DATA 303/473 Assignment 3 Solution

Boston Data Set

In this assignment, we use `Boston` data set. This is a data set containing housing values in 506 suburbs of Boston (a data frame with 506 rows and 13 variables).

- crim: per capita crime rate by town.
- zn: proportion of residential land zoned for lots over 25,000 sq.ft.
- indus: proportion of non-retail business acres per town.
- chas: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
- nox: nitrogen oxides concentration (parts per 10 million).
- rm: average number of rooms per dwelling.
- age: proportion of owner-occupied units built prior to 1940.
- dis: weighted mean of distances to five Boston employment centres.
- rad: index of accessibility to radial highways.
- tax: full-value property-tax rate per $10,000.
- ptratio: pupil-teacher ratio by town.
- lstat: percent of households with low socioeconomic status.
- medv: median value of owner-occupied homes in $1000s.

```
set.seed(1)
library(ISLR2)
head(Boston)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98 24.0
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14 21.6
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94 33.4
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33 36.2
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21 28.7
```

```
dim(Boston)
```

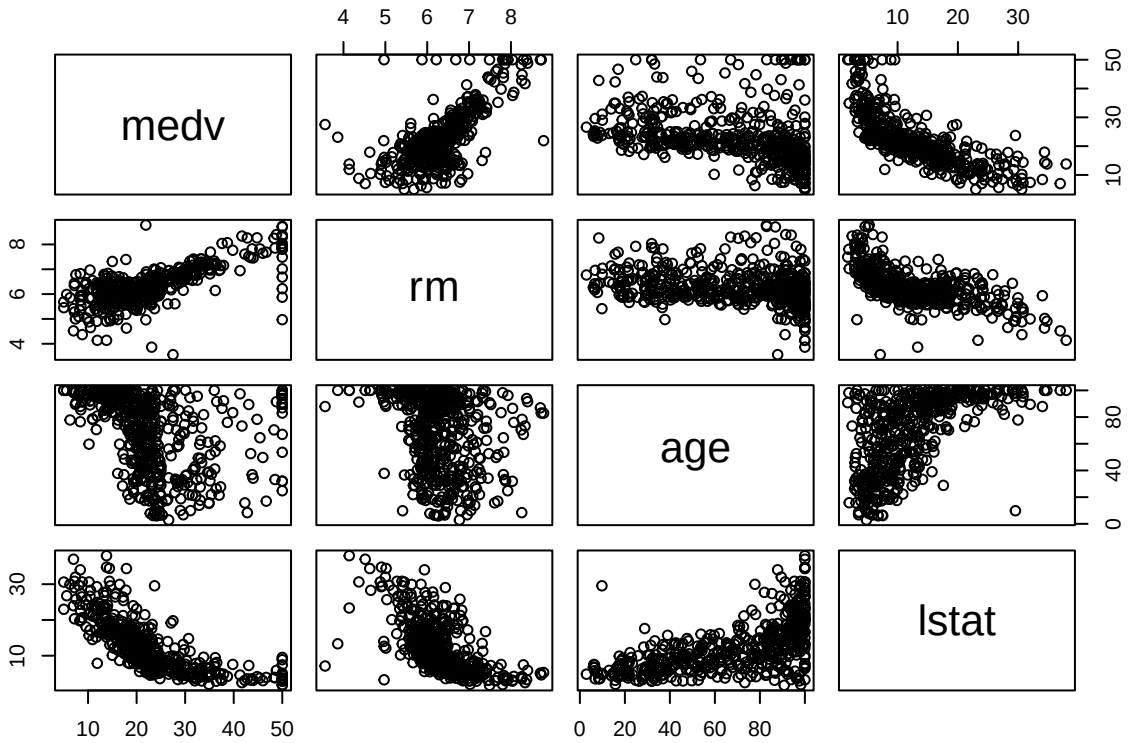
```
## [1] 506  13
```

Q1 (Deviance test, AIC, test MSE)

Our interest is to predict `medv` (median house value) using predictors

- `rm`(average number of rooms per house),
- `age`(proportion of owner-occupied units built prior to 1940) and
- `lstat`(percent of households with low socioeconomic status).

```
pairs(~ medv + rm + age + lstat,  data = Boston)
```



We fit the following models:

```
m1 : medv ~ rm  + lstat,
m2 : medv ~ rm  + poly(lstat, df=2)
m3 : medv ~ rm + age + lstat
m4 : medv ~ rm + age + poly(lstat, df=2)
```

(a) **(10 marks)** Fit the model and use `anova()` function to do the deviance test to compare the models. Choose the best model.

```
m1 <- lm(medv ~  rm  + lstat,  data = Boston)
m2 <- lm(medv ~ rm  + poly(lstat, df=2),  data = Boston)
m3 <- lm(medv ~ rm + age + lstat,  data = Boston)
m4 <- lm(medv ~ rm + age + poly(lstat, df=2),  data = Boston)

anova(m1, m2)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ rm + lstat
## Model 2: medv ~ rm + poly(lstat, df = 2)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    503 15439
## 2    502 12684  1    2755.6 109.06 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m3, m4)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ rm + age + lstat
## Model 2: medv ~ rm + age + poly(lstat, df = 2)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    502 15419
## 2    501 12231  1    3188.1 130.59 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For both comparison `m1 vs m2` and `m3 vs m4`, the second degree polynomial `poly(lstat, df=2)` significantly improve the fit compared to `lstat`.

```
anova(m2, m4)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ rm + poly(lstat, df = 2)
## Model 2: medv ~ rm + age + poly(lstat, df = 2)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    502 12684
## 2    501 12231  1    452.74 18.545 1.997e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Given that the model includes the second degree polynomial `poly(lstat, df=2)`, addition of `age` improve the fit significantly.

The deviance tests imply the model `m4` is the best model.

(b) **(5 marks)** Calculate `AIC` for each model fitted in (a). Choose the best model using the value of `AIC`.

```
AIC(m1,m2,m3,m4)
```

```
##    df      AIC
## m1  4 3173.542
## m2  5 3076.065
## m3  5 3174.880
## m4  6 3059.673
```

The model `m4` has the smallest AIC. `m4` is the best model.

(c) **(10 marks)** Split the data set (100%) into a training set (80%) and a test set (20%). Then fit model1–model5 on the training set, and calculate the test MSE for each model. Choose the best model.

```
set.seed(1)
#set.seed(12420352)
n <- dim(Boston)[1]
train_index <- sample(1:n, n*0.8)
train <- Boston[train_index,]
test <- Boston[-train_index,]

dim(train)
```

```
## [1] 404  13
```

```
dim(test)
```

```
## [1] 102  13
```

```
m1 <- lm(medv ~  rm  + lstat,  data = train)
m2 <- lm(medv ~ rm  + poly(lstat, df=2),  data = train)
m3 <- lm(medv ~ rm + age + lstat,  data = train)
m4 <- lm(medv ~ rm + age + poly(lstat, df=2),  data = train)

mse1 <- mean((test$medv - predict(m1, test))^2)
mse2 <- mean((test$medv - predict(m2, test))^2)
mse3 <- mean((test$medv - predict(m3, test))^2)
mse4 <- mean((test$medv - predict(m4, test))^2)

test_mse <- c(mse1,mse2,mse3,mse4)
test_mse
```

```
## [1] 26.02175 24.20456 26.28640 24.59196
```

The model `m2` has the smallest test MSE. `m2` is the best model.

(d) **(10 marks)** By combining the result from (a), (b) and (c), decide the best model. Refit the chosen model using all of the `Boston` data set. Make a prediction of `medv` for a suburb with values `rm=10`, `age=50` and `lstat=10`. Interpret the predicted value.

I choose model 4 is the best model. Reason:

- (1) anova(m2,m4) given highly significant effect of AGE.
- (2) m4 has the smallest AIC value.
- (3) test MSE for m2 and m4 are different by about 0.3 which is not big (this difference may be due to random split of test and training data).

Combining these I would keep AGE in the model and choose m4 over m2.

```
m4_all <- lm(medv ~ rm + age + poly(lstat, df=2),  data = Boston)

new_dat <- data.frame(rm=c(10), age=c(50), lstat=c(10))
medv_hat <- predict(m4_all,  new_dat)
medv_hat
```

```
##        1
## 36.70091
```

The predicted value of the median house value for a suburb with

- average number of rooms per house = 10,
- proportion of owner-occupied units built prior to 1940 = 50 percent,
- percent of households with low socioeconomic status is 10 percent

is 3.6700912×10^4 USD.

Q2 (LASSO, best subset selection)

We continue to work on `Boston` data set. The aim in Q2 is to predict `medv` (median house value) using all predictors in `Boston` data set. In the following questions, we apply LASSO and the best subset selection methods.

(a) **(10 marks)** (LASSO) Fit a lasso model on the training set, with λ chosen by cross-validation with the `1 se rule` . Report the test error obtained, along with the values of non-zero coefficient estimates. We use the training set and the test set created in Q1 (c).

```
x_train <- model.matrix(medv ~ ., data=train)[,-1]
x_test <- model.matrix(medv ~ ., data=test)[,-1]
```

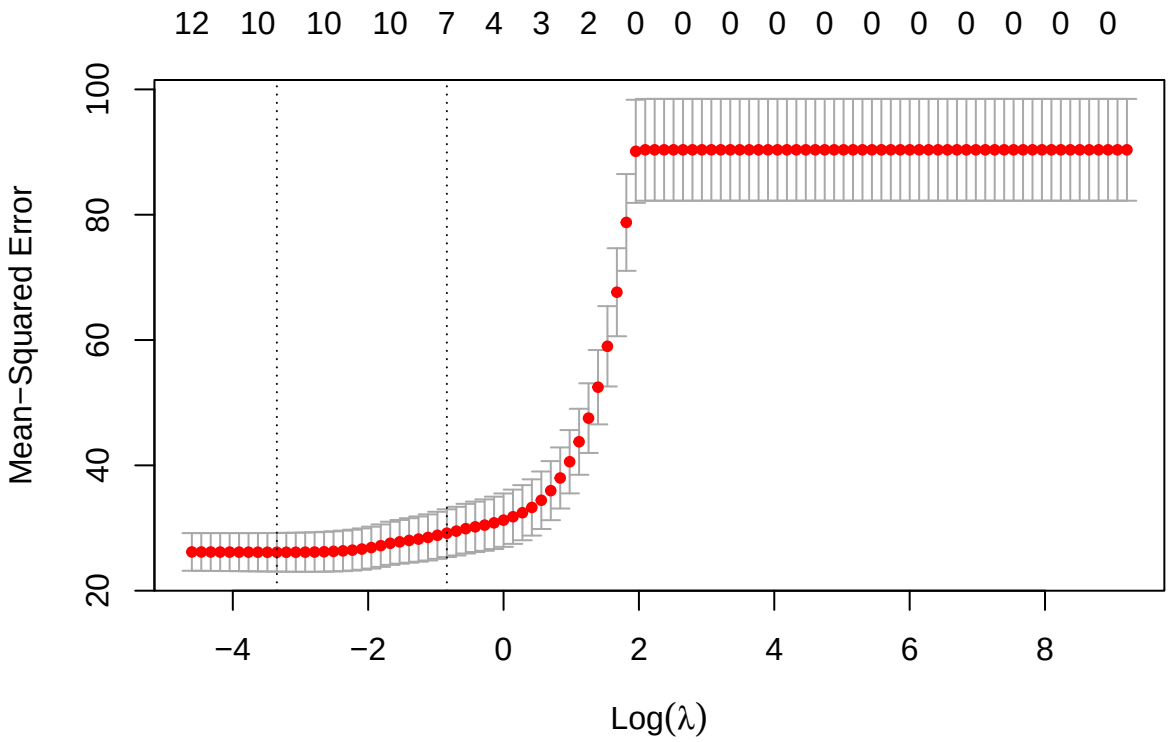
```
set.seed(1)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-7
```

```
grid <- 10^seq(4,-2, length=100)

cv.lasso <- cv.glmnet(x_train, train$medv, alpha=1, lambda=grid,  thresh=1e-12)
plot(cv.lasso)
```



- lambda.1se

```
lam1se <- cv.lasso$lambda.1se
log(lam1se)
```

```
## [1] -0.8373037
```

- Test MSE

```
mean((test$medv - predict(cv.lasso, s=lam1se, newx=x_test ))^2)
```

```
## [1] 19.90622
```

- Non-zero coefficient estimates

```
predict(cv.lasso, s=lam1se, type="coefficients")
```

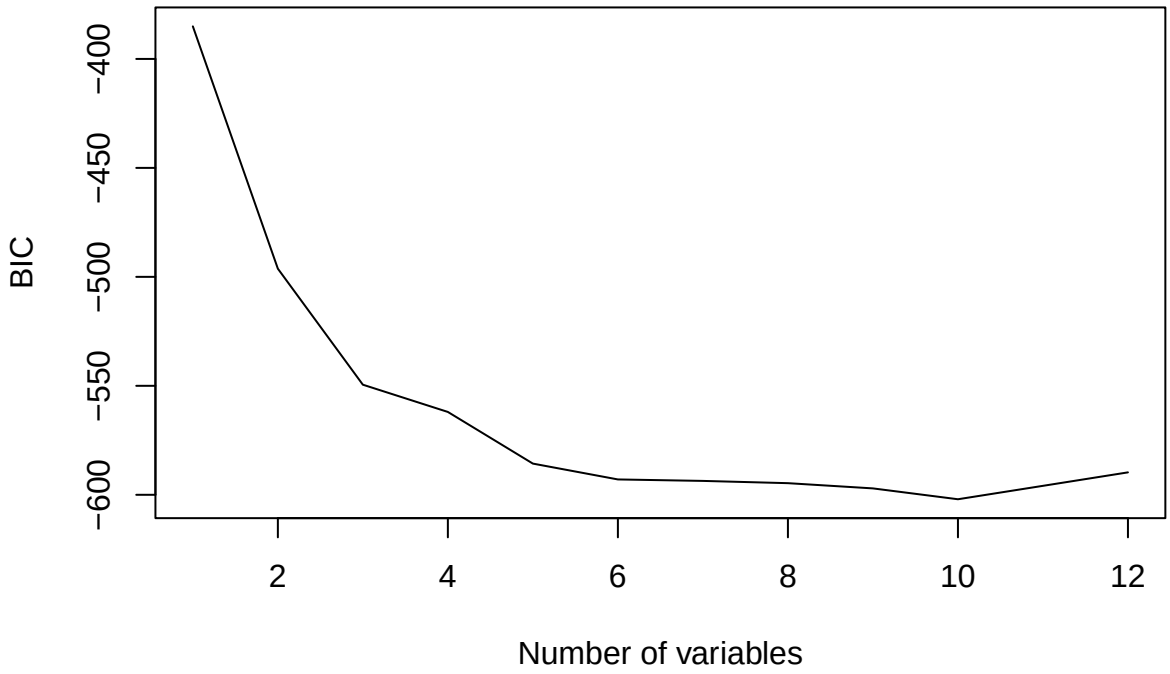
```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept) 15.5316765075
## crim        -0.0225734399
## zn           .
## indus        .
## chas         2.5534277612
## nox          .
## rm           4.4468345467
## age          .
## dis         -0.2056772096
## rad          .
## tax         -0.0008418822
## ptratio     -0.6800720658
## lstat       -0.5660710068
```

(b) **(10 marks)** (Best subset selection) Do the best subset selection with `BIC` and choose the best model. Report the values of coefficient estimates in the best model.

```
library(leaps)

m_best <- regsubsets(medv ~ ., data=Boston, nvmax=15)
m_best_summary <- summary(m_best)

plot(m_best_summary$bic, xlab="Number of variables", ylab="BIC", type="l")
```



```
which.min(m_best_summary$bic)
```

```
## [1] 10
```

```
coef(m_best, 10)
```

```
##  (Intercept)         crim           zn         chas          nox           rm
##  41.45174748  -0.12166488   0.04619119   2.87187265 -18.26242664   3.67295747
##          dis          rad          tax      ptratio        lstat
##  -1.51595105   0.28393226  -0.01229150  -0.93096144  -0.54650916
```

(c) **(10 marks)** Comparing the LASSO chosen model and the best subset selected model, which is the better model? Explain why?

That depends on what we want for the model. If we prefer the simple model, then we choose LASSO. If we prefer unbiased estimates, we choose the best subset selected model.

(d) **(10 marks)** How can you improve the fit of the best subset selected model?

One possible improvement is suggested in Q1. We replace `lstat` with its 2nd degree polynomial `poly(lstat, df=2)`.

```
m_best <- lm(medv ~ crim + tax + zn + chas + ptratio +
               lstat + nox  + rm  + dis + rad, data=Boston)
m_imp <- lm(medv ~ crim + tax + zn + chas  + ptratio +
               poly(lstat, df=2) + nox  + rm  + dis + rad, data=Boston)

BIC(m_best, m_imp)
```

```
##        df      BIC
## m_best 12 3084.662
## m_imp  13 2990.892
```

As BIC get smaller, the modification improved the fit of the model.

[**Total: 75 marks**]