Homework 3

Liu Huihang

12/15/2019

Variable Selection

Data Preparation

```
test <- read.csv('test.txt', header = TRUE, sep=" ")
train <- read.csv('training.txt', header=TRUE, sep=" ")

Y.test <- test[, 1]
X.test <- as.matrix(test[,-1])
Y.train <- train[, 1]
X.train <- as.matrix(train[,-1])
```

Linear Model

```
fit.lm <- lm(paste0("Y~",paste("X",1:20,sep="",collapse="+")), data=train)
summary(fit.lm)
```

```
##
## Call:
## lm(formula = paste0("Y~", paste("X", 1:20, sep = "", collapse = "+")),
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.41496 -0.57431  0.05948  0.67349  2.20278
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.09476    0.12365   0.766 0.445734
## X1           1.89672    0.29736   6.378 1.12e-08 ***
## X2          -1.84688    0.38011  -4.859 5.87e-06 ***
## X3           0.04642    0.39496   0.118 0.906731
## X4          -0.57919    0.33141  -1.748 0.084409 .
## X5           0.32361    0.36552   0.885 0.378666
## X6           0.07419    0.36851   0.201 0.840954
## X7           0.14948    0.38306   0.390 0.697429
## X8          -0.35901    0.34909  -1.028 0.306892
## X9           0.03714    0.39164   0.095 0.924688
## X10         -0.02658    0.37669  -0.071 0.943930
## X11          2.23001    0.39541   5.640 2.55e-07 ***
## X12         -1.65636    0.41881  -3.955 0.000166 ***
## X13          0.24762    0.42698   0.580 0.563615
## X14         -0.75995    0.43843  -1.733 0.086935 .
## X15          0.01858    0.38901   0.048 0.962019
## X16          0.37625    0.40257   0.935 0.352828
## X17         -0.74810    0.50770  -1.474 0.144587
## X18          0.66028    0.40236   1.641 0.104767
```

```
## X19         -0.59440    0.39847  -1.492 0.139764
## X20          0.46504    0.31757   1.464 0.147063
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.109 on 79 degrees of freedom
## Multiple R-squared:  0.605,  Adjusted R-squared:  0.505
## F-statistic: 6.051 on 20 and 79 DF,  p-value: 2.728e-09
```

We can obtain the information of residuals and the estimators of coefficients with significance levels. It shows that X1, X2, X4, X11, X12, X14 are significant. Other variables are close to zero but not equal to zero.

Variable Selection by Lasso, SCAD and MCP

I use package *glmnet* and *ncvreg* to apply Lasso, SCAD and MCP. They are very easy to use and return friendly results.

```
# Lasso
library(glmnet)
fit.lasso.cv <- cv.glmnet(X.train, Y.train)
fit.lasso <- glmnet(X.train,Y.train, lambda=fit.lasso.cv$lambda.min)
res.lasso <- fit.lasso$beta[which(fit.lasso$beta != 0)]
names(res.lasso) <- colnames(X.train)[which(fit.lasso$beta != 0)]
print(res.lasso)
```

```
##          X1          X2          X4          X6         X11         X12
##  1.50348685 -1.45891189 -0.30547908  0.09014220  1.78024998 -1.16650539
##         X14         X18         X20
## -0.55781040  0.02226962  0.06515478
```

```
# SCAD
library(ncvreg)
fit.scad.cv <- cv.ncvreg(X.train, Y.train, family="gaussian", penalty="SCAD")
fit.scad <- ncvreg(X.train, Y.train, family="gaussian", penalty="SCAD", lambda=fit.scad.cv$lambda.min)
res.scad <- fit.scad$beta[which(fit.scad$beta[2:21] != 0)+1]
names(res.scad) <- colnames(X.train)[which(fit.scad$beta[2:21] != 0)]
print(res.scad)
```

```
##          X1          X2          X4         X11         X12         X14
##  1.85497989 -1.94422281 -0.06266673  2.12895428 -1.52726806 -0.54008315
```

```
# MCP
fit.mcp.cv <- cv.ncvreg(X.train, Y.train, family="gaussian", penalty="MCP")
fit.mcp <- ncvreg(X.train, Y.train, family="gaussian", penalty="MCP", lambda=fit.mcp.cv$lambda.min)
res.mcp <- fit.mcp$beta[which(fit.mcp$beta[2:21] != 0)+1]
names(res.mcp) <- colnames(X.train)[which(fit.mcp$beta[2:21] != 0)]
print(res.mcp)
```

```
##         X1         X2        X11        X12        X14
##  1.8553869 -1.9836081  2.1056606 -1.5198180 -0.5406908
```

I use cross validation to choose turning parameters. The None-Zero coefficients are printed above. We can obtain they small model from the non-zero coefficients.

MCP gives the most sparse model, and lasso returns the most number of non-zero coefficients.

Analysis Prediction Errors

```
Y.hat.lasso <- predict.glmnet(fit.lasso, X.test)
Y.hat.scad <- predict(fit.scad, X.test)
Y.hat.mcp <- predict(fit.mcp, X.test)

err <- c(lasso=sum((Y.hat.lasso-Y.test)^2) / 100,
         scad=sum((Y.hat.scad-Y.test)^2) / 100,
         mcp=sum((Y.hat.mcp-Y.test)^2) / 100)
# print("Prediction Errors: ")
print(err)
```

```
##    lasso     scad      mcp
## 1.154580 1.082511 1.084787
```

Summary

I apply Lasso, SCAD and MCP to select variables.

Lasso

Lasso consider the following optimal problem.

$$\arg\min \|y - X\beta\|_2^2 \quad \text{subject to } |\beta| \leq t \tag{1}$$

where t is a tuning parameter.

Both SCAD and MCP consider the objective function

$$\frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{d} P_j\left(\beta_j|\lambda,\gamma\right). \tag{2}$$

where $P(\beta|\lambda,\gamma)$ is a folded concave penalty.

SCAD

The smoothly clipped absolute deviations (SCAD) penalty is defined as

$$P(x|\lambda,\gamma) = \begin{cases} \lambda|x| & \text{if } |x| \leq \lambda \\ \frac{2\gamma\lambda|x| - x^2 - \lambda^2}{2(\gamma-1)} & \text{if } \lambda < |x| < \gamma\lambda \\ \frac{\lambda^2(\gamma+1)}{2} & \text{if } |x| \geq \gamma\lambda \end{cases} \tag{3}$$

or the continuous differentiable penalty function defined by

$$P(x|\lambda,\gamma)'(\theta) = \lambda\left\{I(\theta \leq \lambda) + \frac{(a\lambda - \theta)_+}{(a-1)\lambda} I(\theta > \lambda)\right\} \tag{4}$$

for some $a > 2$ and $\theta > 0$.

MCP

The idea behind the minimax concave penalty (MCP) is very similar with SCAD:

$$P_\gamma(x;\lambda) = \begin{cases} \lambda|x| - \frac{x^2}{2\gamma}, & \text{if } |x| \leq \gamma\lambda \\ \frac{1}{2}\gamma\lambda^2, & \text{if } |x| > \gamma\lambda \end{cases} \tag{5}$$

for $\gamma > 1$.

Its derivative is

$$P'_\gamma(x;\lambda) = \begin{cases} \left(\lambda - \frac{|x|}{\gamma}\right)\text{sign}(x), & \text{if } |x| \leq \gamma\lambda \\ 0, & \text{if } |x| > \gamma\lambda \end{cases} \tag{6}$$

The primary way in which SCAD, and MCP differ from the lasso is that they allow the estimated coefficients to reach large values more quickly than the lasso.

In other words, SCAD and MCP apply less shrinkage to the nonzero coefficients to achieve bias reduction.

From the result above, we can find that SCAD and MCP have lower prediction errors than Lasso. Although, all of these three method give sapse estimations.