Practice6

A.Lukyanova

7 апреля 2018 г

Загрузим данные для выполнения задания: загрузим пакет Boston и установим ядро.

my.seed <- 1  
data(Boston)  
str(Boston)

## 'data.frame': 506 obs. of 14 variables:  
## $ crim : num 0.00632 0.02731 0.02729 0.03237 0.06905 ...  
## $ zn : num 18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...  
## $ indus : num 2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...  
## $ chas : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ nox : num 0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...  
## $ rm : num 6.58 6.42 7.18 7 7.15 ...  
## $ age : num 65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...  
## $ dis : num 4.09 4.97 4.97 6.06 6.06 ...  
## $ rad : int 1 2 2 3 3 3 5 5 5 5 ...  
## $ tax : num 296 242 242 222 222 222 311 311 311 311 ...  
## $ ptratio: num 15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...  
## $ black : num 397 397 393 395 397 ...  
## $ lstat : num 4.98 9.14 4.03 2.94 5.33 ...  
## $ medv : num 24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...

fix(Boston)  
names(Boston)

## [1] "crim" "zn" "indus" "chas" "nox" "rm" "age"   
## [8] "dis" "rad" "tax" "ptratio" "black" "lstat" "medv"

dim(Boston)

## [1] 506 14

Выполним пошаговое включение регрессоров.

regfit.fwd <- regsubsets(crim ~ ., data = Boston,  
 nvmax = 13, method = 'forward')  
summary(regfit.fwd)

## Subset selection object  
## Call: regsubsets.formula(crim ~ ., data = Boston, nvmax = 13, method = "forward")  
## 13 Variables (and intercept)  
## Forced in Forced out  
## zn FALSE FALSE  
## indus FALSE FALSE  
## chas FALSE FALSE  
## nox FALSE FALSE  
## rm FALSE FALSE  
## age FALSE FALSE  
## dis FALSE FALSE  
## rad FALSE FALSE  
## tax FALSE FALSE  
## ptratio FALSE FALSE  
## black FALSE FALSE  
## lstat FALSE FALSE  
## medv FALSE FALSE  
## 1 subsets of each size up to 13  
## Selection Algorithm: forward  
## zn indus chas nox rm age dis rad tax ptratio black lstat medv  
## 1 ( 1 ) " " " " " " " " " " " " " " "\*" " " " " " " " " " "   
## 2 ( 1 ) " " " " " " " " " " " " " " "\*" " " " " " " "\*" " "   
## 3 ( 1 ) " " " " " " " " " " " " " " "\*" " " " " "\*" "\*" " "   
## 4 ( 1 ) " " " " " " " " " " " " " " "\*" " " " " "\*" "\*" "\*"   
## 5 ( 1 ) "\*" " " " " " " " " " " " " "\*" " " " " "\*" "\*" "\*"   
## 6 ( 1 ) "\*" " " " " " " " " " " "\*" "\*" " " " " "\*" "\*" "\*"   
## 7 ( 1 ) "\*" " " " " "\*" " " " " "\*" "\*" " " " " "\*" "\*" "\*"   
## 8 ( 1 ) "\*" " " " " "\*" " " " " "\*" "\*" " " "\*" "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" " " "\*" " " " " "\*" "\*" " " "\*" "\*" "\*" "\*"   
## 10 ( 1 ) "\*" "\*" " " "\*" "\*" " " "\*" "\*" " " "\*" "\*" "\*" "\*"   
## 11 ( 1 ) "\*" "\*" " " "\*" "\*" " " "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## 12 ( 1 ) "\*" "\*" "\*" "\*" "\*" " " "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## 13 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"

round(coef(regfit.fwd, 13), 3)

## (Intercept) zn indus chas nox rm   
## 17.033 0.045 -0.064 -0.749 -10.314 0.430   
## age dis rad tax ptratio black   
## 0.001 -0.987 0.588 -0.004 -0.271 -0.008   
## lstat medv   
## 0.126 -0.199

Запишем функцию для прогноза функции:

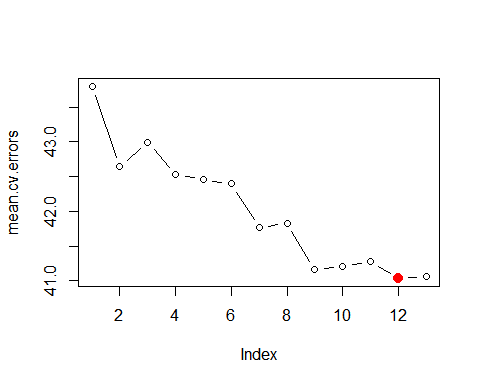
predict.regsubsets <- function(object, newdata, id, ...){  
 form <- as.formula(object$call[[2]])  
 mat <- model.matrix(form, newdata)  
 coefi <- coef(object, id = id)  
 xvars <- names(coefi)  
 mat[, xvars] %\*% coefi  
}

Выполним кросс-валидацию для нахождения оптимальной модели

# k-кратная кросс-валидация  
# отбираем 10 блоков наблюдений  
k <- 10  
set.seed(my.seed)  
folds <- sample(1:k, nrow(Boston), replace = T)  
  
# заготовка под матрицу с ошибками  
cv.errors <- matrix(NA, k, 13, dimnames = list(NULL, paste(1:13)))  
  
# заполняем матрицу в цикле по блокам данных  
for (j in 1:k){  
 best.fit <- regsubsets(crim ~ ., data = Boston[folds != j, ],  
 nvmax = 13)  
 # теперь цикл по количеству объясняющих переменных  
 for (i in 1:13){  
 # модельные значения Boston  
 pred <- predict(best.fit, Boston[folds == j, ], id = i)  
 # вписываем ошибку в матрицу  
 cv.errors[j, i] <- mean((Boston$crim[folds == j] - pred)^2)  
 }  
}  
  
# усредняем матрицу по каждому столбцу (т.е. по блокам наблюдений),   
# чтобы получить оценку MSE для каждой модели с фиксированным   
# количеством объясняющих переменных  
mean.cv.errors <- apply(cv.errors, 2, mean)  
round(mean.cv.errors, 0)

## 1 2 3 4 5 6 7 8 9 10 11 12 13   
## 44 43 43 43 42 42 42 42 41 41 41 41 41

# на графике  
plot(mean.cv.errors, type = 'b')  
points(which.min(mean.cv.errors), mean.cv.errors[which.min(mean.cv.errors)],  
 col = 'red', pch = 20, cex = 2)



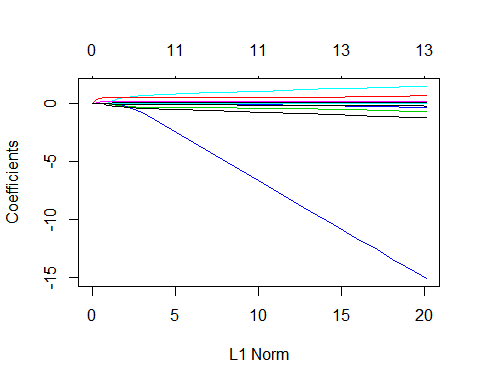
# перестраиваем модель с 12 объясняющими переменными на всём наборе данных  
reg.best <- regsubsets(crim ~ ., data = Boston, nvmax = 13)  
round(coef(reg.best, 12), 3)

## (Intercept) zn indus chas nox rm   
## 16.986 0.045 -0.064 -0.744 -10.202 0.440   
## dis rad tax ptratio black lstat   
## -0.994 0.588 -0.004 -0.270 -0.008 0.128   
## medv   
## -0.199

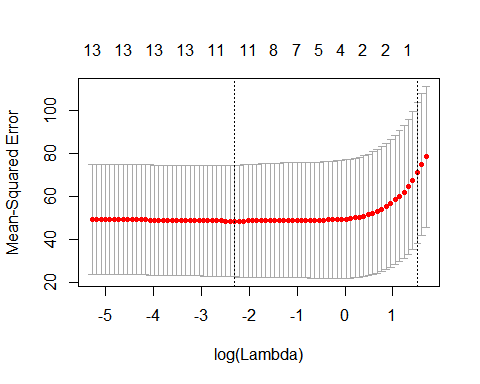
Наименьшая MSE на тестовой выборке оказалась у модели с 12 предикторами (MSE=41).

Воспльзуемся методом сжатия, а именно лассо-регрессией:

#лассо-регрессия  
# из-за синтаксиса glmnet() формируем явно матрицу объясняющих...  
x <- model.matrix(crim ~ ., Boston)[, -1]  
  
# и вектор значений зависимой переменной  
y <- Boston$crim  
  
set.seed(my.seed)  
train <- sample(1:nrow(x), nrow(x)/2)  
test <- -train  
y.test <- y[test]  
  
# вектор значений гиперпараметра лямбда  
grid <- 10^seq(10, -2, length = 100)  
  
lasso.mod <- glmnet(x[train, ], y[train], alpha = 1, lambda = grid)  
plot(lasso.mod)



set.seed(my.seed)  
cv.out <- cv.glmnet(x[train, ], y[train], alpha = 1)  
plot(cv.out)



bestlam <- cv.out$lambda.min  
lasso.pred <- predict(lasso.mod, s = bestlam, newx = x[test, ])  
#MSE на тестовой  
round(mean((lasso.pred - y.test)^2), 0)

## [1] 38

# коэффициенты лучшей модели  
out <- glmnet(x, y, alpha = 1, lambda = grid)  
lasso.coef <- predict(out, type = 'coefficients',  
 s = bestlam)[1:13, ]  
round(lasso.coef, 3)

## (Intercept) zn indus chas nox rm   
## 9.263 0.031 -0.051 -0.513 -3.755 0.041   
## age dis rad tax ptratio black   
## 0.000 -0.601 0.495 0.000 -0.108 -0.008   
## lstat   
## 0.118

round(lasso.coef[lasso.coef != 0], 3)

## (Intercept) zn indus chas nox rm   
## 9.263 0.031 -0.051 -0.513 -3.755 0.041   
## dis rad ptratio black lstat   
## -0.601 0.495 -0.108 -0.008 0.118

МSE оказалась ниже чем у метода пошагового включения, MSE=38.