Lab 02

[Sicily Xie 54385315]

2022-10-05

Brief description

In the lectures last week, you learned how to select features (predictors) using stepwise regression based on the Akaike information criterion (AIC), or to minimize the mean squared prediction error. You were also introduced to the concept of ridge regression, one of the many shrinkage (regularization) methods you will learn in this course.

In this lab, we will perform ridge regression on the prostate cancer data set studied last time. Recall that the 9th variable, `lpsa`, is the response while the rest are potential predictors. the variable `lpsa` in the 9th column is the response while the rest are potential predictors. The data set is available in the `{ElemStatLearn}` package and a description can be found at http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/prostate.info.txt

Questions

Like last time, read the data into R using the following command (with correct file path and name):

```
data(prostate, package = "ElemStatLearn")
prostate <- subset(prostate, select = -train)
```

Q1

What are the coefficients for the predictors `lcavol`, `svi` and `lcp` in the full linear regression model? (If you recall, these are the three predictors most correlated with the response.)

```
# some code
lm(lpsa~., data=prostate)
```

```
##
## Call:
## lm(formula = lpsa ~ ., data = prostate)
##
## Coefficients:
## (Intercept)       lcavol      lweight          age         lbph          svi
```

```
##    0.181561     0.564341     0.622020    -0.021248     0.096713     0.761673
##         lcp      gleason        pgg45
##   -0.106051     0.049228     0.004458
```

The 'lcavol' predictors has the coefficient 0.564341 in the full linear regression model. The 'svi' predictors has the coefficient 0.761673 in the full linear regression model. The 'lcp' predictors has the coefficient -0.106051 in the full linear regression model.

For responses $y_1, \dots, y_n$, vector of predictors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ and fixed penalization parameter λ, the ridge regression solves the optimization problem

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left[\sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^\intercal \boldsymbol{x}_i)^2 + \lambda \boldsymbol{\beta}^\intercal \boldsymbol{\beta} \right],$$

where $\boldsymbol{\beta}$ is the parameter vector (regression coefficients) and λ is a tuning parameter that penalizes $\boldsymbol{\beta}$'s that are "too large". It can be shown that the solution to ridge regression is

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\intercal \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\intercal \boldsymbol{y},$$

where $\boldsymbol{X}$ is the design matrix, $\boldsymbol{y}$ is the response vector and $\boldsymbol{I}$ is the identity matrix of appropriate size.

Q2

What is the value of λ for the regression considered in **Q1**? As $\lambda \to \infty$, what is the behaviour of the vector $\hat{\boldsymbol{\beta}}$ (the ridge regression estimator)?

The value of λ for the regression considered in **Q1** is 0. The behavior of the vector $\hat{\boldsymbol{\beta}}$ will go the zero since the impact of the shrinkage penalty grows and will dominate the minimization.

Now, we perform ridge regression using the function `glmnet` in the package of the same name. Run `install.packages('glmnet')` if you do not have this package installed. Once installed, load the package using `library(glmnet)`. Read the documentation of the function by typing `?glmnet` in the console. We are interested in these four arguments of the function: `x`, `y`, `alpha` and `lambda`.

We consider the sequence of penalties $\boldsymbol{\lambda} = (e^{-3}, e^{-2.8}, e^{-2.6}, \dots, e^{2.6}, e^{2.8}, e^{3})$. Create this vector of length 31 and store it as variable `lam`. Fit the series of ridge regressions to the data set. You will need to use the following command:

```
fittedobject <- glmnet(x=..., y=..., alpha=0, lambda=...)
```

where `alpha=0` specifies the ridge regression (as opposed to other shrinkage methods). Replace the dots by the appropriate variable names.

Hint: The argument `x` accepts a matrix. You can use `as.matrix` to convert a data frame to matrix.

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-4

library(dplyr)


##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

lam <- exp(seq(from = -3, to = 3, by = 0.2))
X <- prostate %>% select(-lpsa) %>% as.matrix()
fittedobject <- glmnet(
x = X,
y = prostate$lpsa,
alpha = 0,
lambda = lam)
```

Q3

Inspect the fitted `glmnet` object using the command `str(fittedobject)`. Find out the variable that stores the fitted coefficients (you may also find the function documentation helpful). Write down the coefficients for the predictors `lcavol`, `svi` and `lcp` when $\lambda = e^3$ and $\lambda = e^0$, respectively.

Hint: `fittedobject` is a list. Variables in a list can be extracted using the `$` operator. For example, `fittedobject$lambda` gives you the λ values used in the series of ridge regressions.

```
# some code
v.3 <- which(fittedobject$lambda == exp(3))
v.0 <- which(fittedobject$lambda == exp(0))
e.3 <-fittedobject$beta[,v.3]
e.0 <-fittedobject$beta[,v.0]
print(e.3)


##       lcavol      lweight          age         lbph          svi          lcp
## 0.0359260221 0.0590038132 0.0011243644 0.0071556936 0.0770832831 0.0214301081
##      gleason        pgg45
## 0.0270053939 0.0007959566
```

```
print(e.0)
```

```
##       lcavol      lweight          age         lbph          svi          lcp
##  0.258691999  0.413684476 -0.002251015  0.048916248  0.445394501  0.076113319
##      gleason        pgg45
##  0.084125760  0.002615953
```

According to the output, as $\lambda = e^3$, the coefficients for the predictors 'lcavol', 'svi', and 'lcp' are 0.0359260221, 0.0770832831, and 0.0214301081 respectively; as $\lambda = e^0$, the coefficients for the predictors 'lcavol', 'svi', and 'lcp' are 0.258691999, 0.445394501, and 0.076113319, respectively.

Next, we find the optimal value of λ (that yields the best predictive ability) via cross validation. The relevant function is `cv.glmnet`. We will also need to use the `nfolds` argument in addition to those above (check the documentation of this function).

Q4

Carry out a 5-fold cross validation, using the same λ values as above. Run `set.seed(406)` **immediately before** the `cv.glmnet()` function to obtain reproducible results. Inspect the fitted object; which variable stores the cross validation mean squared errors?

```
set.seed(406)
cv_fit <- cv.glmnet(
x = X,
y = prostate$lpsa,
nfolds = 5,
alpha = 0,
lambda = lam)
cv_fit
```

```
##
## Call:  cv.glmnet(x = X, y = prostate$lpsa, lambda = lam, nfolds = 5,      alpha = 0)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure      SE Nonzero
## min 0.0907    28  0.5403 0.04671       8
## 1se 0.5488    19  0.5782 0.05088       8
```

```
cv_fit$cvm
```

```
##  [1] 1.1512407 1.1182379 1.0820311 1.0430026 1.0017361 0.9589988 0.9156922
##  [8] 0.8727784 0.8311912 0.7917500 0.7550940 0.7216492 0.6916200 0.6650597
## [15] 0.6418469 0.6218106 0.6047035 0.5902670 0.5782309 0.5683527 0.5603715
## [22] 0.5540657 0.5492236 0.5456493 0.5431340 0.5415100 0.5406087 0.5402706
## [29] 0.5403618 0.5407550 0.5413456
```

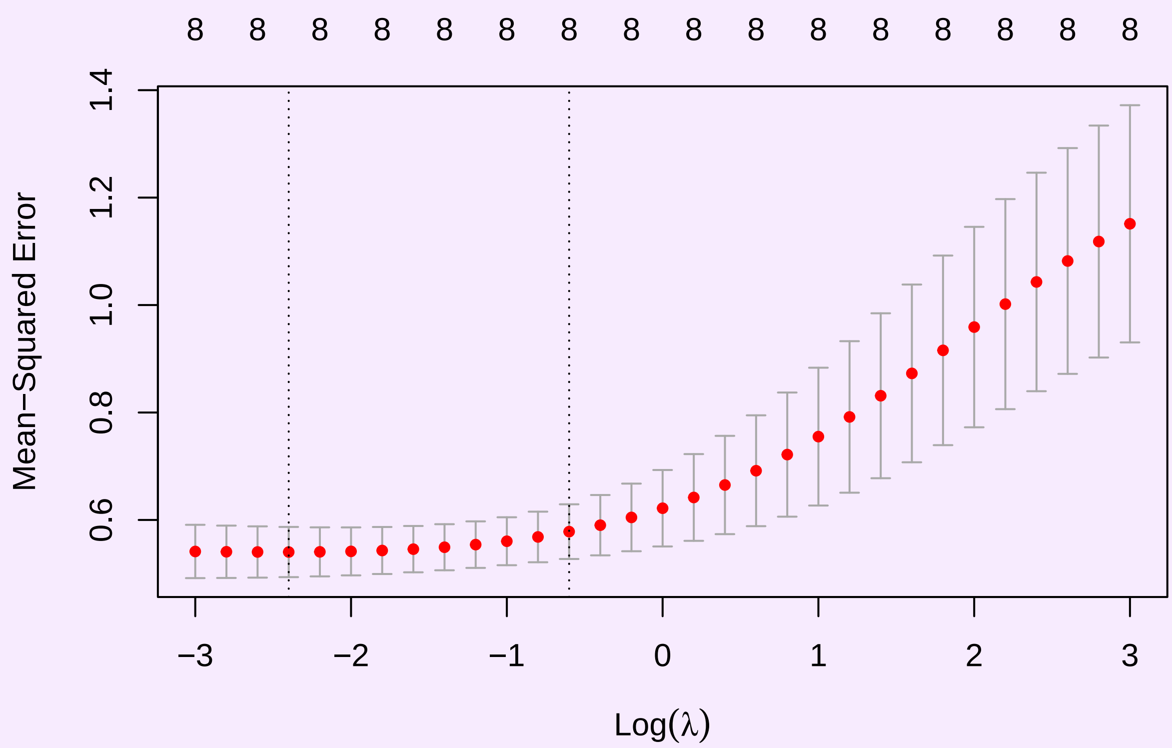
The ‘cvm’ variable stores the cross validation mean squared errors.

Q5

What is the value of λ that minimizes the CV error and what is its associated mean squared error?

Hint: You can visualize the result using `plot` on the fitted object in **Q4**.

```
# some code
plot(cv_fit)
```




```
cv_fit$lambda.min

## [1] 0.09071795

min(cv_fit$cvm)

## [1] 0.5402706
```

$\lambda = 0.09071795$ minimizes the CV error and its associated mean squared error is 0.5402706.

Q6

For this (optimal) λ, what are the fitted coefficients for the predictors `lcavol`, `svi` and `lcp`?

```r
# some code
coefficients(cv_fit, s="lambda.min")
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept)  0.004075402
## lcavol       0.487917541
## lweight      0.602586147
## age         -0.016449738
## lbph         0.085162736
## svi          0.681154529
## lcp         -0.036158357
## gleason      0.064403334
## pgg45        0.003365638
```

The fitted coefficients for the predictors `lcavol`, `svi` and `lcp` are 0.487917541, 0.681154529, and -0.036158357 respectively.