Modern Data Mining - HW 2

Anirudh Bajaj
Esther Shin
Matt LeBaron

Overview / Instructions

This is homework #2 of STAT 471/571/701. It will be **due on Oct 10, 2018 by 11:59 PM** on Canvas. You can directly edit this file to add your answers. Submit the Rmd file, a PDF or word or HTML version with only 1 submission per HW team.

Problem 0

Review the code and concepts covered during lecture: multiple regression, model selection and penalized regression through elastic net.

Problem 1

Do ISLR, page 262, problem 8 only part (a) to (d) and write up the answer here. This question is designed to help us understanding model selections through simulations. (e) Describe as accurate as possible what Cp and BIC are estimating?

(a)

```
# Use rnorm() to generate a predictor X of length n = 100, and a noise vector of length n = 100
x <- rnorm(100)
noise <- rnorm(100)
```

(b)

```
# Generate a response vector Y of length n = 100 according to the model
y <- 1 + 2*x + 3*x^2 + 4*x^3 + noise
```

(c)

```
# Create the predictors x^2 through x^10
x2 <- x^2
x3 <- x^3
x4 <- x^4
x5 <- x^5
x6 <- x^6
x7 <- x^7
x8 <- x^8
x9 <- x^9
x10 <- x^10

# Use the regsubsets() function to perform best subset selection
new_data <- data.frame(x,x2,x3,x4,x5,x6,x7,x8,x9,x10,noise,y)
new_subset <- regsubsets(y ~ x+x2+x3+x4+x5+x6+x7+x8+x9+x10,data = new_data, nvmax = 10)
new_summary <- summary(new_subset)
new_summary$which
```

```
##    (Intercept)     x    x2    x3    x4    x5    x6    x7    x8    x9   x10
## 1         TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2         TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3         TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 4         TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 5         TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
## 6         TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE
## 7         TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## 8         TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 9         TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 10        TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
data.frame(variables=(1:length(new_summary$rsq)), r_squared=new_summary$rsq)
```

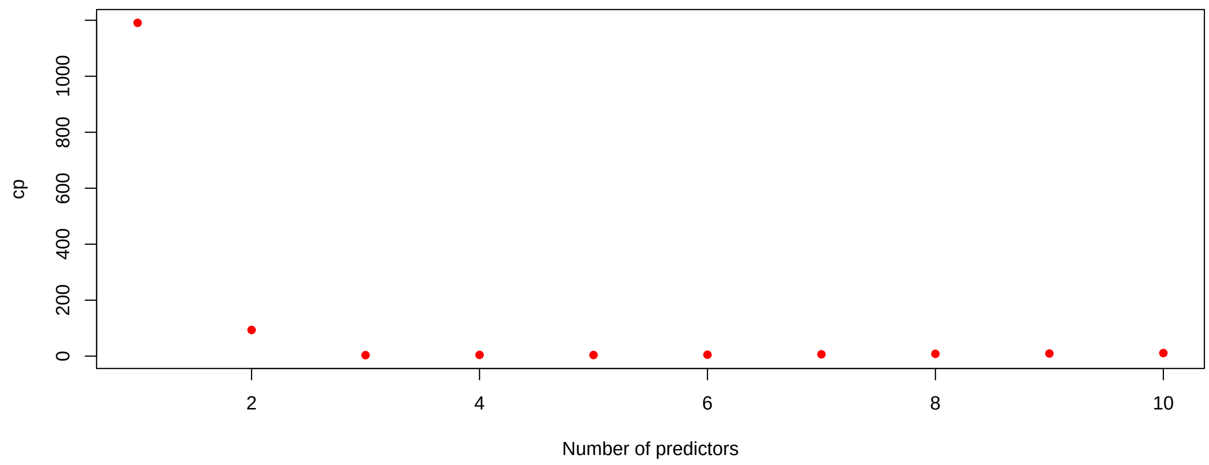
```
##    variables r_squared
## 1          1 0.9539451
## 2          2 0.9932856
## 3          3 0.9965803
## 4          4 0.9966217
## 5          5 0.9967046
## 6          6 0.9967374
## 7          7 0.9967582
## 8          8 0.9967705
## 9          9 0.9968003
## 10        10 0.9968148
```

```
# What is the best model obtained?
data.frame(variables = (1:length(new_summary$rsq)),
           r_squared = new_summary$rsq,
           rss = new_summary$rss,
           bic = new_summary$bic,
           cp = new_summary$cp)
```

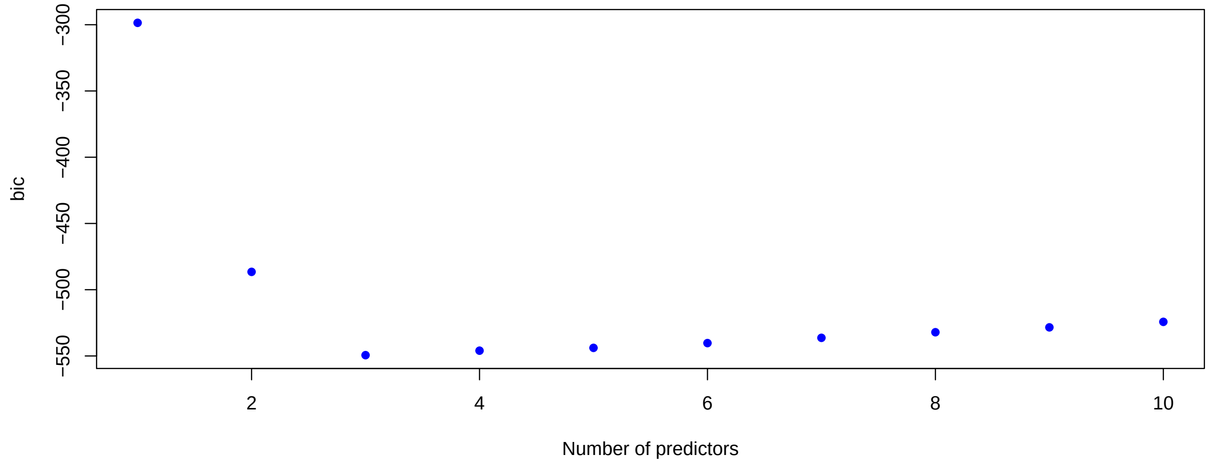
```
##    variables r_squared       rss       bic          cp
## 1          1 0.9539451 1889.1417 -298.5817 1190.846312
## 2          2 0.9932856  275.4192 -486.5349   93.610132
## 3          3 0.9965803  140.2722 -549.4007    3.550694
## 4          4 0.9966217  138.5736 -546.0138    4.393643
## 5          5 0.9967046  135.1737 -543.8928    4.077709
## 6          6 0.9967374  133.8278 -540.2883    5.160885
## 7          7 0.9967582  132.9774 -536.3206    6.581579
## 8          8 0.9967705  132.4701 -532.0977    8.236023
## 9          9 0.9968003  131.2504 -528.4175    9.405222
## 10        10 0.9968148  130.6555 -524.2666   11.000000
```

The best model obtained has three variables: x, x2, and x3. This makes sense because these three variables were used to generate Y in the first place, so they should be most predictive, although they aren't perfect because we added in noise as well.

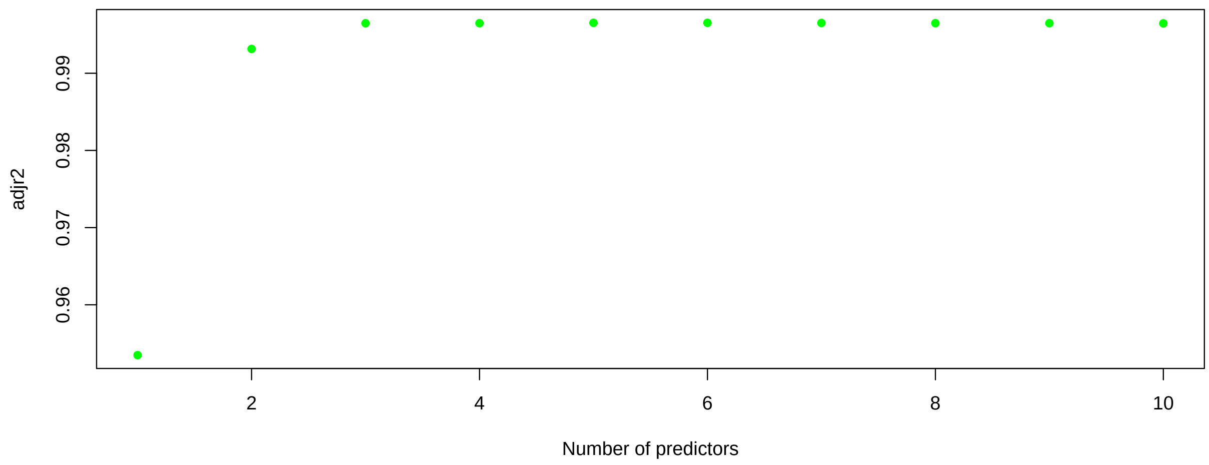
```
# Comparing the plots for Cp, bic, and r2
plot(new_summary$cp, xlab="Number of predictors",
     ylab="cp", col="red", type="p", pch=16)
```



```
plot(new_summary$bic, xlab="Number of predictors",
     ylab="bic", col="blue", type="p", pch=16)
```



```
plot(new_summary$adjr2, xlab="Number of predictors",
     ylab="adjr2", col="green", type="p", pch=16)
```



```r
# Find the optimal model size for Cp
opt.size <- which.min(new_summary$cp)
opt.size
```

```
## [1] 3
```

```r
# Find the optimal model size for bic
opt.size <- which.min(new_summary$bic)
opt.size
```

```
## [1] 3
```

```r
# Find the optimal model size for r2
opt.size <- which.min(new_summary$rsq)
opt.size
```

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

```r
# Report the coefficients of the best model obtained
coef(new_subset, 3)
```

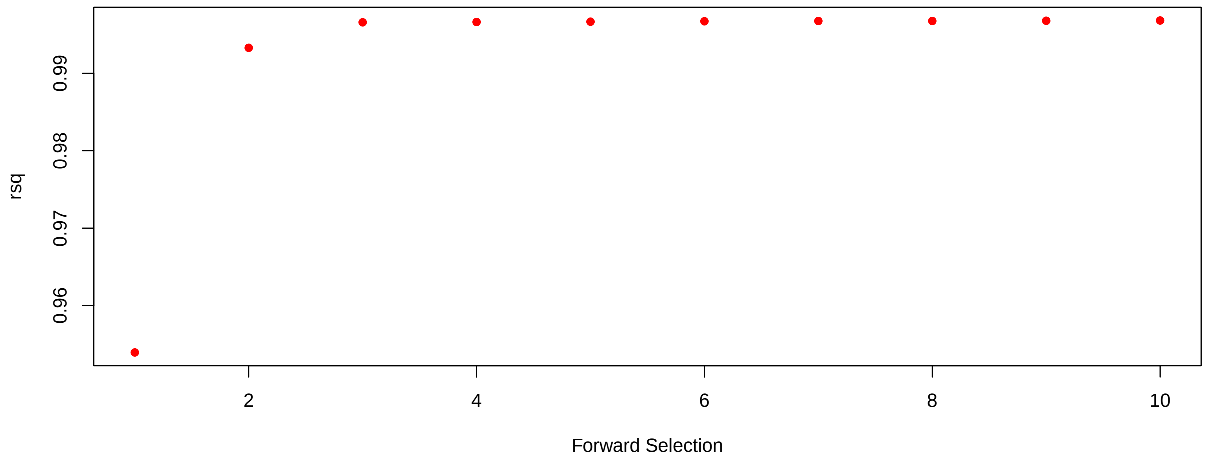
```
## (Intercept)           x          x2          x3
##    1.135632    1.966578    2.933214    4.040205
```

(d)

```r
# Repeat using forward selection
forward <- regsubsets(y ~ x+x2+x3+x4+x5+x6+x7+x8+x9+x10,data = new_data, nvmax = 10, method = "forward"
forward_sum <- summary(forward)
forward_sum
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ x + x2 + x3 + x4 + x5 + x6 + x7 + x8 +
##     x9 + x10, data = new_data, nvmax = 10, method = "forward")
## 10 Variables  (and intercept)
##     Forced in Forced out
## x       FALSE      FALSE
## x2      FALSE      FALSE
## x3      FALSE      FALSE
## x4      FALSE      FALSE
## x5      FALSE      FALSE
## x6      FALSE      FALSE
## x7      FALSE      FALSE
## x8      FALSE      FALSE
## x9      FALSE      FALSE
## x10     FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: forward
##            x   x2  x3  x4  x5  x6  x7  x8  x9  x10
## 1  ( 1 )  " " " " "*" " " " " " " " " " " " " " "
## 2  ( 1 )  " " "*" "*" " " " " " " " " " " " " " "
## 3  ( 1 )  "*" "*" "*" " " " " " " " " " " " " " "
## 4  ( 1 )  "*" "*" "*" " " "*" " " " " " " " " " "
## 5  ( 1 )  "*" "*" "*" "*" "*" " " " " " " " " " "
## 6  ( 1 )  "*" "*" "*" "*" "*" "*" " " " " " " " "
## 7  ( 1 )  "*" "*" "*" "*" "*" "*" "*" " " " " " "
## 8  ( 1 )  "*" "*" "*" "*" "*" "*" "*" "*" " " " "
## 9  ( 1 )  "*" "*" "*" "*" "*" "*" "*" "*" " " "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

```
plot(forward_sum$rsq, ylab="rsq", col="red", type="p", pch=16,
     xlab="Forward Selection")
```



```
# Find the optimal model size for forward selection
opt.size <- which.min(forward_sum$rsq)
opt.size
```

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

```
# Report the coefficients for forward selection
coef(forward, 3)
```

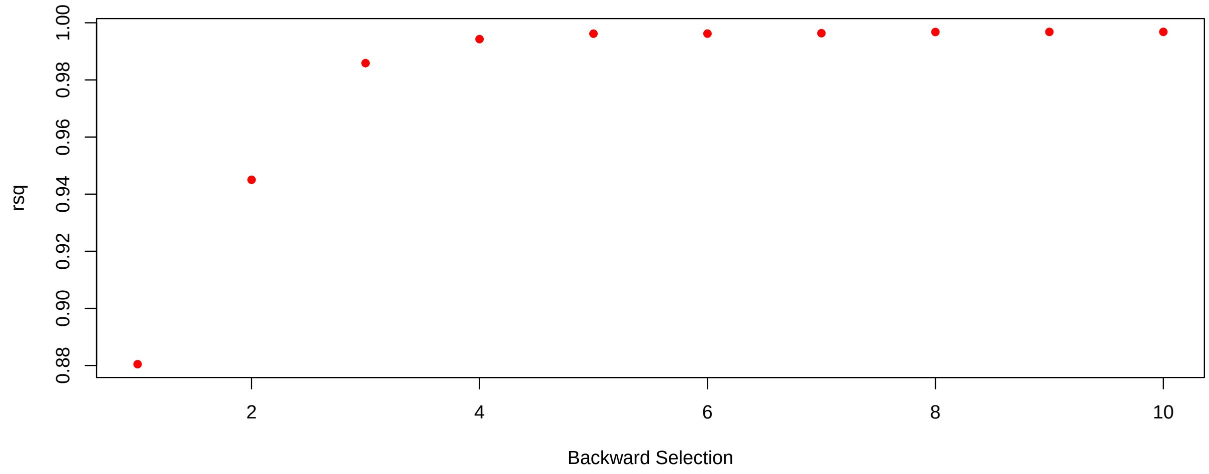
```
## (Intercept)           x          x2          x3
##    1.135632    1.966578    2.933214    4.040205
```

```
# Repeat using backward selection
backward <- regsubsets(y ~ x+x2+x3+x4+x5+x6+x7+x8+x9+x10,data = new_data, nvmax = 10, method = "backwar
backward_sum <- summary(backward)
```

```
backward_sum
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ x + x2 + x3 + x4 + x5 + x6 + x7 + x8 + 
##     x9 + x10, data = new_data, nvmax = 10, method = "backward")
## 10 Variables  (and intercept)
##     Forced in Forced out
## x       FALSE      FALSE
## x2      FALSE      FALSE
## x3      FALSE      FALSE
## x4      FALSE      FALSE
## x5      FALSE      FALSE
## x6      FALSE      FALSE
## x7      FALSE      FALSE
## x8      FALSE      FALSE
## x9      FALSE      FALSE
## x10     FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: backward
##           x   x2  x3  x4  x5  x6  x7  x8  x9  x10
## 1  ( 1 )  " " " " " " " " "*" " " " " " " " " " "
## 2  ( 1 )  "*" " " " " " " "*" " " " " " " " " " "
## 3  ( 1 )  "*" "*" " " " " "*" " " " " " " " " " "
## 4  ( 1 )  "*" "*" " " " " "*" " " "*" " " " " " "
## 5  ( 1 )  "*" "*" " " " " "*" " " "*" " " "*" " "
## 6  ( 1 )  "*" "*" " " " " "*" "*" "*" " " "*" " "
## 7  ( 1 )  "*" "*" " " " " "*" "*" "*" "*" "*" " "
## 8  ( 1 )  "*" "*" " " " " "*" "*" "*" "*" "*" "*"
## 9  ( 1 )  "*" "*" "*" " " "*" "*" "*" "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

```
plot(backward_sum$rsq, ylab="rsq", col="red", type="p", pch=16,
     xlab="Backward Selection")
```



```
# Find the optimal model size for backward selection
opt.size <- which.min(backward_sum$rsq)
opt.size
```

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

```
# Report the coefficients for backward selection
coef(backward, 3)
```

```
## (Intercept)           x          x2          x5
##   1.2107862   7.1103432   2.9268606   0.4372269
```

Using forward and backward stepwise selection, we only use one variable (x3) as opposed to three variables (x, x2, x3).

(e)

Describe what Cp and BIC are estimating.

Cp: an unbiased estimator of average prediction errors.

BIC: uses k to estimate the number of parameters. It heavily penalizes additional variables when they are introduced into the model. Therefore, BIC models tend to have fewer variables.

Problem 2:

This will be the last part of the Auto data from ISLR. The original data contains 408 observations about cars. It has some similarity as the data CARS that we use in our lectures. To get the data, first install the package ISLR. The data Auto should be loaded automatically. We use this case to go through methods learnt so far.

You can access the necessary data with the following code:

```
# check if you have ISLR package, if not, install it
if(!requireNamespace('ISLR')) install.packages('ISLR')
auto_data <- ISLR::Auto
```

Final modelling question: we want to explore the effects of each feature as best as possible. You may explore interactions, feature transformations, higher order terms, or other strategies within reason. The model(s) should be as parsimonious (simple) as possible unless the gain in accuracy is significant from your point of view. Use Mallow's Cp or BIC to select the model. * Describe the final model and its accuracy. Include diagnostic plots with particular focus on the model residuals. * Summarize the effects found. * Predict the mpg of a car that is: built in 1983, in US, red, 180 inches long, 8 cylinders, 350 displacement, 260 as horsepower and weighs 4000 pounds. Give a 95% CI.

Problem 3: Lasso

Crime data continuation: We use a subset of the crime data discussed in class, but only look at Florida and California. `crimedata` is available on Canvas; we show the code to clean here.

```
crime <- read.csv("CrimeData.csv", stringsAsFactors = F, na.strings = c("?"))
crime <- dplyr::filter(crime, state %in% c("FL", "CA"))
```

Our goal is to find the factors which relate to violent crime. This variable is included in crime as `crime$violentcrimes.perpop`.

Use LASSO to choose a reasonable, small model. Fit an OLS model with the variables obtained. The final model should only include variables with p-values < 0.05. Note: you may choose to use lambda 1st or lambda min to answer the following questions where apply.

1. What is the model reported by LASSO?
2. What is the model after running OLS?

3. What is your final model, after excluding high p-value variables?

Now, instead of Lasso, we want to consider how changing the value of alpha (i.e. mixing between Lasso and Ridge) will affect the model. Cross-validate between alpha and lambda, instead of just lambda. Note that the final model may have variables with p-values higher than 0.05; this is because we are optimizing for accuracy rather than parsimoniousness.

1. What is your final elastic net model? What were the alpha and lambda values? What is the prediction error?
2. Use the elastic net variables in an OLS model. What is the equation, and what is the prediction error.
3. Summarize your findings, with particular focus on the difference between the two equations.