Assignment 5

Tina Roha

4/08/2021

Linear Model Selection and Regularization

Chapter 06 (page 259): 2, 9 and 11

Question-2

2. For parts (a) through (c), indicate which of i. through iv. is correct. Justify your answer.

i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.
ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.
iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.
iv. Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

a. The lasso, relative to least squares, is:

For the lasso, relative to least squares, the correct answer is iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance. The lasso has two major advantages over least squares. The first advantage stems from the variance-bias trade-off. Furthermore, when the least squares estimate produces a massive variance result, the lasso solution allows the opportunity to reduce the variance by slightly increasing the overall bias. By doing so, the ability to produce predictions with accuracy enhances. The second advantage of the lasso is its variable selection which allows the overall interpretation to be understood with ease.

b. The ridge regression, relative to least squares, is:

For the ridge regression, relative to least squares, the correct answer is iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance. The advantage the ridge regression has over least squares stems from the variance-bias trade-off. Furthermore, similarly to the lasso, the ridge regression also allows the opportunity to reduce the high variance value by slightly increasing the overall bias. Due to this advantage, ridge regression is most commonly used in the context of multicollinearity.

c. The non-linear methods, relative to least squares, is:

For the non-linear methods, relative to least squares, the correct answer is ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

Question-9

9. In this exercise, we will predict the number of applications received using the other variables in the College data set.

a. Split the data set into a training set and a test set.

```
library(ISLR)
data(College)
set.seed(11)
train = sample(1:dim(College)[1], dim(College)[1] / 2)
test <- -train
College.train <- College[train, ]
College.test <- College[test, ]
```

b. Fit a linear model using least squares on the training set, and report the test error obtained.

```
fit.lm <- lm(Apps ~ ., data = College.train)
pred.lm <- predict(fit.lm, College.test)
mean((pred.lm - College.test$Apps)^2)
```

```
## [1] 1538442
```

The test error, MSE, obtained in the final output is 1538442.

c. Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.

```
train.mat <- model.matrix(Apps ~ ., data = College.train)
test.mat <- model.matrix(Apps ~ ., data = College.test)
grid <- 10 ^ seq(4, -2, length = 100)
fit.ridge <- glmnet(train.mat, College.train$Apps, alpha = 0, lambda = grid, thresh = 1e-12)
cv.ridge <- cv.glmnet(train.mat, College.train$Apps, alpha = 0, lambda = grid, thresh = 1e-12)
bestlam.ridge <- cv.ridge$lambda.min
bestlam.ridge
```

```
## [1] 18.73817
```

```
pred.ridge <- predict(fit.ridge, s = bestlam.ridge, newx = test.mat)
mean((pred.ridge - College.test$Apps)^2)
```

```
## [1] 1608859
```

The test error obtained is 1608859. This value is higher for ridge regression when compared to least squares.

d. Fit a lasso model on the training set, with λ chosen by crossvalidation. Report the test error obtained, along with the number of non-zero coefficient estimates.

```
fit.lasso <- glmnet(train.mat, College.train$Apps, alpha = 1, lambda = grid, thresh = 1e-12)
cv.lasso <- cv.glmnet(train.mat, College.train$Apps, alpha = 1, lambda = grid, thresh = 1e-12)
bestlam.lasso <- cv.lasso$lambda.min
bestlam.lasso
```

```
## [1] 21.54435
```

```
pred.lasso <- predict(fit.lasso, s = bestlam.lasso, newx = test.mat)
mean((pred.lasso - College.test$Apps)^2)
```

```
## [1] 1635280
```

```
predict(fit.lasso, s = bestlam.lasso, type = "coefficients")
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept) -836.50402310
## (Intercept)    .
## PrivateYes  -385.73749394
## Accept         1.17935134
## Enroll         .
## Top10perc     22.70211938
## Top25perc      .
## F.Undergrad    0.07062149
## P.Undergrad    0.01366763
## Outstate      -0.03424677
## Room.Board     0.01281659
## Books         -0.02167770
## Personal       .
## PhD           -1.46396964
## Terminal      -5.17281004
## S.F.Ratio      5.70969524
## perc.alumni   -9.95007567
## Expend         0.14852541
## Grad.Rate      5.79789861
```

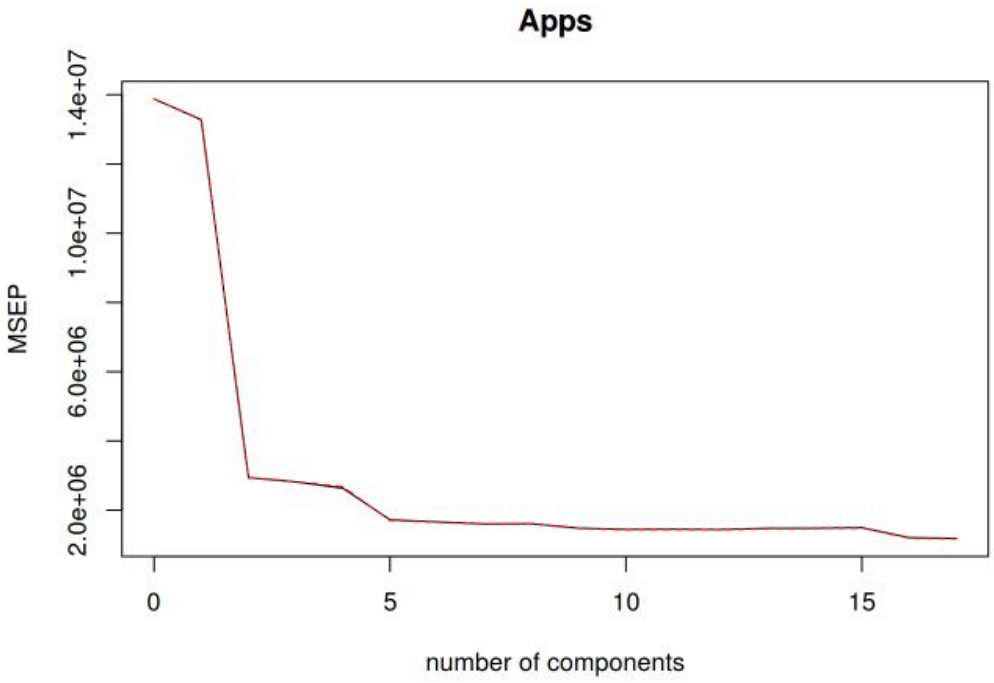
The test error obtained is 1635280 and therefore we can conclude that the lasso model is not as efficient at predicting when compared to the ridge regression with this dataset.

e. Fit a PCR model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.

```
library(pls)
```

```
##
## Attaching package: 'pls'
##
## The following object is masked from 'package:stats':
##
##     loadings
```

```
fit.pcr <- pcr(Apps ~ ., data = College.train, scale = TRUE, validation = "CV")
validationplot(fit.pcr, val.type = "MSEP")
```



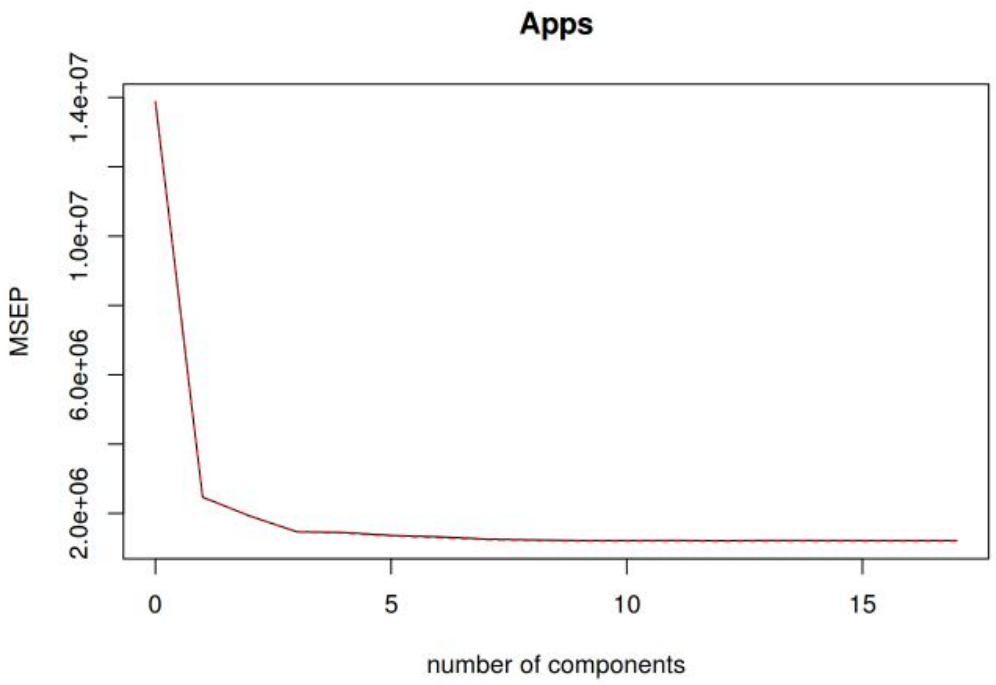

```
pred.pcr <- predict(fit.pcr, College.test, ncomp = 10)
mean((pred.pcr - College.test$Apps)^2)
```

```
## [1] 3014496
```

The test error obtained is 3014496. Additionally, the test MSE is also higher for PCR than for least squares.

f. Fit a PLS model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.

```
fit.pls <- plsr(Apps ~ ., data = College.train, scale = TRUE, validation = "CV")
validationplot(fit.pls, val.type = "MSEP")
```




```
pred.pls <- predict(fit.pls, College.test, ncomp = 10)
mean((pred.pls - College.test$Apps)^2)
```

```
## [1] 1508987
```

The test error obtained is 1508987. In this case, the test MSE is lower for partial least squares (PLS) when compared to least squares.

g. Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these five approaches?

```
test.avg <- mean(College.test$Apps)
lm.r2 <- 1 - mean((pred.lm - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
ridge.r2 <- 1 - mean((pred.ridge - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
lasso.r2 <- 1 - mean((pred.lasso - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
pcr.r2 <- 1 - mean((pred.pcr - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
pls.r2 <- 1 - mean((pred.pls - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
```

From the results obtained the r-squared values for least squares, ridge regression, the lasso, PCR and PLS is 0.9044281, 0.9000536, 0.8984123, 0.8127319 and 0.9062579 respectively. Therefore, it can be concluded that all of the models, excluding PCR, have high accuracy when predicting the number of college applications received.

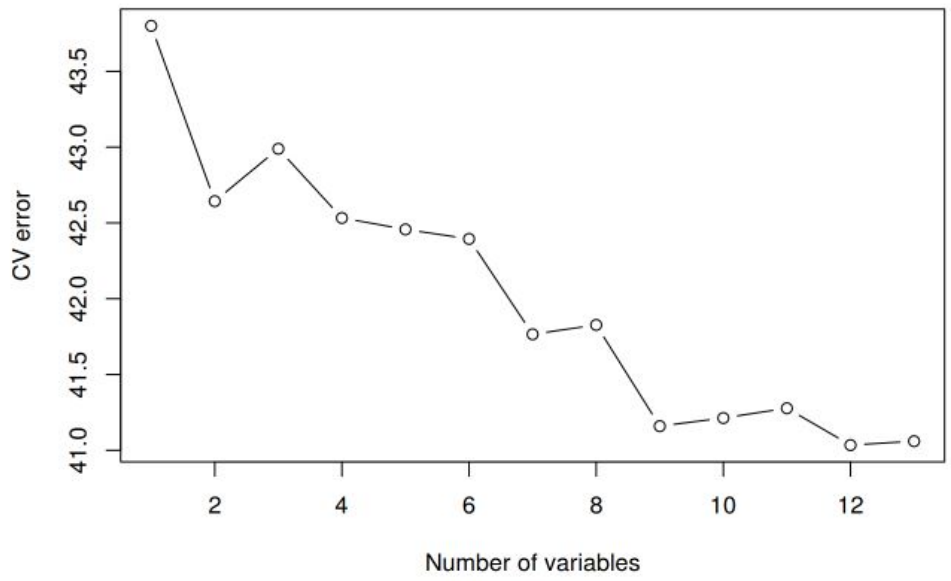
Question-11

11. We will now try to predict per capita crime rate in the Boston data set.

a. Try out some of the regression methods explored in this chapter, such as best subset selection, the lasso, ridge regression, and PCR. Present and discuss results for the approaches that you consider.

i. Regression method: The best subset selection

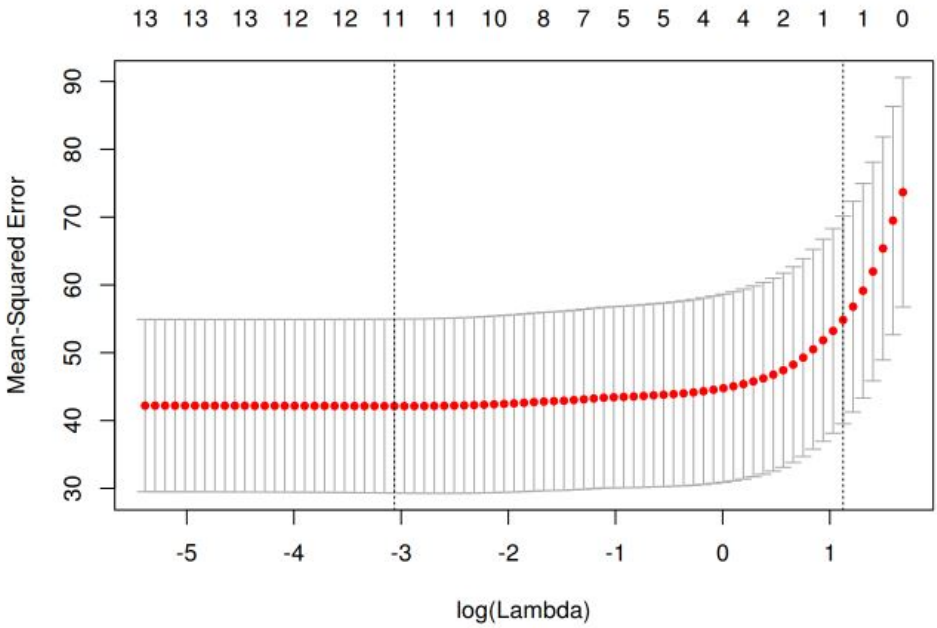
```
library(MASS)
data(Boston)
set.seed(1)

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
}

k = 10
folds <- sample(1:k, nrow(Boston), replace = TRUE)
cv.errors <- matrix(NA, k, 13, dimnames = list(NULL, paste(1:13)))
for (j in 1:k) {
    best.fit <- regsubsets(crim ~ ., data = Boston[folds != j, ], nvmax = 13)
    for (i in 1:13) {
        pred <- predict(best.fit, Boston[folds == j, ], id = i)
        cv.errors[j, i] <- mean((Boston$crim[folds == j] - pred)^2)
    }
}
mean.cv.errors <- apply(cv.errors, 2, mean)
plot(mean.cv.errors, type = "b", xlab = "Number of variables", ylab = "CV error")
```



The best subset selection regression method produced an output with a CV estimate for the test MSE equal to 41.0345657.

ii. Regression method: The lasso

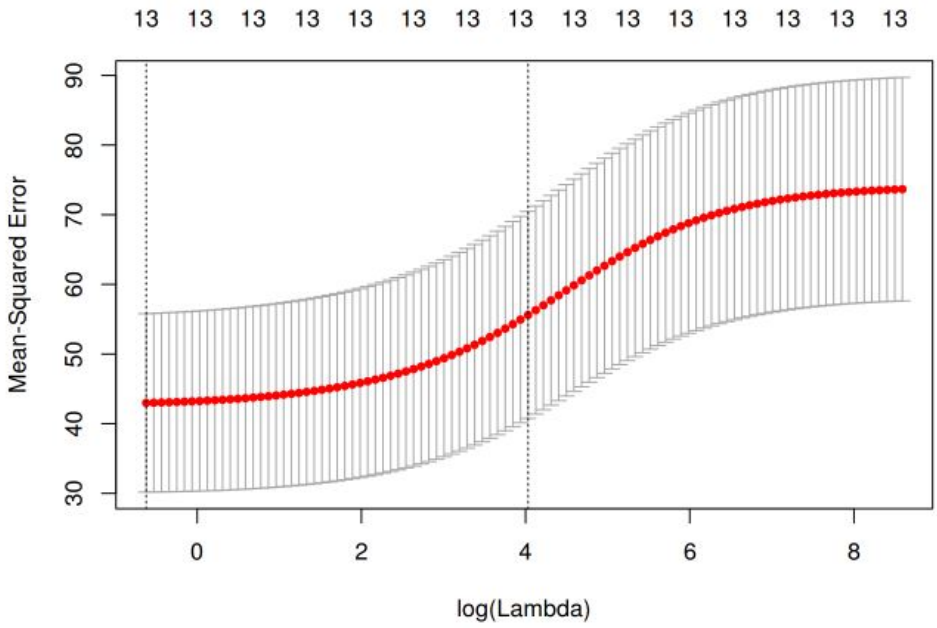
```
x <- model.matrix(crim ~ ., Boston)[, -1]
y <- Boston$crim
cv.out <- cv.glmnet(x, y, alpha = 1, type.measure = "mse")
plot(cv.out)
```



The lasso regression method produced an output with a CV estimate for the test MSE equal to 42.134324.

iii. Regression method: Ridge regression

```
cv.out <- cv.glmnet(x, y, alpha = 0, type.measure = "mse")
plot(cv.out)
```



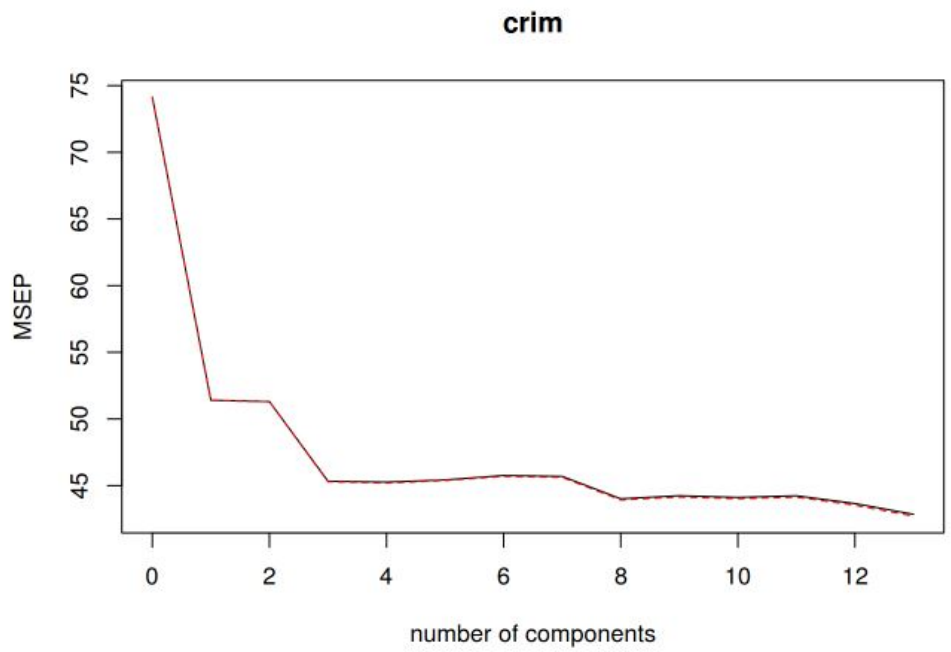
The ridge regression method produced an output with a CV for the test MSE equal to 42.9834518.

iv. Regression method: PCR

```
pcr.fit <- pcr(crim ~ ., data = Boston, scale = TRUE, validation = "CV")
summary(pcr.fit)
```

```
## Data:    X dimension: 506 13
##  Y dimension: 506 1
## Fit method: svdpc
## Number of components considered: 13
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            8.61    7.170    7.163    6.733    6.728    6.740    6.765
## adjCV         8.61    7.169    7.162    6.730    6.723    6.737    6.760
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       6.760    6.634    6.652     6.642     6.652     6.607     6.546
## adjCV    6.754    6.628    6.644     6.635     6.643     6.598     6.536
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X       47.70    60.36    69.67    76.45    82.99    88.00    91.14
## crim    30.69    30.87    39.27    39.61    39.61    39.86    40.14
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## X       93.45    95.40     97.04     98.46     99.52     100.0
## crim    42.47    42.55     42.78     43.04     44.13      45.4
```

```
validationplot(pcr.fit, val.type = "MSEP")
```



The PCR regression method produced an output with a CV estimate for the test MSE equal to 45.693568.

b. Propose a model (or set of models) that seem to perform well on this data set, and justify your answer. Make sure that you are evaluating model performance using validation set error, crossvalidation, or some other reasonable alternative, as opposed to using training error.

Considering the overall evaluation of model performance it can be concluded that the model that performs well on this data set is the best subset selection method. Furthermore, this is due to the fact that this method produced the lowest cross-validation error.

c. Does your chosen model involve all of the features in the data set? Why or why not?

The chosen model, the best subset selection method, does not involve all of the features in the data set. Furthermore, this model only has 13 predictors so that less variation is present. Overall, the goal is to have accurate model prediction with low variance as well as low MSE.