ORF 505: Statistical Analysis of Financial Data

Tuesday Precept 6: Penalized Least-Square Regression

Oct 22, 2024

Lecturer: Qishuo Yin *Scribe: Qishuo Yin*

1 Ridge Regression and LASSO Regression are Penalized Least-Square Regressions

The key idea of the penalized least-square is to minimize a penalized loss function $Q(\cdot)$ as we have done to the least-square loss function.

- Penalized least-squares:

$$Q(\boldsymbol{\beta}) = \frac{1}{2n}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} \mathrm{p}_\lambda(|\beta_j|)$$

  where $\boldsymbol{\beta} \in \mathbb{R}^p$ and $\mathrm{p}_\lambda(\cdot)$ is the penalized term.

- Ridge regression: L_2 penalty $\mathrm{p}_\lambda(\theta) = \lambda|\theta|^2$.

$$Q(\boldsymbol{\beta}) = \frac{1}{2n}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} \beta_j^2$$

- LASSO regression: L_1 penalty $\mathrm{p}_\lambda(\theta) = \lambda|\theta|$.

$$Q(\boldsymbol{\beta}) = \frac{1}{2n}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} |\beta_j|$$

2 Code Basis for Ridge Regression and LASSO Regression

Question: How are we going to conduct ridge regression and LASSO regression on a dataset by R codes? Here is a simple example of polynomial regression: Suppose we have a sample `x.random` of size `n = 1000` from the standard normal distribution and `x` is its ordered vector. `Y` is a vector also of size `n = 1000` created by adding independent Gaussian variates with mean 0 and standard deviation $\sigma = 2$ to the polynomial regression function

$$\varphi_\beta(x) = x^2 - 10x^3 + x^5$$

.

```
library(glmnet) # package we are going to use for penalized least-square
    regression
set.seed(1111)
n=1000
x.random= array(rnorm(n,mean=0,sd=1), dim=c(n,1))
x = x.random[order(x.random)]
X = cbind(x,x^2,x^3,x^4,x^5)
beta = c(0,1,-10,0,1)
Y = X%*%beta + rnorm(n,mean=0,sd=4)
```

Listing 1: set up dataset

2.1 Ridge Regression

For ridge regression, we run the function `glmnet` with the parameter `alpha = 0`. By default, this function runs the ridge regression optimization for a set of 100 values of the hyper-paremeter λ. For convenience, we store these values in a vector called `s`.

```
ridge = glmnet(X,Y, alpha=0)
s=ridge$lambda
```

Listing 2: ridge regression

We plot the scatterplot of x and Y, and graphs of the regression functions with different values of λ.

```
options(repr.plot.width = 8, repr.plot.height = 6)

# plot scatterplot of x and Y
plot(x,Y)

# plot regression function with lambda = s[1]
x.div = seq(from= min(x),to=max(x),length=1000)
X.div = cbind(x.div,x.div^2,x.div^3,x.div^4,x.div^5)
y.hat = X.div%*% ridge$beta[,1]
lines(x.div,y.hat,col=1,lty=1)
txt1 = paste("lambda=",signif(s[1],5))

# plot regression function with lambda = s[33]
y.hat = X.div%*% ridge$beta[,33]
lines(x.div,y.hat,col=2,lty=1)
txt2= paste("lambda=",signif(s[33],5))

# plot regression function with lambda = s[66]
y.hat = X.div%*% ridge$beta[,66]
lines(x.div,y.hat,col=3,lty=1)
txt3 = paste("lambda=",signif(s[66],5))

# plot regression function with lambda = s[100]
y.hat = X.div%*% ridge$beta[,100]
lines(x.div,y.hat,col=4,lty=1)
```

```r
txt4 = paste("lambda=",signif(s[100],5))

# plot regression function with lambda = 0
y.hat = predict(ridge, s=0, newx=X.div)
lines(x.div,y.hat,col=5,lty=1)
txt5 = paste("lambda=",0.0)
legend("bottomleft",c(txt1,txt2,txt3,txt4,txt5),col=c(1:5),lty=c(1,1,1,1,1))
title("Ridge regression for several values of the hyper-parameter lambda")
```

Listing 3: scatterplot and ridge regression function graphs

At the end of this section, we present a way to compute the optimal value of λ as given by 10-fold cross-validation directly from R codes.

```r
set.seed(1111)
cv.ridge = cv.glmnet(X,Y, alpha=0)
options(repr.plot.width = 6, repr.plot.height = 4)
plot(cv.ridge)
cat(paste("Best lambda = ",signif(cv.ridge$lambda.min,5)))
coef(cv.ridge)
```

Listing 4: optimal value of λ in ridge regression

2.2 LASSO Regression

For LASSO regression, we run the function `glmnet` with the parameter `alpha = 0`. And define the different values of λ we want to take by `li`:

```r
lasso = glmnet(X,Y, alpha=1)
s=lasso$lambda
ls = length(s)
li = c(1,floor(ls/3),floor(2*ls/3),ls)
```

Listing 5: LASSO regression

Then, we plot the scatterplot of x and Y, and graphs of the regression function with different values of λ:

```r
options(repr.plot.width = 9, repr.plot.height = 6)

# plot scatterplot of x and Y
plot(x,Y)

# plot regression function with lambda = s[1]
x.div = seq(from= min(x),to=max(x),length=1000)
X.div = cbind(x.div,x.div^2,x.div^3,x.div^4,x.div^5)
y.hat = X.div%*% lasso$beta[,li[1]]
lines(x.div,y.hat,col=1,lty=1)
txt1 = paste("lambda=",signif(s[li[1]],5))

# plot regression function with lambda = s[floor(length(s))/3]
y.hat = X.div%*% lasso$beta[,li[2]]
```

```r
lines(x.div,y.hat,col=2,lty=1)
txt2= paste("lambda=",signif(s[li[2]],5))

# plot regression function with lambda = s[2*floor(length(s))/3]
y.hat = X.div%*% lasso$beta[,li[3]]
lines(x.div,y.hat,col=3,lty=1)
txt3 = paste("lambda=",signif(s[li[3]],5))

# plot regression function with lambda = s[length(s)]
y.hat = X.div%*% lasso$beta[,li[4]]
lines(x.div,y.hat,col=4,lty=1)
txt4 = paste("lambda=",signif(s[li[4]],5))

# plot regression function with lambda = 0
y.hat = predict(lasso, s=0, newx=X.div)
lines(x.div,y.hat,col=5,lty=1)
txt5 = paste("lambda=",0.0)
legend("bottomleft",c(txt1,txt2,txt3,txt4,txt5),col=c(1:5),lty=c(1,1,1,1,1))
title("LASSO regression for several values of the hyper-parameter lambda")
```

Listing 6: scatterplot and LASSO regression function graphs

Similar to ridge regression, there is also a way to compute the optimal value of λ as given by 10-fold cross-validation directly from R codes.

```r
set.seed(1111)
cv.lasso = cv.glmnet(X,Y, alpha=1)
cat(paste("Best lambda = ",signif(cv.lasso$lambda.min,5)))
coef(cv.lasso)
options(repr.plot.width = 6, repr.plot.height = 4)
plot(cv.lasso)
```

Listing 7: optimal value of λ in LASSO regression

3 Extensions of Penalized Least-Square Family

- Smoothly clipped absolute deviation(SCAD)

$$p'_\lambda(\theta) = \lambda\left\{\mathbb{I}(\theta \le \lambda) + \frac{(a\lambda-\theta)_+}{(a-1)\lambda}\mathbb{I}(\theta > \lambda)\right\}, \quad a > 2$$

  with the solution to be

$$\widehat{\theta}_{\text{SCAD}}(z) = \begin{cases} \text{sgn}(z)(|z|-\lambda)_+, & \text{when } |z| \le 2\lambda; \\ \text{sgn}(z)[(a-1)|z| - a\lambda]/(a-2), & \text{when } 2\lambda < |z| \le a\lambda; \\ z, & \text{when } |z| \ge a\lambda. \end{cases}$$

- Minimax concave penalty (MCP)

$$p'_\lambda(t) = (\lambda - t/a)_+$$

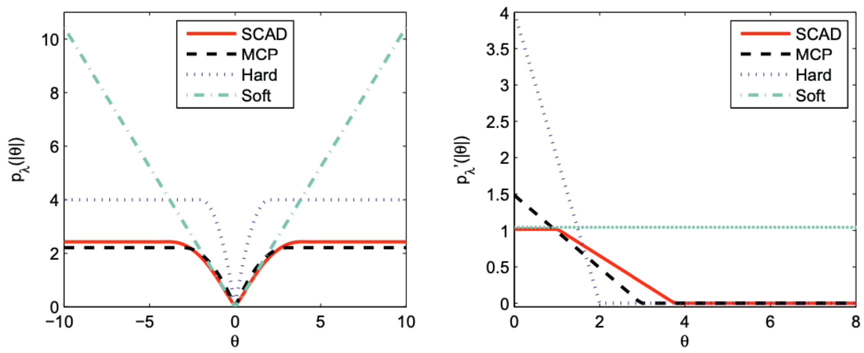

Figure 3.2: Some commonly used penalty functions (left panel) and their derivatives (right panel). They correspond to the risk functions shown in the right panel of Figure 3.3. More precisely, $\lambda = 2$ for hard thresholding penalty, $\lambda = 1.04$ for L_1-penalty, $\lambda = 1.02$ for SCAD with $a = 3.7$, and $\lambda = 1.49$ for MCP with $a = 2$. Taken from Fan and Lv (2010).

Figure 1: SCAD and MCP

with the solution to be

$$\widehat{\theta}_{\text{MCP}}(z) = \begin{cases} \text{sgn}(z)(|z| - \lambda)_+/(1 - 1/a), & \text{when } |z| < a\lambda; \\ z, & \text{when } |z| \geq a\lambda. \end{cases}$$

The reason why these estimators are introduced may be explained by the following plot from section 3.2 Folded-concave Penalized Least Squares of [1].

References

[1] J. Fan, R. Li, C.-H. Zhang, and H. Zou. *Statistical foundations of data science*. Chapman and Hall/CRC, 2020.