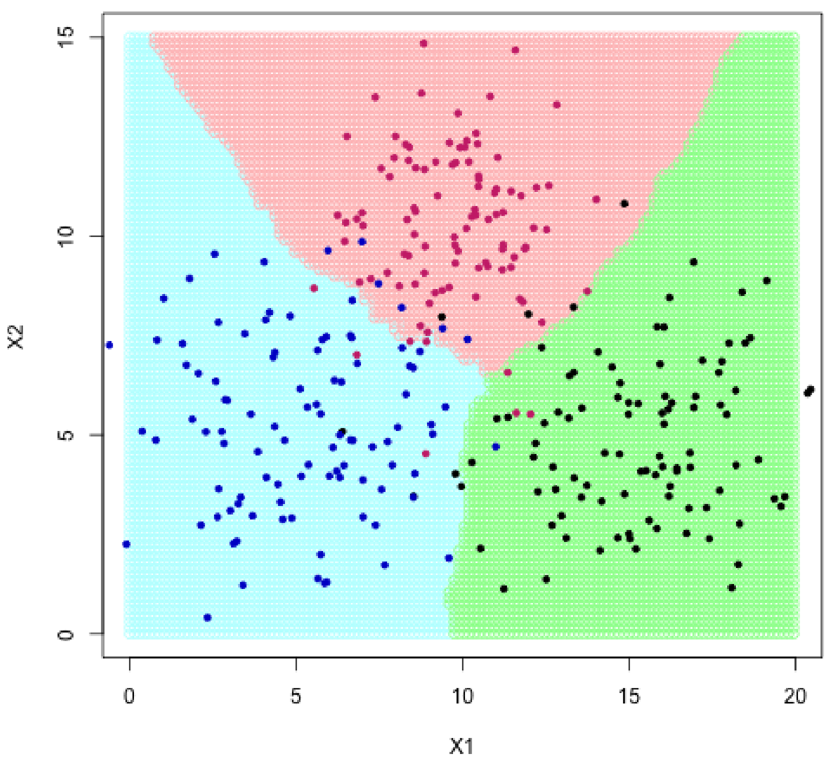

Figure 2: Classification boundaries for three classes distinguished by color using knn=15

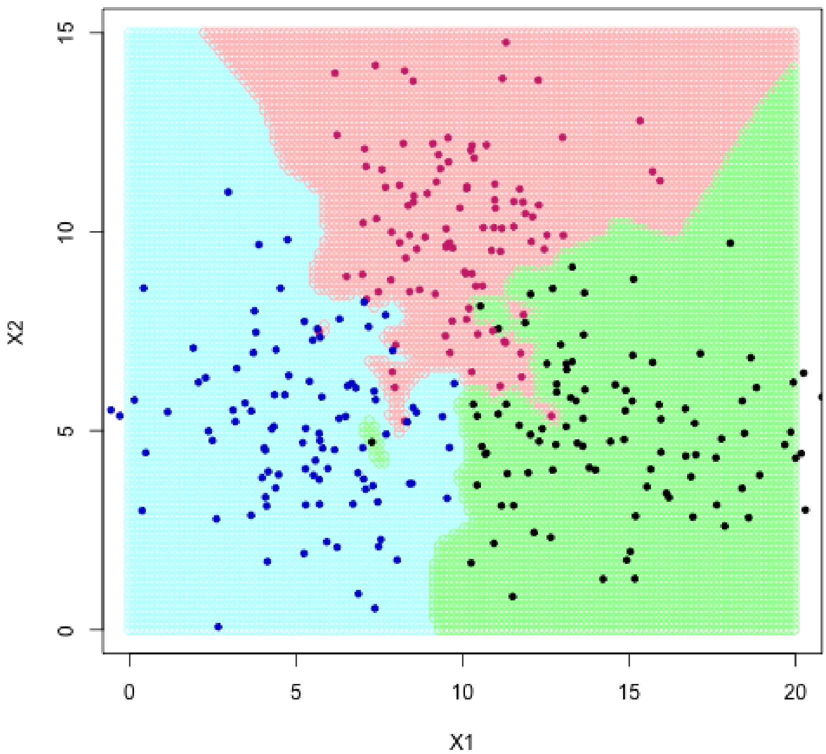

Figure 3: Classification boundaries for three classes distinguished by color using knn=1

2.3 Bayes Classifier

Bayes classifier uses the general form as below:

$$\mathbb{P}(x/0) \sim \mathbb{P}(x/1) \sim \mathbb{P}(x/2)$$

$$\begin{aligned} &\exp(\text{ -}(\mathrm{x}_1 - \mu_1)^2/2\sigma_1^2 - (x_2 - \mu_2)^2/2\sigma_2^2) \\ \sim &\exp(\text{ -}(\mathrm{y}_1 - \mu_1)^2/2\sigma_1^2 - (y_2 - \mu_2)^2/2\sigma_2^2) \\ \sim &\exp(\text{ -}(\mathrm{z}_1 - \mu_1)^2/2\sigma_1^2 - (z_2 - \mu_2)^2/2\sigma_2^2) \end{aligned}$$

In this code, I randomly generate means but precode the variance.

```r
require(MASS)

## Loading required package:  MASS

x <- mvrnorm(10, c(5, 5), matrix(c(1, 0, 0, 2), 2, 2))
y <- mvrnorm(10, c(10, 10), matrix(c(1, 0, 0, 3), 2, 2))
z <- mvrnorm(10, c(15, 15), matrix(c(2, 0, 0, 2), 2, 2))

x1 <- c()
x2 <- c()
y1 <- c()
y2 <- c()
z1 <- c()
z2 <- c()
for (i in 1:100) {
    x_rand <- rbind(x[sample(1:nrow(x), size = 2, )])
    y_rand <- rbind(y[sample(1:nrow(x), size = 2, )])
    z_rand <- rbind(z[sample(1:nrow(x), size = 2, )])
    x1[i] <- rnorm(1, x_rand[, 1], 1)
    x2[i] <- rnorm(1, x_rand[, 2], 2)
    y1[i] <- rnorm(1, y_rand[, 1], 1)
    y2[i] <- rnorm(1, y_rand[, 2], 3)
    z1[i] <- rnorm(1, z_rand[, 1], 2)
    z2[i] <- rnorm(1, z_rand[, 2], 2)
}

mygrid <- expand.grid(X1 = seq(0, 20, by = 0.15), X2 = seq(0, 20, by = 0.15))

BC <- function() {
    classifier <<- rep(0, nrow(mygrid))
    for (i in c(1:nrow(mygrid))) {
        a <- (exp(-(mygrid$X1[i] - x_rand[, 1])^2/2 - (mygrid$X2[i] - x_rand[,
            2])^2/8))
        b <- (exp(-(mygrid$X1[i] - y_rand[, 1])^2/2 - (mygrid$X2[i] - y_rand[,
```

```
            2])^2/18))
        c <- (exp(-(mygrid$X1[i] - z_rand[, 1])^2/8 - (mygrid$X2[i] - z_rand[,
            2])^2/8))
        classifier[i] <- order(c(a, b, c))[3]
    }
    classifier <<- classifier
}

myplotBC <- function() {
    plot(c(0, 20), c(0, 20), type = "n", xlab = "X1", ylab = "X2")
    points(mygrid$X1[classifier == 1], mygrid$X2[classifier == 1], col = "paleturquoise1")
    points(mygrid$X1[classifier == 2], mygrid$X2[classifier == 2], col = "rosybrown1")
    points(mygrid$X1[classifier == 3], mygrid$X2[classifier == 3], col = "palegreen")
    points(z1[1:100], z2[1:100], col = "black", pch = 20)
    points(y1[1:100], y2[1:100], col = "violetred3", pch = 20)
    points(x1[1:100], x2[1:100], col = "mediumblue", pch = 20)
}
BC()
```

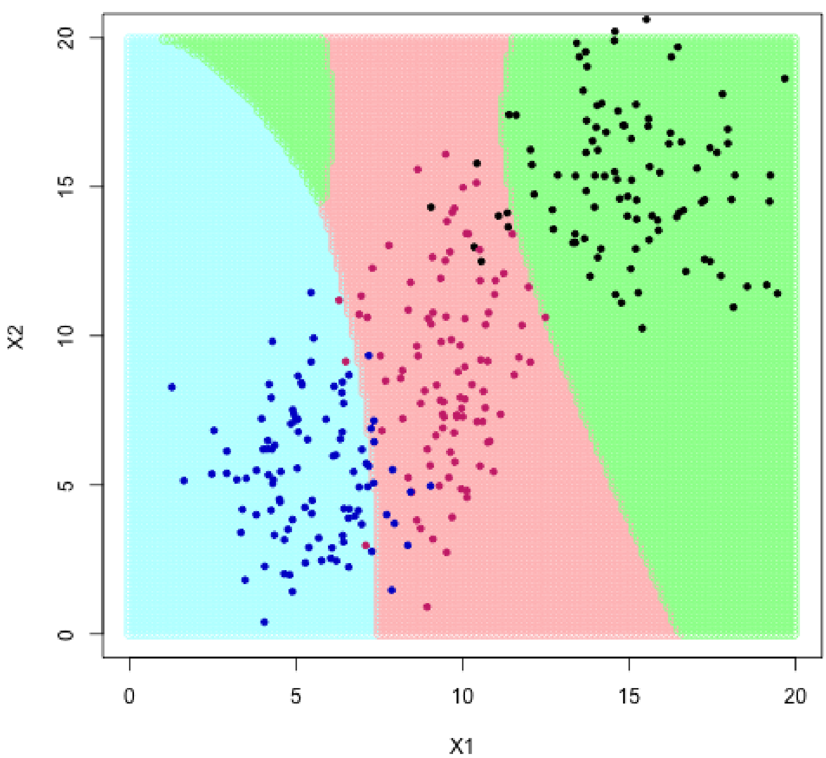

Figure 4: Classification boundaries for three classes distinguished by color using bayes classifier