STAT432_HW5

Taiga Hasegawa(taigah2)

2019/2/25

Question1

(a)$E[(Y - X\hat{\beta})^2] = E[(X\beta + e - X\hat{\beta})^2] = E[(X\beta + e - X\beta + X\beta - E[X\hat{\beta}] + E[X\hat{\beta}] - X\hat{\beta})^2] = E[(X\beta + e - X\beta)^2] + (X\beta - XE[\hat{\beta}])^2 + E[(X\hat{\beta} - E(X\hat{\beta}))^2] = e^2 + 0 + \sigma^2$

irreducible error: square error term(e^2). This cannot be reduced becuase we can't predict the e. bias: 0 because OLS estimator is unbiased estimator. This evaluates how the average of our estimator deviates from the truth. variance: σ^2 . This reflects the sensitivity of the function estimate f(x) to the training sample

(b) In ridge regression, the element of β is shrinked by tuning parameter. $\beta^{ridge} = \alpha\beta^{OLS}$ where α is no more than 0. If we calculate the variance, it(tuning parameter>0) must be smaller than the variance of OLS estimator (tuning parameter=0) by α^2, whereas it(tuning parametr>0) is biased because the OLS estimator is biased (tuning parameter=0). That's why as the tuning parameter increase, the vairance decreases and bias increases.

Question2

```r
data(Boston, package="MASS")
head(Boston)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12
##   lstat medv
## 1  4.98 24.0
## 2  9.14 21.6
## 3  4.03 34.7
## 4  2.94 33.4
## 5  5.33 36.2
## 6  5.21 28.7
```

```r
useLog = c(1,3,5,6,8,9,10,14)
Boston[,useLog] = log(Boston[,useLog])
Boston[,2] = Boston[,2] / 10
Boston[,7] = Boston[,7]^2.5 / 10^4
Boston[,11] = exp(0.4 * Boston[,11])/1000
Boston[,12] = Boston[,12] / 100
Boston[,13] = sqrt(Boston[,13])
```

a)

```r
#fit a linear regression
fit=lm(medv~., data=Boston)
summary(fit)
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9918 -0.1002 -0.0034  0.1117  0.7640
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.176874   0.379017  11.020  < 2e-16 ***
## crim        -0.014606   0.011650  -1.254 0.210527
## zn           0.001392   0.005639   0.247 0.805121
## indus       -0.012709   0.022312  -0.570 0.569195
## chas         0.109980   0.036634   3.002 0.002817 **
## nox         -0.283112   0.105340  -2.688 0.007441 **
## rm           0.421108   0.110175   3.822 0.000149 ***
## age          0.006403   0.004863   1.317 0.188536
## dis         -0.183154   0.036804  -4.977 8.97e-07 ***
## rad          0.068362   0.022473   3.042 0.002476 **
## tax         -0.201832   0.048432  -4.167 3.64e-05 ***
## ptratio     -0.040017   0.008091  -4.946 1.04e-06 ***
## black        0.044472   0.011456   3.882 0.000118 ***
## lstat       -0.262615   0.016091 -16.320  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2008 on 492 degrees of freedom
## Multiple R-squared:  0.765,  Adjusted R-squared:  0.7588
## F-statistic: 123.2 on 13 and 492 DF,  p-value: < 2.2e-16
```

(b)

```r
#Calculate the Mallow's Cp statistic
n=nrow(Boston)
RSS=sum(fit$residuals^2)
p=ncol(Boston)
RSS+2*p*summary(fit)$sigma^2
```

```
## [1] 20.95689
```

```r
#Calculate the -2*loglikelihood
sigma2=RSS/n
loglikelihood= n*log(sigma2) + n + n*log(2*pi)+2
loglikelihood
```

```
## [1] -201.1805
```

```r
#Calculate the AIC
loglikelihood+2*p
```

```
## [1] -173.1805
```

```r
#Calculate the BIC
loglikelihood+log(n)*p
```

```
## [1] -114.0089
```

```r
#Best subset model selection
library(leaps)
#maximum model size is 13
RSSleaps=regsubsets(as.matrix(Boston[,-14]),Boston[,14],nvmax=13)
sumleaps=summary(RSSleaps,matrix=T)
#calculate the AIC
msize=apply(sumleaps$which,1,sum)
AIC = n*log(sumleaps$rss/n) + 2*msize
#13 possible models
sumleaps$which
```

```
##    (Intercept)  crim    zn indus  chas   nox    rm   age   dis   rad   tax
## 1         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 4         TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
## 5         TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE
## 6         TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE
## 7         TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE
## 8         TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE
## 9         TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 10        TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 11        TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 12        TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 13        TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##    ptratio black lstat
## 1    FALSE FALSE  TRUE
## 2     TRUE FALSE  TRUE
## 3     TRUE  TRUE  TRUE
## 4     TRUE FALSE  TRUE
## 5     TRUE FALSE  TRUE
## 6     TRUE  TRUE  TRUE
## 7     TRUE  TRUE  TRUE
## 8     TRUE  TRUE  TRUE
## 9     TRUE  TRUE  TRUE
## 10    TRUE  TRUE  TRUE
## 11    TRUE  TRUE  TRUE
## 12    TRUE  TRUE  TRUE
## 13    TRUE  TRUE  TRUE
```

```r
#show which is the best model based on Cp
which.min(sumleaps$cp)
```

```
## [1] 9
```

```r
#show which is the best model based on AIC
which.min(AIC)
```

```
## 9
## 9
```

```r
#show which is the best model based on BIC
which.min(sumleaps$bic)
```

```
## [1] 9
```

As a result, they all select the 9th model as the best one.

This model has following variables: Intercept, chas, nox, rm, dis, rad, tax, ptratio, black and lstat.

```
sumleaps$which[9,]
```

```
## (Intercept)        crim          zn       indus        chas         nox 
##        TRUE       FALSE       FALSE       FALSE        TRUE        TRUE 
##          rm         age         dis         rad         tax     ptratio 
##        TRUE       FALSE        TRUE        TRUE        TRUE        TRUE 
##       black       lstat 
##        TRUE        TRUE
```

Question3

```
library(MASS)
```

```
## 
## Attaching package: 'MASS'

## The following object is masked _by_ '.GlobalEnv':
## 
##     Boston
```

```
fit = lm.ridge(medv~., Boston, lambda=seq(0,100,by=0.1))
penaltylevel=which.min(fit$GCV)
penaltylevel
```

```
##   6.2 
##    63
```

The best penelty level is 63 based on generalized cross-variation.

The parameter estimates is like below.

```
round(coef(fit)[which.min(fit$GCV), ], 4)
```

```
##            crim      zn   indus    chas     nox      rm     age     dis 
##  4.0428 -0.0134  0.0009 -0.0147  0.1118 -0.2594  0.4538  0.0053 -0.1719 
##     rad     tax ptratio   black   lstat 
##  0.0610 -0.1902 -0.0401  0.0454 -0.2544
```