ComBio HomeWork3

Jiaqi Ma

LASSO

Initialization:

```
library(glmnet)
```

```
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-2
```

```
library(lars)
```

```
## Loaded lars 1.2
```

```
dataset<-read.table("prostate.txt")
rawdata_x <- as.matrix(dataset[,1:8])
rawdata_y <- as.matrix(dataset[,9])
n=length(dataset[,1])
```

Results of glment:

```
res_g <- cv.glmnet(rawdata_x, rawdata_y, nfolds=10)
para <- coef(res_g$glmnet.fit, s=res_g$lambda.1se)
print(para)
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) 0.7820637
## lcavol      0.4485189
## lweight     0.2804033
## age         .
## lbph        .
## svi         0.3383490
## lcp         .
## gleason     .
## pgg45       .
```

My Implementation:

```
# standardization
X <- scale(rawdata_x)
x_scale <- attr(X,"scaled:scale")
Y <- rawdata_y-mean(rawdata_y)
```

```r
# lasso optimization
lambda <- res_g$lambda.1se

A <- t(X)%*%X
B <- t(X)%*%Y
for (i in 1:8)
  {
    A[i,i]=A[i,i]+lambda
  }
w=solve(A)%*%B

for (step in 1:100)
  {
    w0 <- w
    for (i in 1:8)
      {
        c <- t(X[,i])%*%(Y-X%*%w+w[i]*X[,i])/n
        w[i] <- sign(c)*max((abs(c)-lambda),0)
      }
    if(max(abs(w-w0))<1e-8)
      {
        break
      }
  }
w=w/x_scale
print(w)
```

```
##              [,1]
## lcavol  0.4424871
## lweight 0.2791078
## age     0.0000000
## lbph    0.0000000
## svi     0.3416167
## lcp     0.0000000
## gleason 0.0000000
## pgg45   0.0000000
```