# 第八章

**1. 请分析MASS包中的Boston数据集**

**(1) 利用LASSO、MCP和SCAD3种惩罚方法分析找出影响房价medv的因素，比较一下这些方法找出的影响因素。**

**(2) 比较LASSO方法和逐步回归方法筛选出来的结果。**

解：

先将Boston数据集分为自变量和因变量：

# 导入所需的包

library(MASS)

library(glmnet)

# 加载Boston数据集

data(Boston)

# 将数据集分为自变量和因变量

X <- as.matrix(Boston[, -14]) # 自变量

y <- Boston$medv # 因变量

1. LASSO：

# LASSO

lasso\_model <- cv.glmnet(X, y, family="gaussian")

lasso\_coef <- coef(lasso\_model)

lasso\_coef

resid1 <- (X %\*% lasso\_coef [ -1 ] + lasso\_coef [ 1 ] - y)

MSE1 <- sum (resid1 ^ 2)

MSE1

输出结果：

> lasso\_coef

14 x 1 sparse Matrix of class "dgCMatrix"

s1

(Intercept) 17.471486288

crim -0.021826808

zn .

indus .

chas 1.897718337

nox -3.341293717

rm 4.266279454

age .

dis -0.317122209

rad .

tax .

ptratio -0.787265492

black 0.006558204

lstat -0.518129464

> MSE1

[1] 12633.63

通过比较输出结果可发现，zn、indus、age、rad和tax均被移除，留下crim、chas、nox、rm、dis、ptratio、black、lstat8项。

②MCP：

# MCP

fit2 <- cv.ncvreg(X, y, family="gaussian")

fit.mcp <- fit2$fit

beta.fit2 <- fit.mcp$beta [ , fit2$min]

round( beta.fit2 , 3)

resid2 <- (X %\*% beta.fit2 [ -1 ] + beta.fit2 [ 1 ] - y)

MSE2 <- sum (resid2 ^ 2)

MSE2

得到如下的输出结果：

> round( beta.fit2 , 3)

(Intercept) crim zn indus chas

36.351 -0.108 0.046 0.000 2.719

nox rm age dis rad

-17.384 3.801 0.000 -1.493 0.300

tax ptratio black lstat

-0.012 -0.947 0.009 -0.523

> MSE2

[1] 11081.36

通过比较输出结果可发现， indus、age系数为0，均可移除，留下crim、zn、chas、nox、rm、dis、rad、tax、ptratio、black、lstat11项。

③SCAD：

# SCAD

fit3 <- cv.ncvreg(X, y, family="gaussian",penalty="SCAD")

fit.scad <- fit3$fit

beta.fit3 <- fit.scad$beta[, fit3$min]

round(beta.fit3, 3)

resid3 <- (X %\*% beta.fit3 [ -1 ] + beta.fit3 [ 1 ] - y)

MSE3 <- sum (resid3 ^ 2)

MSE3

输出结果：

> round(beta.fit3, 3)

(Intercept) crim zn indus chas

36.346 -0.108 0.046 0.000 2.719

nox rm age dis rad

-17.380 3.802 0.000 -1.493 0.300

tax ptratio black lstat

-0.012 -0.947 0.009 -0.523

> MSE3

[1] 11081.36

通过比较输出结果可发现， indus、age系数为0，均可移除，留下crim、zn、chas、nox、rm、dis、rad、tax、ptratio、black、lstat11项。

总的来说，LASSO模型中zn、indus、age、rad和tax均被移除，而MCP和SCAD均只移除了indus、age变量。通过比较MSE，发现MCP和SCAD的MSE相同，且均比LASSO模型的MSE小，所以MCP和SCAD的估计效果都比LASSO好，且两者相近。

（2）LASSO方法的结果在上题已有，逐步回归的代码：

# 逐步回归

stepwise\_model <- step(lm(y ~ ., data = as.data.frame(X)), direction = "both")

stepwise\_model

MSE4 <- sum(stepwise\_model$residuals^2)

MSE4

结果：

lm(formula = y ~ crim + zn + chas + nox + rm + dis + rad + tax +

ptratio + black + lstat, data = as.data.frame(X))

Coefficients:

(Intercept) crim zn chas

36.341145 -0.108413 0.045845 2.718716

nox rm dis rad

-17.376023 3.801579 -1.492711 0.299608

tax ptratio black lstat

-0.011778 -0.946525 0.009291 -0.522553

> MSE4

[1] 11081.36

逐步回归中剔除了indus和age变量，MSE也小于LASSO模型，所以逐步回归模型同样比LASSO模型更优。

**2. 请分析ISLR包中的Smarket数据集。以Direction为因变量，请用LASSO、MCP和SCAD3种惩罚方法分析找出影响股票交割涨跌方向的因素，并比较3种方法找出的影响因素是否一样。**

解：

本题自变量Direction为字符串类型“Up”和“Down”，故将其转换成1和-1表示方向：

# 导入所需的包

library(glmnet)

library(ncvreg)

library(ISLR)

# 加载Boston数据集

data(Smarket)

# 将数据集分为自变量和因变量

X <- as.matrix(Smarket[, -9]) # 自变量

y\_string <- Smarket$Direction

y <- ifelse(y\_string == "Up", 1, -1) # 转换

① LASSO

# LASSO

lasso\_model <- cv.glmnet(X, y, family="gaussian")

lasso\_coef <- coef(lasso\_model)

lasso\_coef

resid1 <- (X %\*% lasso\_coef [ -1 ] + lasso\_coef [ 1 ] - y)

MSE1 <- sum (resid1 ^ 2)

MSE1

输出结果：

> lasso\_coef

9 x 1 sparse Matrix of class "dgCMatrix"

s1

(Intercept) 0.03496241

Year .

Lag1 .

Lag2 .

Lag3 .

Lag4 .

Lag5 .

Volume .

Today 0.58551662

> MSE1

[1] 587.3379

可见LASSO回归只留下了Today变量，其余均被剔除，MSE为587.3379

② MCP：

# MCP

fit2 <- cv.ncvreg(X, y, family="gaussian")

fit.mcp <- fit2$fit

beta.fit2 <- fit.mcp$beta [ , fit2$min]

round( beta.fit2 , 3)

resid2 <- (X %\*% beta.fit2 [ -1 ] + beta.fit2 [ 1 ] - y)

MSE2 <- sum (resid2 ^ 2)

MSE2

输出结果：

> round( beta.fit2 , 3)

(Intercept) Year Lag1 Lag2 Lag3

-39.909 0.020 0.000 0.000 0.000

Lag4 Lag5 Volume Today

0.000 0.000 0.000 0.642

> MSE2

[1] 579.3507

可见MCP也只选择了Today，MSE为579.3507

③ SCAD：

# SCAD

fit3 <- cv.ncvreg(X, y, family="gaussian",penalty="SCAD")

fit.scad <- fit3$fit

beta.fit3 <- fit.scad$beta[, fit3$min]

round(beta.fit3, 3)

resid3 <- (X %\*% beta.fit3 [ -1 ] + beta.fit3 [ 1 ] - y)

MSE3 <- sum (resid3 ^ 2)

MSE3

输出结果：

> round(beta.fit3, 3)

(Intercept) Year Lag1 Lag2 Lag3

-58.260 0.029 -0.002 0.000 0.000

Lag4 Lag5 Volume Today

0.000 0.009 0.000 0.642

> MSE3

[1] 578.0169

可见SCAD选择了Year、Lag1、Lag5和Today，其MSE为578.0169。

总的来说，LASSO和MCP选择了同样的影响因素，但MCP的MSE较低，SCAD较它俩的MSE更低，且选择的变量多，故SCAD模型更优。

**3. 请模拟生成，由多元正态分布产生，p=100，n=100，对应的相关系数是, =0.1、0.5、0.9，回归系数=(1,1,1,1,1,0.5,0.5,0.5,0.5,0.5,0,…,0), 随机扰动项是标准正态分布，请模拟100次，分别用LASSO、MCP和SCAD筛选变量，比较变量筛选的FNR和FDR。**

解：

代码：

library(glmnet)

library(ncvreg)

set.seed(114514)

# 模拟参数

p <- 100

n <- 100

rho <- c(0.1, 0.5, 0.9)

beta <- c(rep(1, 5), rep(0.5, 5), rep(0, p-10))

num\_simulations <- 100

# 初始化结果向量

fnr\_lasso <- fnr\_mcp <- fnr\_scad <- rep(0, length(rho))

fdr\_lasso <- fdr\_mcp <- fdr\_scad <- rep(0, length(rho))

for (sim in 1:num\_simulations) {

for (r in 1:length(rho)) {

# 生成相关矩阵

Sigma <- matrix(0, nrow = p, ncol = p)

for (i in 1:p) {

for (j in 1:p) {

Sigma[i, j] <- rho[r]^(abs(i-j))

}

}

# 生成数据

X <- MASS::mvrnorm(n, rep(0, p), Sigma)

y <- X %\*% beta + rnorm(n)

# LASSO回归

fit\_lasso <- cv.glmnet(X, y, alpha = 1)

coef\_lasso <- coef(fit\_lasso, s = "lambda.min")

selected\_vars\_lasso <- which(coef\_lasso != 0)

# MCP回归

fit\_mcp <- cv.ncvreg(X, y, family="gaussian", penalty = "MCP")

coef\_mcp <- coef(fit\_mcp)

selected\_vars\_mcp <- which(coef\_mcp != 0)

# SCAD回归

fit\_scad <- cv.ncvreg(X, y, family="gaussian", penalty = "SCAD")

coef\_scad <- coef(fit\_scad)

selected\_vars\_scad <- which(coef\_scad != 0)

# 计算FNR和FDR

true\_vars <- which(beta != 0)

fnr\_lasso[r] <- fnr\_lasso[r] + sum(!(true\_vars %in% selected\_vars\_lasso)) / length(true\_vars)

fdr\_lasso[r] <- fdr\_lasso[r] + sum(!(selected\_vars\_lasso %in% true\_vars)) / length(selected\_vars\_lasso)

fnr\_mcp[r] <- fnr\_mcp[r] + sum(!(true\_vars %in% selected\_vars\_mcp)) / length(true\_vars)

fdr\_mcp[r] <- fdr\_mcp[r] + sum(!(selected\_vars\_mcp %in% true\_vars)) / length(selected\_vars\_mcp)

fnr\_scad[r] <- fnr\_scad[r] + sum(!(true\_vars %in% selected\_vars\_scad)) / length(true\_vars)

fdr\_scad[r] <- fdr\_scad[r] + sum(!(selected\_vars\_scad %in% true\_vars)) / length(selected\_vars\_scad)

}

}

# 计算平均值

fnr\_lasso <- fnr\_lasso / num\_simulations

fdr\_lasso <- fdr\_lasso / num\_simulations

fnr\_mcp <- fnr\_mcp / num\_simulations

fdr\_mcp <- fdr\_mcp / num\_simulations

fnr\_scad <- fnr\_scad / num\_simulations

fdr\_scad <- fdr\_scad / num\_simulations

# 输出结果

result <- data.frame(rho, fnr\_lasso, fdr\_lasso, fnr\_mcp, fdr\_mcp, fnr\_scad, fdr\_scad)

print(result)

输出结果：

> print(result)

rho fnr\_lasso fdr\_lasso fnr\_mcp fdr\_mcp fnr\_scad

1 0.1 0.001 0.6694312 0.008 0.34246268 0.003

2 0.5 0.000 0.5314077 0.085 0.32379577 0.056

3 0.9 0.028 0.3790730 0.566 0.09647619 0.568

fdr\_scad

1 0.5214510

2 0.4387296

3 0.1848373

可见，固定rho时，lasso的fnr都小于mcp和scad，dfr都大于mcp和scad。当rho逐渐增大，即扰动项增大时lasso的fnr较为平稳都接近0，fdr逐渐减小；而mcp的fnr逐渐增大，且当rho=0.9时较为明显，达到0.566，fdr逐渐下降；scad的fnr也逐渐上升，fdr逐渐下降；

# 第九章