621 Assignment 2

Brett Burk

Friday, February 20, 2015

1 For question 1 I created the Perceptron

```
folder <- "C:\\Users\\Brett\\Dropbox\\CUNY\\621\\Week2\\"

# Loading in the data
coin.train <- read.csv(paste0(folder, "coin-training-data.csv"),
                       stringsAsFactor = F)

#Initializing x
x <- cbind(1, coin.train[,2:4])

# Searching for 'cent'
y <- coin.train[,5]
y[y == "cent"] <- 1
y[y == "dime"] <- 0
y <- as.numeric(y)

# Learning rate
alpha <- 0.01

# Sign function, rounds to 1 if above 0.5, down to 0 otherwise
sign <- function(s){
  return(s > 0.5)
}

perceptron <- function(x, y, alpha){
  # Creating the weights
  w <- vector(length = ncol(x))
  # Initializing misclassifications as true
  misclassifications <- TRUE
  while (misclassifications) {
    # Sets to False so loop will finish when there are none
    misclassifications <- FALSE
    for (i in 1:nrow(x)) {
      # Desired result
      desired <- y[i]
      # Algorithms classification -- Simpler than other ways to find the dot product
      class <- sign(sum(x[i,] * w))
      # If they do not match:
      if (desired != class){
        misclassifications <- TRUE
        # Calculates the error
        error <- desired - class
        # Recaluclates the weight
        w <- w + (alpha * error * x[i,])
      }
    }
```

```
  }
  return(w)
}
w <- perceptron(x, y, alpha)
```

testing:

```
coin.test.pub <- read.csv(paste0(folder, "test-public.csv"),
                          stringsAsFactor = F)

#Initializing x
test.x <- cbind(1, coin.test.pub[,2:4])

# Searching for pennies--y
test.y <- coin.test.pub[,5]
test.y[test.y == "cent"] <- 1
test.y[test.y == "dime"] <- 0
test.y <- as.numeric(test.y)

predictions <- sign(as.matrix(test.x) %*% t(w))
all(test.y == predictions)
```

```
## [1] TRUE
```

Creating final Prediction:

```
coin.test.priv <- read.csv(paste0(folder, "test-private.csv"),
                           stringsAsFactor = F)
final.x <- cbind(1, coin.test.priv[,2:4])
predictions.final <- sign(as.matrix(final.x) %*% t(w))
predictions.final[predictions.final] <- "cent"
predictions.final[predictions.final == FALSE] <- "dime"
write.csv(predictions.final,
          file = paste0(folder, "perceptronpredictions.csv"))
as.vector(predictions.final)
```

```
##  [1] "cent" "dime" "dime" "dime" "dime" "cent" "cent" "dime" "cent" "dime"
## [11] "cent" "dime" "dime" "cent" "cent" "dime" "cent" "cent" "cent" "cent"
## [21] "cent" "cent" "dime" "dime" "cent" "cent" "cent" "dime" "dime" "cent"
## [31] "cent" "dime" "cent" "dime" "dime" "cent" "cent" "dime" "dime" "cent"
## [41] "cent" "dime" "dime" "dime" "dime" "dime" "dime" "dime" "cent" "cent"
```

2.a We obviously can't assume independence, as the mass is dependent on the diameter and thickness.

```
coin.likelihood.pub <- read.csv(paste0(folder, "test-public.csv"),
                                stringsAsFactor = F)
coin.likelihood.pub.x <- coin.likelihood.pub[2:4]

mvf <- function(x, mu, E){
  d <- ncol(E)
  first <- 1 / (sqrt((2 * pi) ^ d * det(E)))
```

```
  second <- -0.5 * as.matrix(x - mu) %*% solve(E) %*% t(as.matrix(x - mu))
  return(first * exp(second)[1])
}

likelihood <- function(x){
  prediction <- vector(length = nrow(x))
  cov.cent <- diag(3) * c(0.0025,0.1452,0.0009)
  mu.cent <- c(2.5, 19.05, 1.52)
  cov.dime <- diag(3) * c(0.0021, 0.1283, 0.0007)
  mu.dime <- c(2.268, 17.91, 1.35)
  for (i in 1:nrow(x)) {
    # cent
    cent <- mvf(x[i,], mu.cent, cov.cent)
    # dime
    dime <- mvf(x[i,], mu.dime, cov.dime)
    if(cent > dime){
      prediction[i] <- "cent"
    }
    else{
      prediction[i] <- "dime"
    }
  }
  return(prediction)
}
```

2.c

```
likelihood.prediction <- likelihood(coin.likelihood.pub.x)
all(likelihood.prediction == coin.likelihood.pub[,5])
```

```
## [1] TRUE
```

100% accuracy

2.d

```
coin.likelihood.priv <- read.csv(paste0(folder, "test-private.csv"),
                                 stringsAsFactor = F)
coin.likelihood.priv <- coin.likelihood.priv[2:4]
predictions.mvf.final <- likelihood(coin.likelihood.priv)
write.csv(predictions.mvf.final,
          file = paste0(folder, "mvfpredictions.csv"))
predictions.mvf.final
```

```
##  [1] "cent" "dime" "dime" "dime" "dime" "cent" "cent" "dime" "cent" "dime"
## [11] "cent" "dime" "dime" "cent" "cent" "dime" "cent" "cent" "cent" "cent"
## [21] "cent" "cent" "dime" "dime" "cent" "cent" "cent" "dime" "dime" "cent"
## [31] "cent" "dime" "cent" "dime" "dime" "cent" "cent" "dime" "dime" "cent"
## [41] "cent" "dime" "dime" "dime" "dime" "dime" "dime" "dime" "cent" "cent"
```

```
all(predictions.final == predictions.mvf.final)
```

```
## [1] TRUE
```