STAT 4610

Michael Ghattas

16/11/2022

Chapter 6

```
library(ISLR)
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.1.2
```

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

```
## Warning: package 'Matrix' was built under R version 4.1.2
```

```
## Loaded glmnet 4.1-4
```

```
library(pls)
```

```
## Warning: package 'pls' was built under R version 4.1.2
```

```
##
## Attaching package: 'pls'
```

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

Problem - 9

```
attach(College)
```

Part (a)

```
set.seed(1)
train = sample(c(TRUE,FALSE), nrow(College), rep = TRUE)
test = (!train)

College.train = College[train, ]
College.test = College[test, ]
```

Part (b)

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

```
## [1] 984743.1
```

-> Linear model fit test-error = 984743.1

Part (c)

```
set.seed(1)

train.mat = model.matrix(Apps~., data = College.train)
test.mat = model.matrix(Apps~., data = College.test)

cv.out = cv.glmnet(train.mat, College.train$Apps, alpha = 0)
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 394.2365
```

```
ridge.mod = glmnet(train.mat, College.train$Apps, alpha = 0)
ridge.pred = predict(ridge.mod, s = bestlam, newx = test.mat)
mean((ridge.pred - College.test$Apps)^2)
```

```
## [1] 940970.9
```

-> Ridge regression fit test error with a cross-validation based lambda = 940970.9 -> Lower than linear model test error

Part (d)

```
set.seed(1)

cv.out2 = cv.glmnet(train.mat, College.train$Apps, alpha = 1)
bestlam2 = cv.out2$lambda.min
bestlam2
```

```
## [1] 59.92044
```

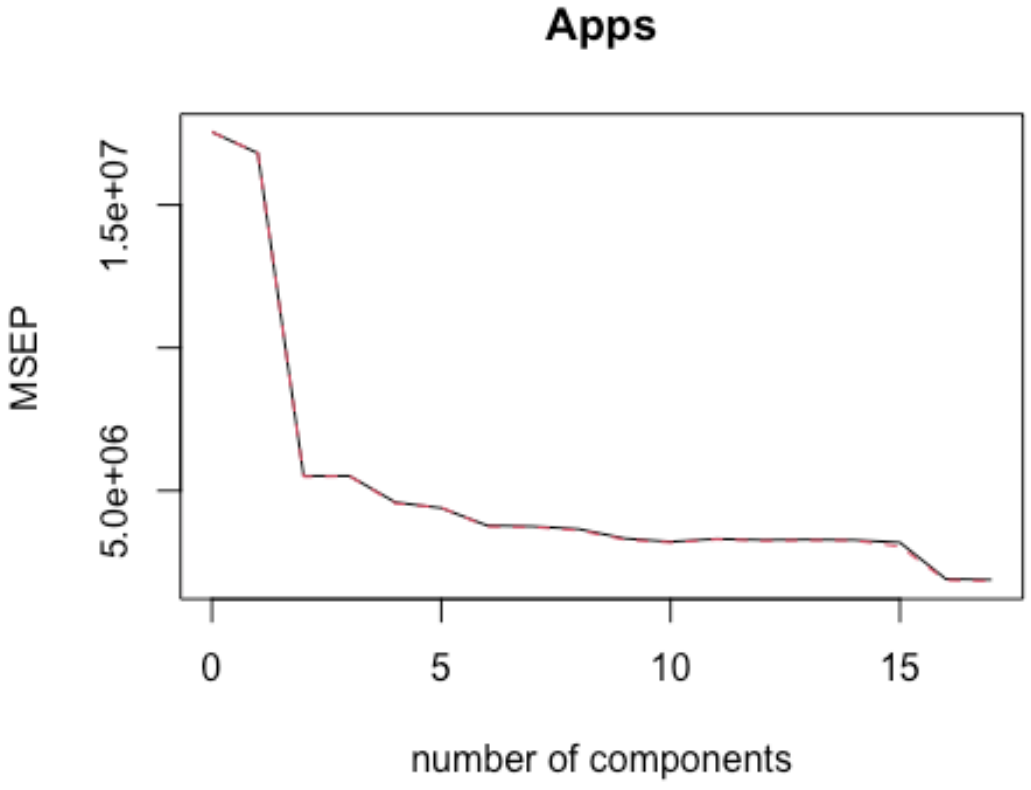
```
lasso.mod = glmnet(train.mat, College.train$Apps, alpha = 1)
lasso.pred = predict(lasso.mod,s = bestlam2, newx = test.mat)
mean((lasso.pred - College.test$Apps)^2)
```

```
## [1] 993741.7
```

-> Lasso model fit test error with a cross-validation based lambda = 993741.7 -> Higher than linear model and ridge regression test error

Part (e)

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



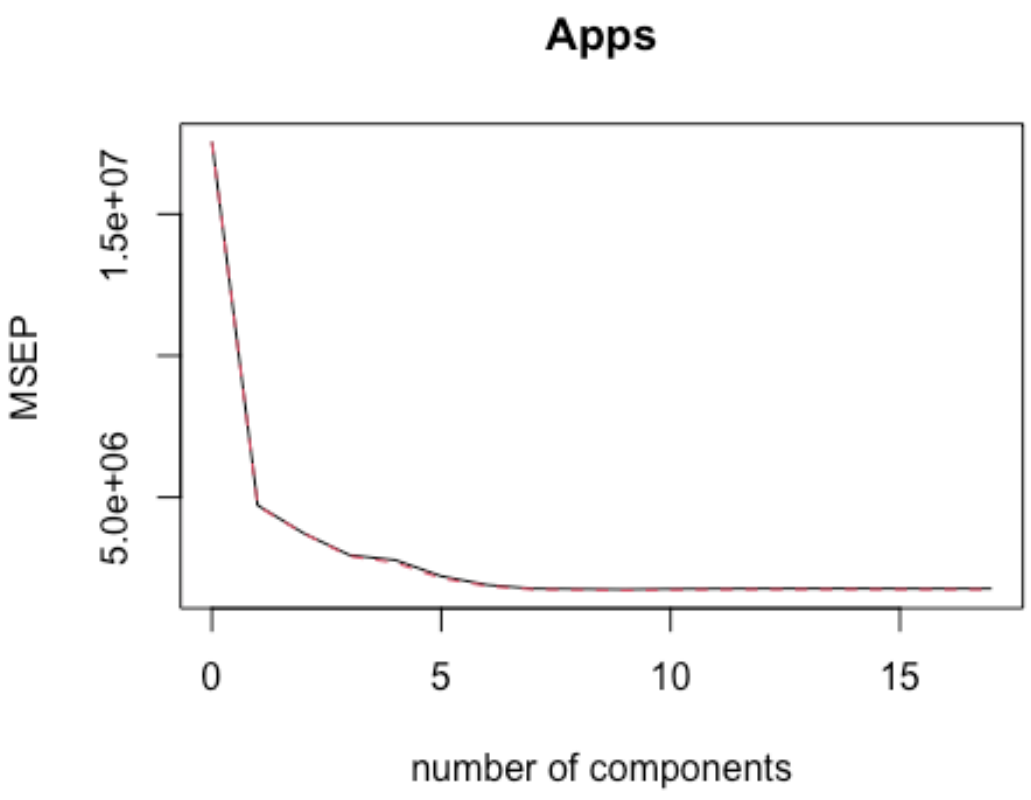

```
summary(pcr.fit)

## Data:    X dimension: 393 17
##  Y dimension: 393 1
## Fit method: svdpc
## Number of components considered: 17
```

```
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            4189     4100     2349     2349     2143     2096     1945
## adjCV         4189     4102     2343     2347     2134     2100     1935
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1937     1912     1824      1793      1820      1810      1814
## adjCV     1931     1899     1812      1785      1815      1802      1806
##        14 comps  15 comps  16 comps  17 comps
## CV         1810      1786      1383      1372
## adjCV      1805      1745      1366      1356
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8
comps
## X      31.858    57.44    64.20    69.91    75.10    80.17    83.82
87.30
## Apps    4.353    70.99    71.18    76.84    78.34    81.03    81.59
82.21
##       9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X       90.26     92.74     94.79     96.70     97.76     98.67     99.37
## Apps    83.31     83.97     83.97     84.34     84.58     84.70     91.28
##       16 comps  17 comps
## X        99.82    100.00
## Apps     92.83     93.02

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

## [1] 1682909
```

-> Lowest MSEP with PCR dimension reduction around M = 10 -> M = 10 has the lowest CV error while accomplishing dimension reduction -> Lasso model fit test error with a cross-validation base lambda = 1682909 -> Higher than all previous models test error

Part (f)

```
set.seed(1)

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




```
summary(pls.fit)

## Data:    X dimension: 393 17
##  Y dimension: 393 1
## Fit method: kernelpls
## Number of components considered: 17
##
## VALIDATION: RMSEP
```

```
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            4189     2172     1932     1720     1669     1489     1382
## adjCV         4189     2163     1930     1709     1640     1463     1365
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1333     1328     1323      1329      1332      1334      1334
## adjCV     1321     1316     1310      1316      1319      1320      1321
##        14 comps  15 comps  16 comps  17 comps
## CV         1335      1333      1333      1333
## adjCV      1321      1320      1320      1320
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8
comps
## X       26.01    44.96    62.49    65.22    68.52    72.89    77.13
80.46
## Apps    75.74    82.40    86.74    90.58    92.34    92.79    92.88
92.93
##       9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X       82.45     84.76     88.08     90.76     92.80     94.45     97.02
## Apps    92.98     93.00     93.01     93.01     93.02     93.02     93.02
##       16 comps  17 comps
## X        98.03    100.00
## Apps     93.02     93.02

pls.pred = predict(pls.fit, College.test, ncomp = 9)
mean((pls.pred - College.test$Apps)^2)

## [1] 1007163
```

-> Lowest MSEP with PCR dimension reduction around M = 8 -> M = 8 best performing PLS model with test error = 978534.3 -> Best performing model second to ridge regression

Part (g)

```
TOTALSUMOFSQUARES = sum((mean(College.test$Apps) - College.test$Apps)^2)
TOTALSUMOFRESIDUALS = sum((ridge.pred - College.test$Apps)^2)
1 - (TOTALSUMOFRESIDUALS) / (TOTALSUMOFSQUARES)

## [1] 0.9240954
```

-> Best to worst performing models based upon test-error: (1) Ridge Regression = 940970.9 | (2) PLS = 978534.3 | (3) Linear Model = 984743.1 | (4) Lasso Model = 993741.7 | (5) PCR = 1682909 -> Best model R^2 from ridge regression means 92.4% of variance in Apps using the model