STA 545 Statistical Data Mining I, Fall 2020

Homework 4

Stella Liao

September 30, 2020

1. In this exercise, we will predict the number of applications received using the other variables in the College data set. Please install the ISLR R package to download this data set.

(a) (5 points) This data set has 777 observations. Please randomly split the data set into a training set (500 observations) and a test set (277 observations). Please use the set.seed() function in this step so that you can reproduce your following analysis results.

```r
set.seed(123)
train.num <- sample(777, size = 500, replace = FALSE)
train.College <- College[train.num,]
test.College <- College[-train.num,]
```

(b) (15 points) Fit a linear model using least squares on the training set. Report the estimated regression coefficients and the test error obtained.

The estimated regression coefficients and the test error are shown in the output of the chunk below.

```r
lm.OLS = lm(Apps ~ ., train.College)
preds.OLS <- predict(lm.OLS, test.College)

#the estimated regression coeficients
coef(lm.OLS)
```

```
##   (Intercept)    PrivateYes        Accept        Enroll     Top10perc
## -2.226785e+02 -6.415244e+02  1.281979e+00  1.201598e-01  4.524285e+01
##     Top25perc   F.Undergrad   P.Undergrad      Outstate    Room.Board
## -1.314847e+01  7.080251e-03  3.458198e-02 -5.634890e-02  1.909362e-01
##         Books      Personal           PhD      Terminal     S.F.Ratio
##  1.373598e-01 -2.548125e-02 -6.015355e+00 -7.648977e+00 -1.416427e+00
##   perc.alumni        Expend     Grad.Rate
## -5.730415e+00  7.647485e-02  9.616039e+00
```

```r
#the test error
MSE <- mean((preds.OLS - test.College$Apps)^2)
MSE
```

```
## [1] 1566875
```

(c) (15 points) If we fit the ridge regression model on the training set considering all possible values of the tuning parameter, which ridge regression model has the lowest training error? If we fit the PCR model on the training set considering all possible values of the tuning parameter M, which PCR model has the lowest training error? Are these two models always the same as the linear model in part (b)? Why?

1) ridge regression model

In order to consider the full range of λ in a ridge regression, we create a grid to contain the values ranging from 10^{-2} to 10^{10}; [1]

```
X.College <- model.matrix(Apps ~ .,train.College)
grid <- 10 ^ seq(10, -2, length = 100)

ridge.mod <- glmnet(X.College,
                    train.College$Apps,
                    alpha = 0,
                    lambda = grid,
                    thresh = 1e-12)

training.errors.ridge <- c()
for (i in 1:length(grid)){
  preds.ridge <- predict(ridge.mod,
                         s = grid[i],
                         newx = X.College)
  training.errors.ridge[i] <- mean((preds.ridge - train.College$Apps)^2)
}

lowest.training.error.ridge <- min(training.errors.ridge)
lowest.training.error.ridge
```

```
## [1] 915123.1
```

```
lowest.lambda <- grid[which.min(training.errors.ridge)]
lowest.lambda
```

```
## [1] 0.01
```

Therefore, in the ridge regression model, when λ is 0.01, we have the lowest training error which is 915123.1.

2) PCR model

```
M <- c()
training.errors.PCR <- c()
for (i in 1:17){
  M[i] <- i
  fit.pcr <- pcr(Apps ~ .,
               data = train.College,
               scale = TRUE,
               ncomp = i)
  pred.pcr <- predict(fit.pcr,
                      train.College,
                      ncomp = i)

  training.errors.PCR[i] <- mean((pred.pcr - train.College$Apps)^2)
}

lowest.training.error.PCR <- min(training.errors.PCR)
```

```r
lowest.M <- which.min(training.errors.PCR)

lowest.M
```

```
## [1] 17
```

```r
lowest.training.error.PCR
```

```
## [1] 915123.1
```

Therefore, in the PCR model, when M is 17, we have the lowest training error which is 915123.1.

3) the coeefficients from different models

```r
coefficient.OLS <- as.data.frame(coef(lm.OLS))
coefficient.ridge <- predict(ridge.mod, s = lowest.lambda, type = "coefficient")
coefficient.OLS
```

```
##              coef(lm.OLS)
## (Intercept) -2.226785e+02
## PrivateYes  -6.415244e+02
## Accept       1.281979e+00
## Enroll       1.201598e-01
## Top10perc    4.524285e+01
## Top25perc   -1.314847e+01
## F.Undergrad  7.080251e-03
## P.Undergrad  3.458198e-02
## Outstate    -5.634890e-02
## Room.Board   1.909362e-01
## Books        1.373598e-01
## Personal    -2.548125e-02
## PhD         -6.015355e+00
## Terminal    -7.648977e+00
## S.F.Ratio   -1.416427e+00
## perc.alumni -5.730415e+00
## Expend       7.647485e-02
## Grad.Rate    9.616039e+00
```

```r
coefficient.ridge
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept) -2.227466e+02
## (Intercept)  .
## PrivateYes  -6.415233e+02
## Accept       1.281938e+00
## Enroll       1.202214e-01
## Top10perc    4.524106e+01
## Top25perc   -1.314740e+01
## F.Undergrad  7.087521e-03
## P.Undergrad  3.457781e-02
## Outstate    -5.634512e-02
## Room.Board   1.909404e-01
## Books        1.373671e-01
## Personal    -2.548455e-02
## PhD         -6.015156e+00
## Terminal    -7.649065e+00
```

```
## S.F.Ratio   -1.416467e+00
## perc.alumni -5.730985e+00
## Expend       7.647562e-02
## Grad.Rate    9.616172e+00
```

In summary, we could find that the lowest training errors in the ridge regression model and the PCR model are same but are different from the test error in part (b). I think the main reason of the difference is because in part(b), we used test data to calculate the test error; while in part(c), we used train data to calculate the training error. And actually the coefficients from the OLS and ridge regression are similar. And in the PCR model, because we finally chose M = 17, meaning we use all predictors to predict the response and in the OLS, we also consider the all predictors.

(d) (5 points) Further split the training set into two parts randomly: set A (250 observations) and set B (250 observations). Please use the set.seed() function in this step so that you can reproduce your following analysis results.

```
set.seed(123)
set.num <- sample(500, size = 250, replace = FALSE)
set.A <- College[set.num,]
set.B <- College[-set.num,]
```

(e) (15 points) Fit a ridge regression model on the set A, with the tuning parameter λ chosen by the set B. Report the estimated regression coefficients and the test error obtained.

In order to find the ridge regression model with the lowest training errors, we could use cross-validation to find the best λ by applying the function `cv.glmnet()`.[1]

```
X.set.A <- model.matrix(Apps ~ .,set.A)
X.set.B <- model.matrix(Apps ~ .,set.B)
test.matrix.College <- model.matrix(Apps ~ .,test.College)
grid <- 10 ^ seq(10, -2, length = 100)

cv.ridge <- cv.glmnet(X.set.B,
                      set.B$Apps,
                      alpha = 0,
                      lambda = grid,
                      thresh = 1e-12)
set.seed(1)
bestlambda.B <- cv.ridge$lambda.min
bestlambda.B
```

```
## [1] 0.01
```

```
ridge.mod2 <- glmnet(X.set.A,
                     set.A$Apps,
                     alpha = 0,
                     lambda = bestlambda.B,
                     thresh = 1e-12)

preds.ridge2 <- predict(ridge.mod2,
                        s = bestlambda.B,
                        newx = test.matrix.College)
#test error in ridge model
test.error.ridge <- mean((preds.ridge2 - test.College$Apps)^2)
test.error.ridge
```

```
## [1] 1652713
```

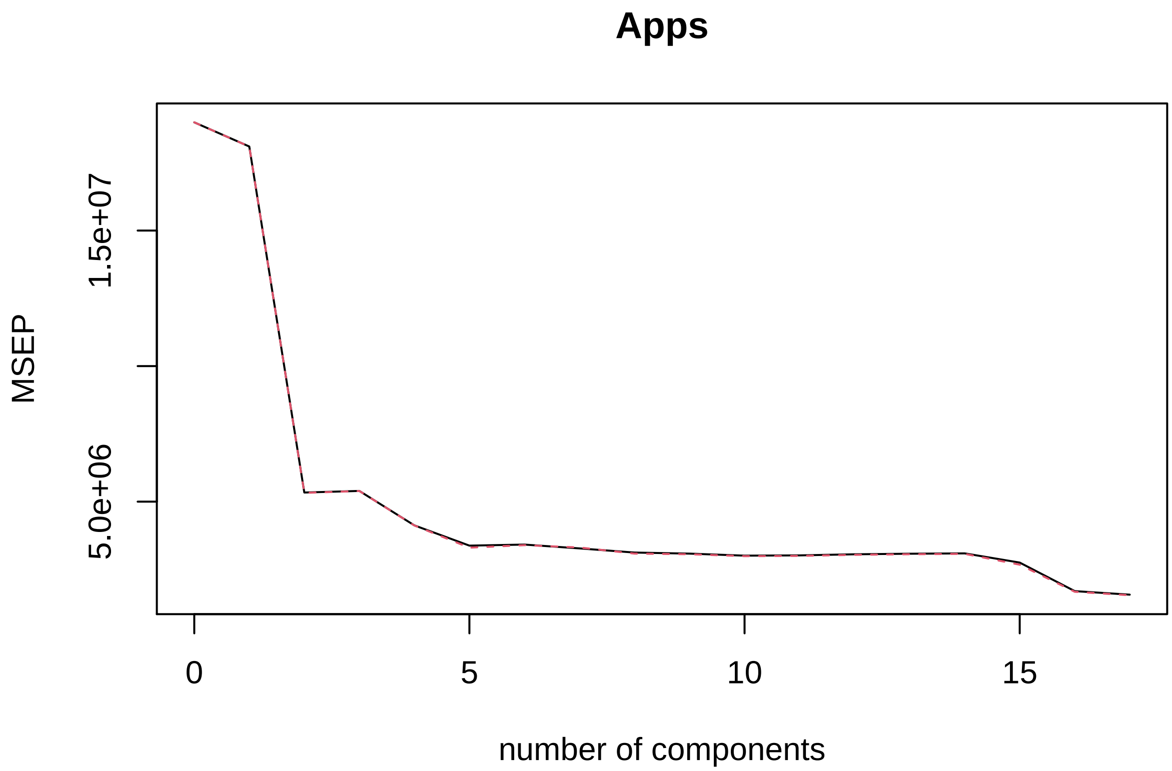
```r
#the coefficients of ridge model
coef(ridge.mod2)
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)  138.63632955
## (Intercept)    .
## PrivateYes  -583.38513264
## Accept         1.30454060
## Enroll         0.34999894
## Top10perc     18.74028819
## Top25perc     -1.84402160
## F.Undergrad   -0.05699445
## P.Undergrad    0.08073815
## Outstate      -0.03430440
## Room.Board     0.10191916
## Books         -0.32159877
## Personal       0.05701666
## PhD            1.44507833
## Terminal     -10.76851858
## S.F.Ratio    -24.20451256
## perc.alumni  -12.35188074
## Expend         0.08161151
## Grad.Rate     10.27349873
```

(f) (15 points) Fit a PCR model on the set A, with the parameter M chosen by the set B. Report the value of M selected by the set B, the estimated regression coefficients of the original input variables, and the test error obtained.

In PCR model, we still use cross validation to choose M by setting the argument `validation` equal to `"CV"`. And we choose the M which could make the cross validation error lowest, which will be shown in the output of `summary(fit.pcr2)`.[2]

```r
fit.pcr2 <- pcr(Apps ~ .,
                data = set.B,
                scale = TRUE,
                validation = "CV")
validationplot(fit.pcr2,val.type = "MSEP")
```




```r
summary(fit.pcr2)
```

```
## Data:    X dimension: 527 17
##  Y dimension: 527 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            4358     4255     2310     2323     2031     1838     1848
## adjCV         4358     4254     2307     2322     2030     1817     1842
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1810     1767     1756      1734      1737      1749      1755
## adjCV     1819     1756     1749      1729      1731      1743      1749
##        14 comps  15 comps  16 comps  17 comps
## CV         1758      1659      1304      1252
## adjCV      1753      1635      1294      1244
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X      31.232    57.24    64.56    70.22    75.51    80.47    84.02    87.53
## Apps    5.098    72.58    72.70    80.00    83.59    83.63    84.00    85.16
##       9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X       90.54     93.02     95.01     96.83     97.84     98.72     99.36
## Apps    85.36     85.73     85.77     85.79     85.80     85.82     90.36
##       16 comps  17 comps
## X        99.83    100.00
## Apps     93.09     93.52
```

According to the output of `summary(fit.pcr2)`, we find that when `ncomp` = 17, the cross validation error is lowest.

```
fit.pcr3 <- pcr(Apps ~ .,
                data = set.A,
                ncomp = 17)

pred.pcr2 <- predict(fit.pcr3,
                     test.College,
                     ncomp = 17)

# test error in PCR model
test.error.PCR <- mean((pred.pcr2 - test.College$Apps)^2)
test.error.PCR
```

```
## [1] 1652639
```

```
#the coefficients of the original outputs
as.data.frame(fit.pcr3$coefficients[,,17])
```

```
##             fit.pcr3$coefficients[, , 17]
## PrivateYes                  -583.39838762
## Accept                         1.30460049
## Enroll                         0.34988992
## Top10perc                     18.74143409
## Top25perc                     -1.84446757
## F.Undergrad                   -0.05700282
## P.Undergrad                    0.08074157
## Outstate                      -0.03430913
## Room.Board                     0.10191458
## Books                         -0.32161239
## Personal                       0.05701892
## PhD                            1.44504705
## Terminal                     -10.76860455
## S.F.Ratio                    -24.20528910
## perc.alumni                  -12.35171467
## Expend                         0.08161147
## Grad.Rate                     10.27343395
```

Problem 2

Because $y_1+y_2=0$, $x_{11}+x_{21}=0$, $x_{12}+x_{22}=0$, we could know that the estimate for the intercept in a ridge regression should be zero. At this point, $\hat{\beta}_0=0$.

In ridge regression problem, we need to find the coefficients to minimize

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2 \quad ①.$$

And according to the settings that $n=2$, $p=2$, $x_{11}=x_{12}=x_1$, $x_{21}=x_{22}=x_2$, the expression ① could be represented by that

$$(y_1-\hat{\beta}_1x_1-\hat{\beta}_2x_1)^2+(y_2-\hat{\beta}_1x_2-\hat{\beta}_2x_2)^2+\lambda(\hat{\beta}_1^2+\hat{\beta}_2^2). \quad ②$$

Therefore, we need to make the derivatives of the expression ② with respects to $\hat{\beta}_1$ and $\hat{\beta}_2$ seperately and make them equal to zero, so we will have that

$$\hat{\beta}_1(x_1^2+x_2^2+\lambda)+\hat{\beta}_2(x_1^2+x_2^2)=y_1x_1+y_2x_2 \quad ①$$

$$\hat{\beta}_1(x_1^2+x_2^2)+\hat{\beta}_2(x_1^2+x_2^2+\lambda)=y_1x_1+y_2x_2 \quad ②.$$

It's easy to find that the left sides of the equation ① and ② are equal because they are all equal to $y_1x_1+y_2x_2$.

$$\hat{\beta}_1(x_1^2+x_2^2+\lambda)+\hat{\beta}_2(x_1^2+x_2^2)=\hat{\beta}_1(x_1^2+x_2^2)+\hat{\beta}_2(x_1^2+x_2^2+\lambda)$$

$$\hat{\beta}_1[x_1^2+x_2^2+\lambda-(x_1^2+x_2^2)]=\hat{\beta}_2[(x_1^2+x_2^2+\lambda)-(x_1^2+x_2^2)]$$

$$\hat{\beta}_1\cdot\lambda=\hat{\beta}_2\cdot\lambda$$

$$\hat{\beta}_1=\hat{\beta}_2$$

Problem 3

1. $X = UDV^T$

U: a $n \times p$ orthogonal matrix, $U^TU = I$

D: $p \times p$ matrix, and $D_{ii} \geq 0$; when $i \neq j$, $D_{ij} = 0$

V: a $p \times p$ orthogonal matrix, $V^TV = VV^T = I$

2. $\hat{\beta}_{ridge} = \text{argmin}_{w'} = \|y - \hat{y}\|_2^2 + \lambda \|w'\|_2^2 = (X^TX + \lambda I)^{-1} X^T y$

3.
$$
\begin{aligned}
\hat{y}_{ridge} = X\hat{\beta}_{ridge} &= X(X^TX + \lambda I)^{-1} X^T y \\
&= UDV^T(VD^2V^T + \lambda I)^{-1} VDU^T y \\
&= UDV^T(VD^2V^T + \lambda VV^T)^{-1} VDU^T y \\
&= UDV^T(V(D^2 + \lambda I)V^T)^{-1} VDU^T y \\
&= UDV^TV(D^2 + \lambda I)^{-1} V^TVDU^T y \\
&= UD(D^2 + \lambda I)^{-1} DU^T y \quad ①
\end{aligned}
$$

4. we denote that $\tilde{D} = D(D^2 + \lambda I)^{-1} D$, then we have

$$\tilde{D}_{jj} = \frac{D_{jj}^2}{D_{jj}^2 + \lambda} = \frac{d_j^2}{d_j^2 + \lambda} \quad ②$$

5. we put the equation ② into the equation ①, and finally we will get that

$$\hat{y}_{ridge} = X\hat{\beta}_{ridge} = \sum_{j=1}^{p} u_j \frac{d_j^2}{d_j^2 + \lambda} u_j^T y.$$

References

[1] 6.5 Lab 2: Ridge Regression and the Lasso, Chapter 6 Linear Model Selection and regularization
[2] 6.6 Lab 3: PCR and PLS Regression, Chapter 6 Linear Model Selection and regularization