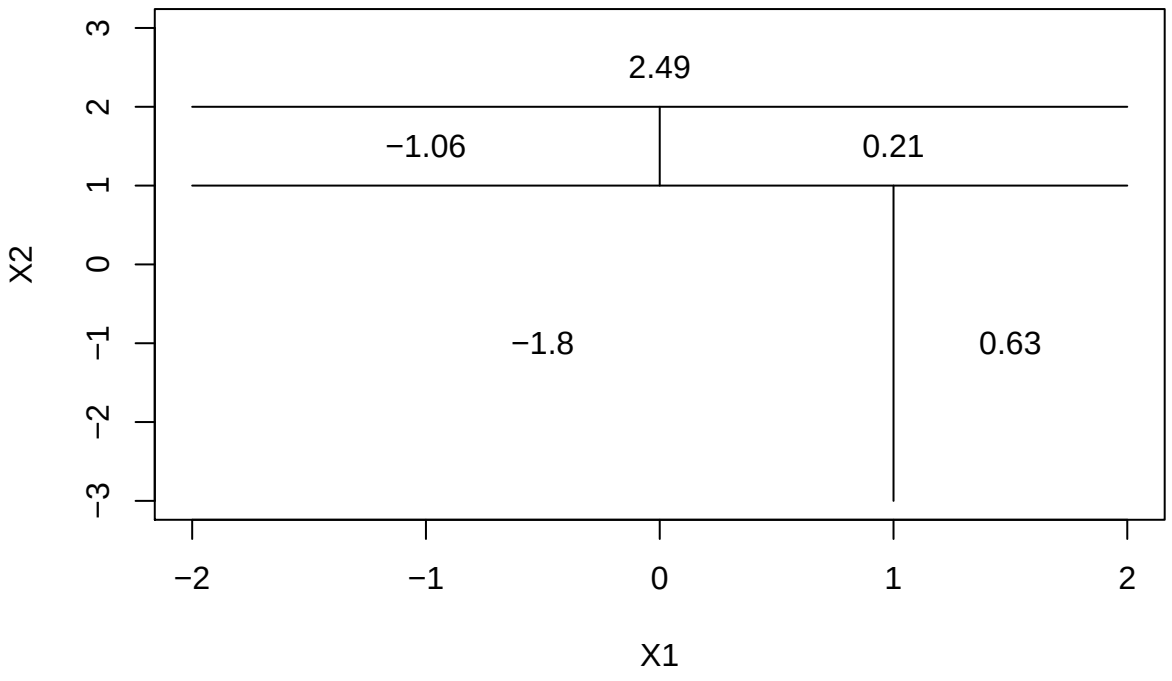
5241.hw4__cm3700

Chi Ma

4/4/2018

Q1 part B

```
par(xpd = NA)
plot(NA, NA, type = "n", xlim = c(-2, 2), ylim = c(-3, 3), xlab = "X1", ylab = "X2")

lines(x = c(-2, 2), y = c(1, 1))

lines(x = c(1, 1), y = c(-3, 1))
text(x = (-2 + 1)/2, y = -1, labels = c(-1.8))
text(x = 1.5, y = -1, labels = c(0.63))

lines(x = c(-2, 2), y = c(2, 2))
text(x = 0, y = 2.5, labels = c(2.49))

lines(x = c(0, 0), y = c(1, 2))
text(x = -1, y = 1.5, labels = c(-1.06))
text(x = 1, y = 1.5, labels = c(0.21))
```



#Q2

```r
p2 <- c(0.1,0.15,0.2,0.2,0.55,0.6,0.6,0.65,0.7,0.75)
```

Majority Approach

```r
sum(p2>=0.5)>sum(p2<0.5)
```

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

```r
#red predictions are greater than green predictions base on 50% threshold, thus RED
```

```r
mean(p2)
```

```
## [1] 0.45
```

Average Approach

```r
# Average probability is 0.45, thus Green
```

Q3

train

```r
train <- function(X,w,y){
  num_row <- dim(X)[1]
  num_col <- dim(X)[2]

  loss <- rep(0,times = num_col)
  theta <- rep(0,times = num_col)
  mode <- rep(0,times = num_col)

  for(j in 1:num_col){

    index <- order(X[,j])
    X_j <- X[index, j]
    X_j2 <-  X[rev(index),j]

    w_cum <- cumsum(w[index]*y[index])
    w_cum[rev(duplicated(X_j2) == 1)] <- NA

    max_val <- max(abs(w_cum), na.rm = TRUE)
    max_index <- min(which(abs(w_cum)== max_val))
    mode[j] <- (w_cum[max_index] < 0)*2-1
    theta[j] <- X_j[max_index]
    c <- ((X[,j] > theta[j])*2 - 1)*mode[j]
    loss[j] <- w %*% (c != y)
  }
    max_val <- min(loss)
```

```
    j_star <- min(which(loss == max_val))

    pars <- list(j = j_star, theta = theta[j_star], mode = mode[j_star])
    return(pars)

}
```

classify

```
classify <- function(X, pars){
 label <- (2*(X[,pars$j]>pars$theta)-1)*pars$mode
 return(label)
}
```

agg_class

```
agg_class <- function(X,alpha,allPars){
  n = dim(X)[1]
  B = length(alpha)
  labels = matrix(0, nrow = n,ncol = B)

  for(b in 1:B){
    labels[,b] <- classify(X, allPars[[b]])
  }
  labels <- labels %*% alpha
  c_hat <- sign(labels)
  return(c_hat)
}
```

adaboost

```
adaBoost <- function(X, y, B){

   n <- nrow(X)
   w <- rep(1/n, times = n)
   alpha <- rep(0,times = B)
   allPars <- rep(list(list()), B)

   for(b in 1:B){
     allPars[[b]] <- train(X,w,y)

     missClass <- (y != classify(X,allPars[[b]]))
     error <- (w %*% missClass/sum(w))[1]

     alpha[b] <- log((1-error)/error)
     w <- w *exp(alpha[b]*missClass)
```

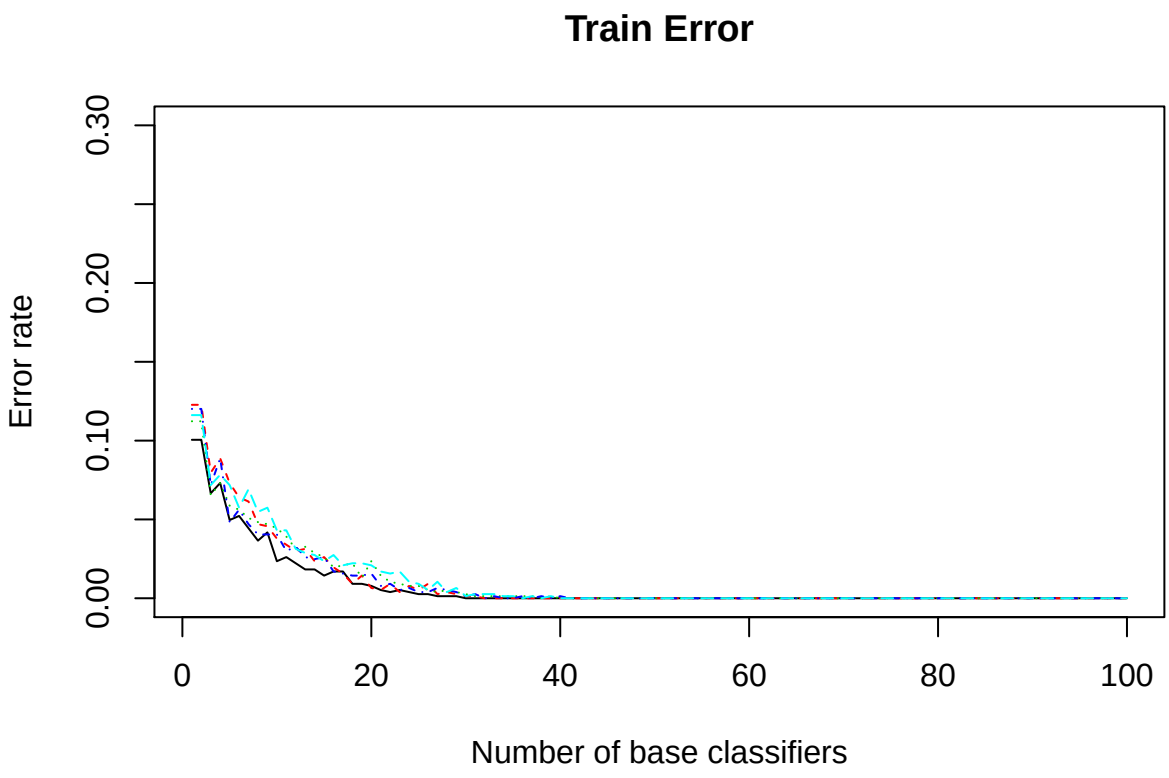
```
  }
  return(list(allPars = allPars, alpha = alpha))
}
```

Run

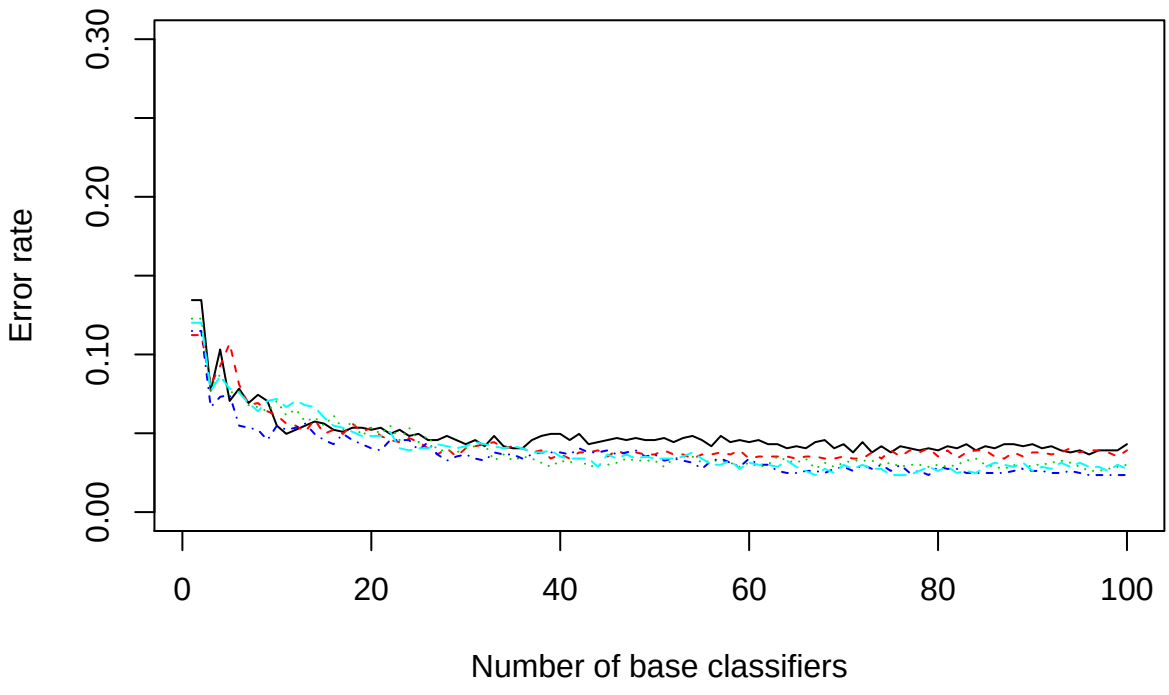
```
df3 = read.table("train_3.txt",header = F,sep = ",")
df8 = read.table("train_8.txt",header = F,sep = ",")
df3 = as.matrix(df3)
df8 = as.matrix(df8)
y_train = as.matrix(rep(c(-1,1),c(658,542)))

X_train = rbind(df3,df8)

dft = read.table("zip_test.txt")
dft2 = dft[dft[1]==3|dft[1]==8,]
y_test = as.matrix(dft2[,1])
X_test = as.matrix(dft2[,2:257])
y_test[y_test == 3] <- -1
y_test[y_test == 8] <- 1

set.seed(0)
X_final <- rbind(X_train,X_test)
y_final <- rbind(y_train,y_test)

B_max = 100
nCV = 5
n <- nrow(X_final)

train_error = matrix(0,nrow = B_max, ncol = nCV)
test_error = matrix(0,nrow = B_max,ncol = nCV)
```

```
for(i in 1:nCV){

  p <-sample.int(n)
  train_index <- p[1:round(n/2)]
  test_index <- p[-(1:round(n/2))]

  ada <- adaBoost(X_final[train_index,],y_final[train_index],B_max)
  allPars <- ada$allPars
  alpha <- ada$alpha
  for(B in 1:B_max){
    c_hat_train = agg_class(X_final[train_index,],alpha[1:B],allPars[1:B])
    c_hat_test = agg_class(X_final[test_index,],alpha[1:B],allPars[1:B])
    train_error[B,i] = mean(y_final[train_index] != c_hat_train)
    test_error[B,i] = mean(y_final[test_index] != c_hat_test)
  }
}
```

```
matplot(train_error,type = "l",lty = 1:nCV,main = "Train Error",xlab = "Number of base classifiers",ylab
```




```
matplot(test_error,type = "l",lty = 1:nCV,main = "Test Error",xlab = "Number of base classifiers",ylab =
```




```
# When B close to 30, seems has no effect on train set since all points are classified correctly. When
```