HW 4

Carlyle Morgan

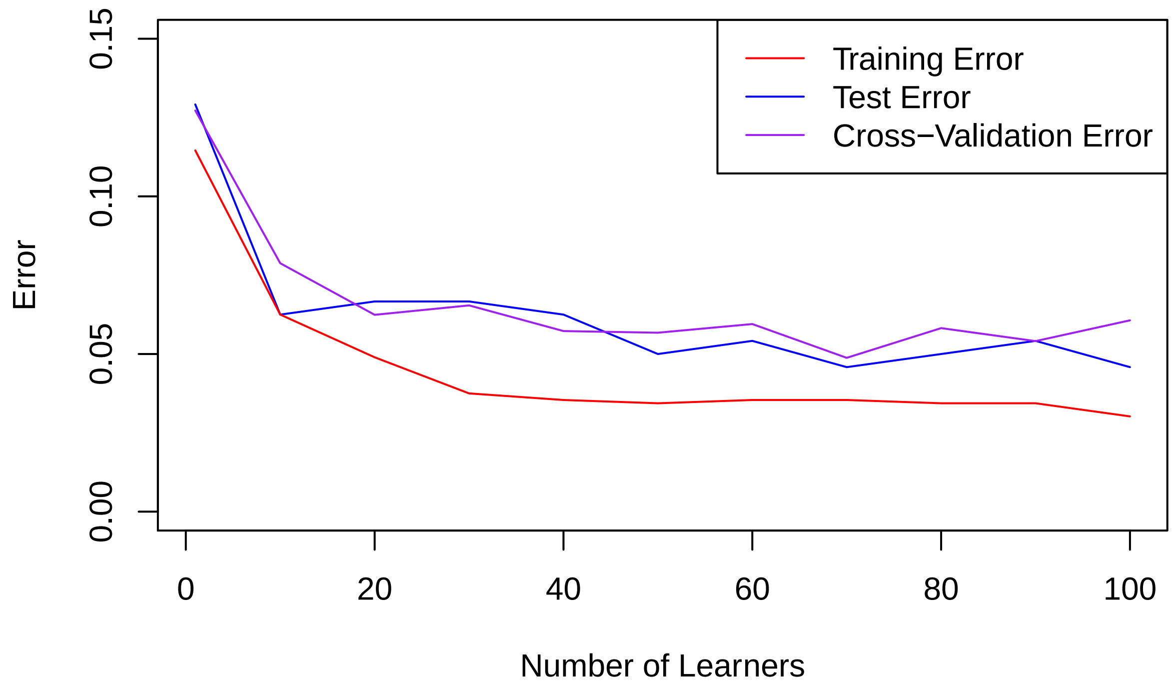
5/9/2022

Problem 1

```
train_3<-read.csv('train_3.txt', header = FALSE)
train_8<-read.csv('train_8.txt', header = FALSE)
train_3$digit<--1
train_8$digit<-1

X<-rbind(train_3,train_8)
testind<-sample(seq(1,nrow(X)),nrow(X)*.2)
test<-X[testind,]
X<-X[-testind,]
y<-X$digit

adaclassify<-function(case, j_stars=j_stars, theta_stars=theta_stars, alphas=alphas){
  class_sum<-0
  for(g in 1:length(j_stars)){
    class_sum<-class_sum+(alphas[g]*ifelse(case[j_stars[g]]>theta_stars[g],1,-1))
  }
  ifelse(class_sum>0,1,-1)
}


theta_err<-function(X,theta,index,w,y){
  c<-rep(0,length(X[,index]))
  c<-ifelse(X[,index]>theta,1,-1)
  errs<-w*ifelse(y==c,0,1)
  sum(errs)
}

ada_errors<-function(ada,test){
  y_pred_train<-apply(X, 1, function(x) adaclassify(x, ada$j_stars, ada$theta_stars, ada$alphas))
  y_pred_test<-apply(test, 1, function(x) adaclassify(x, ada$j_stars, ada$theta_stars, ada$alphas))
  train_error<-mean(ifelse(y_pred_train==X$digit,0,1))
  test_error<-mean(ifelse(y_pred_test==test$digit,0,1))
  c(train_error,test_error)

}

adaboost_digits<-function(X, B){
  y<-X$digit
  weights<-rep(1/nrow(X), nrow(X))
```

```r
  j_stars<-rep(0,B)
  theta_stars<-rep(0,B)
  alphas<-rep(0,B)

  for (b in 1:B){

    j_star<-which.min(sapply(seq(1,256),function(j) min(sapply(seq(-1,1,.1), function(x) theta_err(X,x,
    theta_star<-seq(-1,1,.1)[which.min(sapply(seq(-1,1,.1), function(x) theta_err(X,x,j_star,weights,y)

    c<-ifelse(X[,j_star]>theta_star,1,-1)

    j_stars[b]<-j_star
    theta_stars[b]<-theta_star

    epsilon<-sum(weights*ifelse(y==c,0,1))/sum(weights)
    alphas[b]<-log((1-epsilon)/epsilon)
    weights<-weights*exp(alphas[b]*ifelse(y==c,0,1))
  }
  list(j_stars=j_stars,theta_stars=theta_stars,alphas=alphas)
}

crossvalerror<-function(X, B, k){
  X$fold<-ceiling(runif(nrow(X))*k)
  folderrors<-c()
  for (i in 1:k){
    curfold<-X[X$fold==i,]
    temptrain<-X[X$fold!=i,]
    tempada<-adaboost_digits(temptrain,B)
    folderrors<-append(folderrors, ada_errors(tempada, curfold)[2])
  }
  mean(folderrors)
}


train_errors<-c()
test_errors<-c()

## These had to be manually calculated using the above crossvalerror() function as they took several mi
crossval_errors<-c(0.12724,0.07879108, 0.06243371, 0.06541612, 0.05727944,
                   0.05675727,0.05950183, 0.04878585, 0.05821127, 0.05409029, 0.06067585)

bs<-c(1,seq(10,100,10))

for (i in c(1,seq(10,100,10))){
  temp<-ada_errors(adaboost_digits(X,i),test)
  train_errors<-append(train_errors,temp[1])
  test_errors<-append(test_errors,temp[2])
}



plot(bs, test_errors, type = "l", col = "blue", xlab = "Number of Learners", ylab = "Error", ylim = c(0
lines(bs, train_errors, col = "red")
```

```r
lines(bs, crossval_errors, col = "purple")
legend("topright", legend = c("Training Error", "Test Error", "Cross-Validation Error"), lty = c(1,1,1)
```



From this plot, it appears that ~70 learners would be an appropriate choice, as it minimizes cross-validation error.

Problem 2

i) For $Q(\beta) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda|\beta|$

Case 1: $\beta > 0$

$$\Rightarrow Q(\beta) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda\beta$$

$$\frac{\partial Q(\beta)}{\partial\beta} = \frac{1}{2n}\sum_{i=1}^{n} -2x_i(y_i - x_i\beta) + \lambda$$

First Deriv test $\Rightarrow$ $0 = \frac{1}{2n}\sum_{i=1}^{n} -2x_i(y_i - x_i\beta) + \lambda$

$$= -\frac{1}{n}\sum_{i=1}^{n} x_i y_i + \beta\sum_{i=1}^{n} x_i^2 + \lambda$$

$\sum_{i=1}^{n} x_i^2 = 1 \Rightarrow 0 = -\frac{1}{n}\sum x_i y_i + \beta + \lambda$

$$\beta = \frac{1}{n}\sum x_i y_i - \lambda$$

$\beta > 0 \Rightarrow \frac{1}{n}\sum x_i y_i - \lambda > 0$

$\Leftrightarrow \frac{1}{n}\sum x_i y_i > \lambda$ ✓

Case 2: $\beta < 0$

$$\Rightarrow Q(\beta) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i\beta)^2 - \lambda\beta$$

$$\frac{\partial Q(\beta)}{\partial\beta} = \frac{1}{2n}\sum_{i=1}^{n} -2x_i(y_i - x_i\beta) - \lambda$$

First Deriv test $\Rightarrow$ $0 = \frac{1}{2n} \sum_{i=1}^{n} -2x_i(y_i - x_i\beta) - \lambda$

$$= -\frac{1}{n} \sum_{i=1}^{n} x_i y_i + \beta \sum_{i=1}^{n} x_i^2 - \lambda$$

$\sum_{i=1}^{n} x_i^2 = 1 \Rightarrow 0 = -\frac{1}{n} \sum x_i y_i + \beta - \lambda$

$\beta = \frac{1}{n} \sum x_i y_i + \lambda$

$\beta > 0 \Rightarrow \frac{1}{n} \sum x_i y_i + \lambda < 0$

$\Leftrightarrow \frac{1}{n} \sum x_i y_i < -\lambda$ ✓

Case 3: $\beta = 0$

$$Q(\beta) = \frac{1}{2n} \sum_{i=1}^{n} y_i^2$$

$$\frac{\partial Q(\beta)}{\partial \beta} = 0 \checkmark$$

ii)

For $Q(\beta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j|$

Case 1: $\beta_j > 0$

$$Q(\beta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \sum_{h=1}^{p} \beta_h x_{ih})^2 + \lambda \sum_{h=1}^{p} \beta_h$$

$$\frac{\partial Q(\beta)}{\partial \beta_j} = \frac{1}{2n} \sum_{i=1}^{n} -2 x_{ij} (y_i - \sum_{h=1}^{p} \beta_h x_{ih})^2 + \lambda$$

First Deriv Test $0 = \frac{1}{2n} \sum_{i=1}^{n} -2 x_{ij} (y_i - \sum_{h \neq j} \beta_h x_{ih} - \beta_j x_{ij}) + \lambda$

$$= \frac{1}{n} \sum_{i=1}^{n} - x_{ij} \, r_i^{(j)} + \beta_j \sum x_{ij}^2 + \lambda$$

$\sum x_{ij}^2 = 1 \Rightarrow -\frac{1}{n} x_j^T r_i^{(j)} + \beta_j + \lambda = 0$

$\Leftrightarrow \quad \beta_j = \frac{1}{n} x_j^T - \lambda$

$\beta_j > 0 \Rightarrow \frac{1}{n} x_j^T - \lambda > 0$

$\Leftrightarrow \frac{1}{n} x_j^T > \lambda$ ✓

Case 2: $B_j > 0$

$$Q(B) = \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \sum_{h=1}^{p} B_h x_{ih}\right)^2 - \lambda \sum_{h=1}^{p} B_h$$

$$\frac{\partial Q(B)}{\partial B_j} = \frac{1}{2n} \sum_{i=1}^{n} -2 x_{ij} \left(y_i - \sum_{h=1}^{p} B_h x_{ih}\right)^2 - \lambda$$

First Deriv Test $\quad 0 = \frac{1}{2n} \sum_{i=1}^{n} -2 x_{ij} \left(y_i - \sum_{h \neq j} B_h x_{ih} - B_j x_{ij}\right) - \lambda$

$$= \frac{1}{n} \sum_{i=1}^{n} -x_{ij} r_i^{(j)} + B_j \sum x_{ij}^2 - \lambda$$

$$\sum x_{ij}^2 = 1 \Rightarrow -\frac{1}{n} x_j^T r_i^{(j)} + B_j - \lambda = 0$$

$$\Leftrightarrow \quad B_j = \frac{1}{n} x_j^T + \lambda$$

$$B_j > 0 \Rightarrow \frac{1}{n} x_j^T + \lambda < 0$$

$$\Leftrightarrow \quad \frac{1}{n} x_j^T < -\lambda \quad \checkmark$$

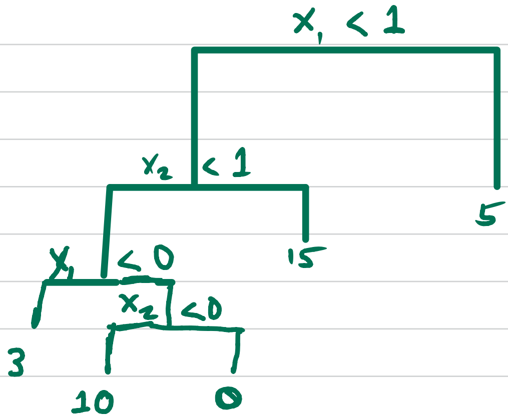
Case 3: $B_j = 0$

$$Q(B) = \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \sum_{h \neq j} B_h x_{ih}\right)^2 - \lambda \sum_{h \neq j} B_h$$

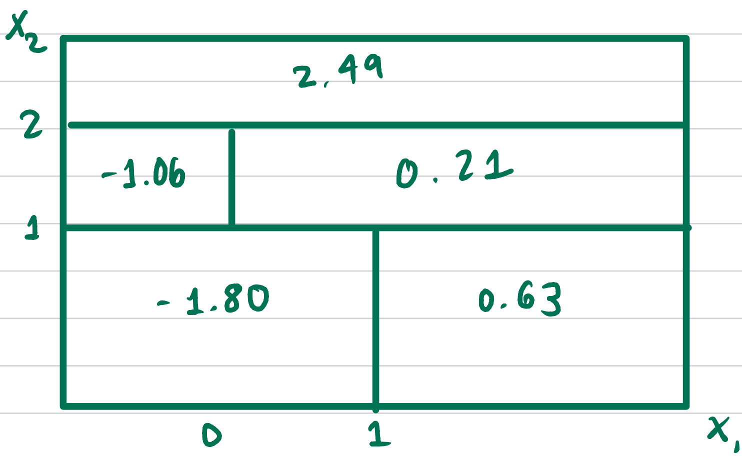
$$\frac{\partial Q(B)}{\partial Q B_j} = 0 \quad \checkmark \qquad \square$$

Problem 3

i)



ii)



Problem 4

1. As $\lambda \to \infty$,

$$\lambda \int [g^{(3)}(x)]^2 \to 0 \quad \text{and} \quad \lambda \int [g^{(4)}(x)]^2 dx \to 0$$

for the expressions for $\hat{g}_1$ and $\hat{g}_2$ to be minimized.

Thus, as $\lambda \to 0$:

$$\hat{g}_1 \to \arg\min_g \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 \right)$$

$$\hat{g}_2 \to \arg\min_g \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 \right)$$

which are least squares estimators.

Thus $\hat{g}_1$ and $\hat{g}_2$ will have similar training RSS.

2. By identical logic to above, $\hat{g}_1$ and $\hat{g}_2$ will have similar training errors.

3. Since there exists zero penalty for how rough $\hat{g}_1$ and $\hat{g}_2$ are, they will be identically fit to the training set resulting in zero training error. Test RSS will be determined by how overfit $\hat{g}_1$ and $\hat{g}_2$ are from the training data, so which of $\hat{g}_1$ and $\hat{g}_2$ will have greater test error is indeterminate.