# HW4

Xinyue Lu

2020/2/17

## 1. Compare the results of analysis of prostate cancer dataset. Use 3 models: best subset selection, ridge regression and lasso. First, compare the set of important variables between lasso and best subset selection. Use 10 fold cross validation. Report cross validated MSE for each method. Compare.

Summary: The best subset selection methods choosed 5 variables: cavol, lweight, age, lbph, svi. The mean 10-fold cross validation error was 0.5275879. The Lasso model also chooed 5 variables: lcavol, lweight, lbph, svi, pgg45, which were different than the best subset method choosed. The mean cross validated error was 0.5519, which was larger than that of the best subset method.

```
library('lasso2')
```

```
## R Package to solve regression problems while imposing
##   an L1 constraint on the parameters. Based on S-plus Release 2.1
## Copyright (C) 1998, 1999
## Justin Lokhorst   <jlokhors@stats.adelaide.edu.au>
## Berwin A. Turlach <bturlach@stats.adelaide.edu.au>
## Bill Venables     <wvenable@stats.adelaide.edu.au>
##
## Copyright (C) 2002
## Martin Maechler <maechler@stat.math.ethz.ch>
```

```
data(Prostate)
sum(is.na(Prostate))
```

```
## [1] 0
```

```
Prostate.s = Prostate
for (i in 1:8){Prostate.s[i] = scale(Prostate[i])}
```

### 1.1 Best Subset Selection

The model chosen with smallest cross validation error had 5 variables: cavol, lweight, age, lbph, svi. The mean 10-fold cross validation error was 0.5275879.

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 3.6.2
```

```
prostate.leaps <- regsubsets( lpsa ~ . ,method="exhaustive", data=Prostate.s, nbest = 70,really.big = TRUE )
prostate.leaps.sum = summary(prostate.leaps)
prostate.models <- prostate.leaps.sum$which
prostate.models.adjr2 <- prostate.leaps.sum$adjr2

index.best.adjr2 = which( prostate.models.adjr2 == max(prostate.leaps.sum$adjr2))
prostate.models[index.best.adjr2,]
```

```
## (Intercept)      lcavol     lweight         age        lbph         svi
##        TRUE        TRUE        TRUE        TRUE        TRUE        TRUE
##         lcp     gleason       pgg45
##        TRUE       FALSE        TRUE
```

```
prostate.models.size <- as.numeric(attr(prostate.models, "dimnames")[[1]])
prostate.models.rss <- prostate.leaps.sum$rss
prostate.models.best.rss <- tapply( prostate.models.rss, prostate.models.size, min )
prostate.models.best.rss
```

```
##        1        2        3        4        5        6        7        8
## 58.91478 52.96636 47.78496 46.48490 45.52565 44.86669 44.20436 44.16313
```

```
prostate.dummy <- lm( lpsa ~ 1, data=Prostate.s ) # only intercept model
prostate.models.best.rss <- c(sum(resid(prostate.dummy)^2),prostate.models.best.rss)

cat('The best model with 4 predictors:\n\n')
```

```
## The best model with 4 predictors:
```

```
index.best4 = which( prostate.models.rss == prostate.models.best.rss[5])
prostate.models[index.best4,]
```

```
## (Intercept)      lcavol     lweight         age        lbph         svi
##        TRUE        TRUE        TRUE       FALSE        TRUE        TRUE
##         lcp     gleason       pgg45
##       FALSE       FALSE       FALSE
```

```
cat('The best model with 5 predictors:\n\n')
```

```
## The best model with 5 predictors:
```

```
index.best5 = which( prostate.models.rss == prostate.models.best.rss[6])
prostate.models[index.best5,]
```

```
## (Intercept)      lcavol     lweight         age        lbph         svi
##        TRUE        TRUE        TRUE        TRUE        TRUE        TRUE
##         lcp     gleason       pgg45
##       FALSE       FALSE       FALSE
```

```
cat('The best model with 6 predictors:\n\n')
```

```
## The best model with 6 predictors:
```

```
index.best6 = which( prostate.models.rss == prostate.models.best.rss[7])
prostate.models[index.best6,]
```

```
## (Intercept)      lcavol     lweight         age        lbph         svi
##        TRUE        TRUE        TRUE        TRUE        TRUE        TRUE
##         lcp     gleason       pgg45
##       FALSE       FALSE        TRUE
```

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.6.2
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 3.0-2
```

```
y=Prostate.s$lpsa
grid = 10 ^ seq(5, -2, length=100)

X=model.matrix(lpsa~., data=Prostate.s)
l.cv = cv.glmnet(X,y,alpha=0,lambda=c(0,0.1),thresh=1e-12,nfolds = 10, seed=123)
l.cv$cvm[1]
```

```
## [1] 0.5480749
```

```
X=model.matrix(lpsa~lcavol+lweight+age+lbph+svi+lcp+pgg45, data=Prostate.s)
l.cv = cv.glmnet(X,y,alpha=0,lambda=c(0,0.1),thresh=1e-12,nfolds = 10, seed=123)
l.cv$cvm[1]
```

```
## [1] 0.5488294
```

```
X=model.matrix(lpsa~lcavol+lweight+age+lbph+svi+pgg45, data=Prostate.s)
l.cv = cv.glmnet(X,y,alpha=0,lambda=c(0,0.1),thresh=1e-12,nfolds = 10, seed=123)
l.cv$cvm[1]
```

```
## [1] 0.5514267
```

```
X=model.matrix(lpsa~lcavol+lweight+age+lbph+svi, data=Prostate.s)
l.cv = cv.glmnet(X,y,alpha=0,lambda=c(0,0.1),thresh=1e-12,nfolds = 10, seed=123)
l.cv$cvm[1]
```

```
## [1] 0.5821415
```

```
predict(l.cv,s=0,type="coefficients")[,1]
```

```
## (Intercept) (Intercept)       lcavol      lweight          age         lbph
##   2.4783869   0.0000000   0.6666405   0.2104160  -0.1108751   0.1622583
##         svi
##   0.2984716
```

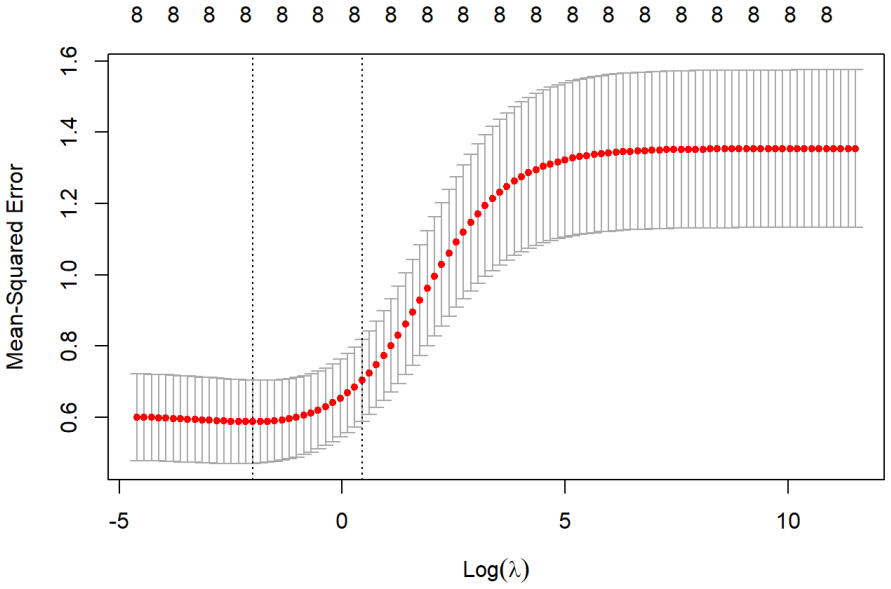
```
X=model.matrix(lpsa~lcavol+lweight+lbph+svi, data=Prostate.s)
l.cv = cv.glmnet(X,y,alpha=0,lambda=c(0,0.1),thresh=1e-12,nfolds = 10, seed=123)
l.cv$cvm[1]
```

```
## [1] 0.5316039
```

## 1.2 Ridge Regression

The best lambda was 0.0599. The cross validated MSE was 0.5514.

```
X=model.matrix(lpsa~., data=Prostate.s)
ridge.cv = cv.glmnet(X,y,alpha=0,lambda=grid,thresh=1e-12,nfolds = 10, seed=123)
plot(ridge.cv)
```




```
min(ridge.cv$cvm)
```

```
## [1] 0.5876315
```

```
lambda.best=ridge.cv$lambda.min
lambda.best
```

```
## [1] 0.1353048
```

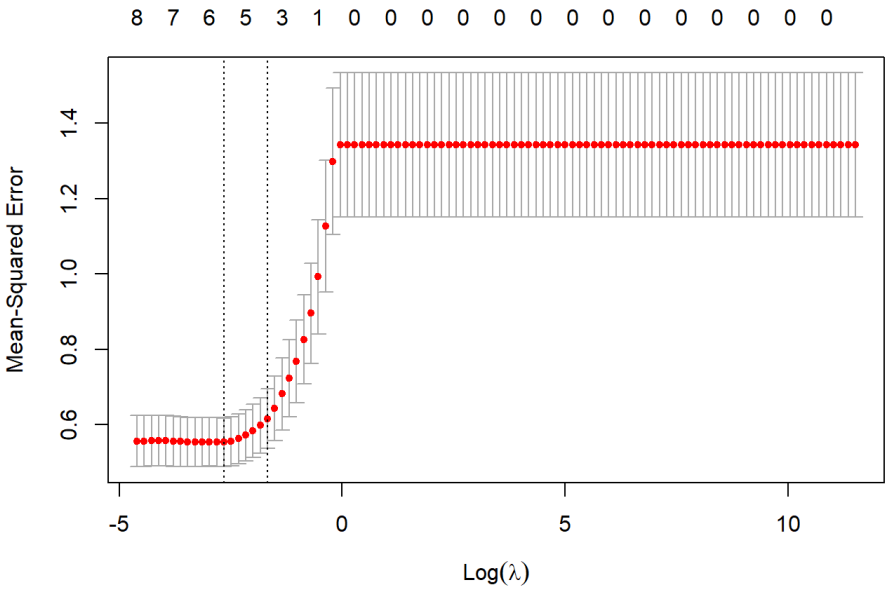
```
predict(ridge.cv,s=lambda.best,type="coefficients")
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept)  2.47838688
## (Intercept)  .
## lcavol       0.56386793
## lweight      0.21523482
## age         -0.09857621
## lbph         0.13029436
## svi          0.27291445
## lcp         -0.01774472
## gleason      0.04805068
## pgg45        0.08874225
```

## 1.3 lasso Regression

5 variables were choosen in the Lasso model. They were lcavol, lweight, lbph, svi, pgg45. The best lambda was 0.01. The cross validated MSE was 0.5519.

```
lasso.cv = cv.glmnet(X,y,alpha=1,lambda=grid,thresh=1e-12,nfolds = 10,seed=123)
plot(lasso.cv)
```




```
min(lasso.cv$cvm)
```

```
## [1] 0.5532694
```

```
lambda.best=lasso.cv$lambda.min
lambda.best
```

```
## [1] 0.07054802
```

```
lasso.coef=predict(lasso.cv,s=lambda.best,type="coefficients")[,1]
lasso.coef[lasso.coef!=0]
```

```
## (Intercept)      lcavol     lweight        lbph         svi       pgg45
##  2.47838688  0.60425472  0.16577408  0.06468257  0.22776662  0.03679961
```

## 2. Following in class discussion Build LASSO model for the dataset LiNK using two packages glmnet and biglasso. Use 20 fold cross validation. Report the results. Compare the performance

```
data<-readRDS('D:/luxinyve/OO Linear graph/HW4/bcTCGA.rds')
summary(data)
```

```
##       Length  Class      Mode
## X     9284592 -none-     numeric
## y         536 -none-     numeric
## fData       2 data.frame list
```

```
dim(data)
```

```
## NULL
```

### 2.1 fit with biglasso

The biglasso method selected 96 variables. The best lambda was 0.04233. The mean 20-fold cross validation error was 0.19977.

```
library(biglasso)
```

```
## Warning: package 'biglasso' was built under R version 3.6.2
```

```
## Loading required package: bigmemory
```

```
## Warning: package 'bigmemory' was built under R version 3.6.2
```

```
## Loading required package: ncvreg
```

```
## Warning: package 'ncvreg' was built under R version 3.6.2
```

```
##
## Attaching package: 'ncvreg'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     Prostate
```

```
X.bm <- as.big.matrix(data$X)
dim(X.bm)
```

```
## [1]   536 17322
```

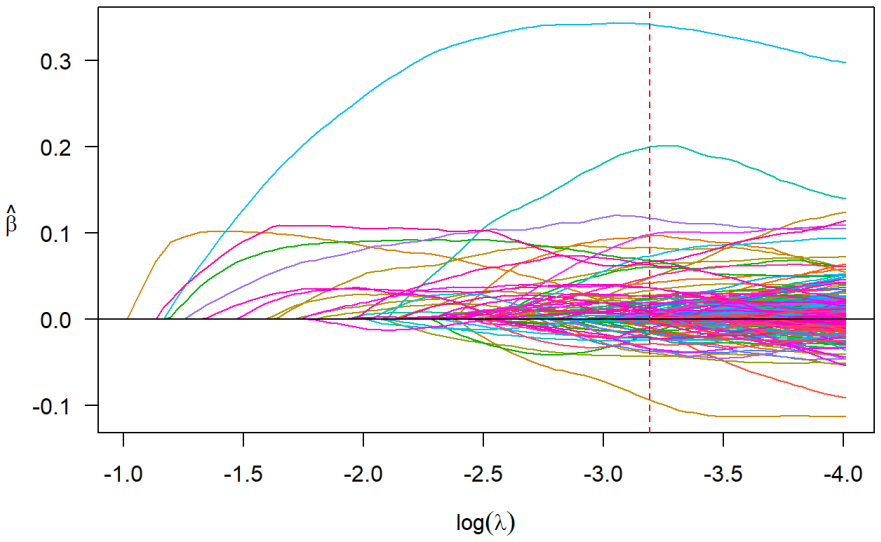
```
cvfit <- cv.biglasso(X.bm, data$y, family = "gaussian", seed = 1234,nfolds = 20, ncores = 4)
cvfit$lambda.min
```

```
## [1] 0.04106738
```

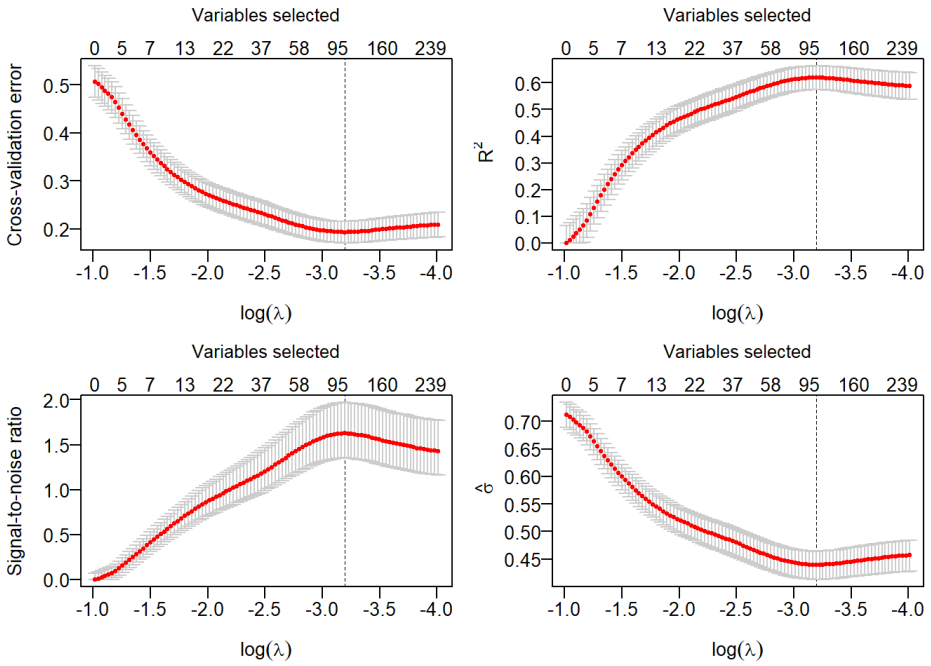
```
min(cvfit$cve)
```

```
## [1] 0.1929226
```

```
plot(cvfit$fit)
abline(v = log(cvfit$lambda.min), col = 2, lty = 2)
```




```
par(mfrow = c(2, 2), mar = c(3.5, 3.5, 3, 1), mgp = c(2.5, 0.5, 0))
plot(cvfit, type = "all")
```




```
coefs <- as.matrix(coef(cvfit))
length(coefs[coefs != 0, ])
```

```
## [1] 104
```

## 2.2 fit with glmnet

The lasso method selected 51 variables. The best lambda was 0.04329. The mean 20-fold cross validation error was 0.1983196. The general lasso selected much fewer variables than the biglasso method and had smaller mean cross validation error. Its operation speed was slower than the biglasso's, but it was acceptable.

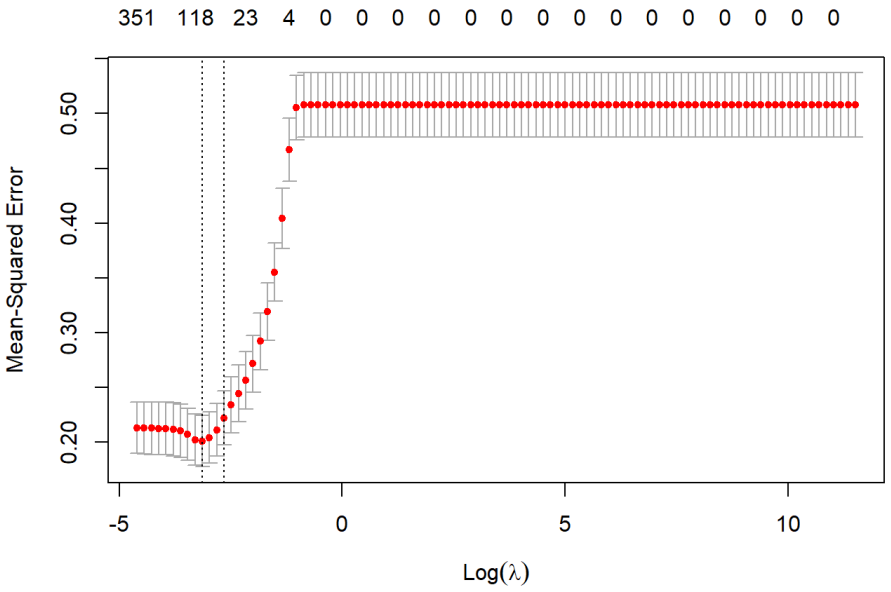
```
X.m <- as.matrix(data$X)
dim(X.m)
```

```
## [1]   536 17322
```

```
one = rep(1, times=536)
X.m.1=cbind(one,X.m)
dim(X.m.1)
```

```
## [1]   536 17323
```

```
lasso.cv = cv.glmnet(X.m.1, data$y,alpha=1,lambda=grid,thresh=1e-12,seed = 1234,nfolds = 20)
plot(lasso.cv)
```




```
min(lasso.cv$cvm)
```

```
## [1] 0.2010968
```

```
lambda.best=lasso.cv$lambda.min
lambda.best
```

```
## [1] 0.04328761
```

```
coefs <- as.matrix(coef(lasso.cv))
length(coefs[coefs != 0, ])
```

```
## [1] 51
```