Problem Set 3 - POLS6394

Tom Hanna

10/29/2020

Problem Set 3

#Chapter 6

4. a. iii - In order to minimize the equation with increasing lambda, the B coefficients must converge to 0.

b. ii - Decrease then increase. With lamba = 0, test RSS is high because of overfitting. As lambda increases overfitting is reduce to a point until the increased shrinkage becomes inefficient.

c. iv - As lambda increases, the B coefficients converge on zero. All betas = 0 is essentially a horizontal line with no variance.

d. iii - Increasing lambda decreases flexibility, which increases bias.

e. v - irreducible error is a horizontal line, a constant, that can not be improved regardless of model fit.

```
rm(list = ls())
setwd("C:/R Studio Files/POLS6394-Machine-Learning/Problem Set 3")

#ISLR, Chapter 6, #4, #9, #11
#ISLR, Chapter 7, #3, #4, #9, #10

College <- read.csv("C:/R Studio Files/POLS6394-Machine-Learning/Problem Set 3/College.csv")
names(College)
```

```
##  [1] "X"           "Private"     "Apps"        "Accept"      "Enroll"
##  [6] "Top10perc"   "Top25perc"   "F.Undergrad" "P.Undergrad" "Outstate"
## [11] "Room.Board"  "Books"       "Personal"    "PhD"         "Terminal"
## [16] "S.F.Ratio"   "perc.alumni" "Expend"      "Grad.Rate"
```

```
set.seed(75)

#9a

train.size = dim(College)[1] / 2
train = sample(1:dim(College)[1], train.size)
test = -train
Train.college = College[train, ]
Test.college = College[test, ]

model1 <- lm(Apps ~ Private + Accept + Enroll + Top10perc + Top25perc + F.Undergrad + P.Undergrad + Outs
                 Room.Board + Books + Personal + PhD + Terminal + S.F.Ratio + perc.alumni + Expend + Gra
             data = Train.college)
model1.pred <- predict(model1, Test.college)
mean((Test.college[, "Apps"] - model1.pred)^2)
```

```
## [1] 1672201
```

```r
#b - Test RSS is 1672201

library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.0.3
## Loading required package: Matrix
## Loaded glmnet 4.0-2
```

```r
mat.train <- model.matrix(Apps ~ Private + Accept + Enroll + Top10perc + Top25perc + F.Undergrad + P.Und
                               Room.Board + Books + Personal + PhD + Terminal + S.F.Ratio + perc.alumni +
                           data = Train.college)
mat.test <- model.matrix(Apps ~ Private + Accept + Enroll + Top10perc + Top25perc + F.Undergrad + P.Unde
                              Room.Board + Books + Personal + PhD + Terminal + S.F.Ratio + perc.alumni +
                          data = Test.college)
grid <- 10 ^ seq(4, -2, length=100)
model.ridge <- cv.glmnet(mat.train, Train.college[, "Apps"], alpha=0, lambda=grid, thresh=1e-12)
lambda1 <- model.ridge$lambda.min
lambda1
```

```
## [1] 14.17474
```

```r
model.ridge.pred <- predict(model.ridge, newx=mat.test, s=lambda1)
mean((Test.college[, "Apps"] - model.ridge.pred)^2)
```

```
## [1] 1732682
```

```r
#c - Test error is 1732682

model.lasso <- cv.glmnet(mat.train, Train.college[, "Apps"], alpha=1, lambda=grid, thresh=1e-12)
lambda2 <- model.lasso$lambda.min
lambda2
```

```
## [1] 9.326033
```

```r
model.lasso.pred <- predict(model.lasso, newx=mat.test, s=lambda2)
mean((Test.college[, "Apps"] - model.lasso.pred)^2)
```

```
## [1] 1731804
```

```r
#Test error is 1731804
#Coefficients:

model.lasso <- glmnet(model.matrix(Apps ~ Private + Accept + Enroll + Top10perc + Top25perc + F.Undergra
                                    Room.Board + Books + Personal + PhD + Terminal + S.F.Ratio + perc.al
                                data = College), College[, "Apps"], alpha=1)
predict(model.lasso, s=lambda2, type="coefficients")
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept) -5.284919e+02
## (Intercept)  .
## PrivateYes  -4.813682e+02
## Accept       1.528758e+00
## Enroll      -4.372593e-01
## Top10perc    4.269076e+01
## Top25perc   -8.657751e+00
```

```
## F.Undergrad  2.751930e-05
## P.Undergrad  4.227091e-02
## Outstate    -7.488284e-02
## Room.Board   1.416735e-01
## Books        .
## Personal     2.147966e-02
## PhD         -7.456087e+00
## Terminal    -3.056619e+00
## S.F.Ratio    1.136791e+01
## perc.alumni -5.334028e-01
## Expend       7.432130e-02
## Grad.Rate    7.032033e+00
```

```r
library(pls)
```

```
## Warning: package 'pls' was built under R version 4.0.3

##
## Attaching package: 'pls'

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

```r
model.pcr <- pcr(Apps ~ Private + Accept + Enroll + Top10perc + Top25perc + F.Undergrad + P.Undergrad +
                     Room.Board + Books + Personal + PhD + Terminal + S.F.Ratio + perc.alumni + Expend +
                 data = Train.college, scale=T, validation="CV")
validationplot(model.pcr, val.type="MSEP")
```

problem-set-3--Tom-Hanna_files/figure-latex/unnamed-chunk-1-1.pdf

```r
model.pcr.pred <- predict(model.pcr, Test.college, ncomp = 10)
mean((Test.college[, "Apps"] - model.pcr.pred)^2)
```

```
## [1] 3360184
```

```r
#Test error is 3360184

model.pls <- plsr(Apps ~ Private + Accept + Enroll + Top10perc + Top25perc + F.Undergrad + P.Undergrad +
                      Room.Board + Books + Personal + PhD + Terminal + S.F.Ratio + perc.alumni + Expend
                  data = Train.college, scale=T, validation="CV")
validationplot(model.pls, val.type="MSEP")
```

problem-set-3--Tom-Hanna_files/figure-latex/unnamed-chunk-1-2.pdf

```r
model.pls.pred <- predict(model.pls, Test.college, ncomp=10)
mean((Test.college[, "Apps"] - model.pls.pred)^2)
```

```
## [1] 1670383
```

```r
#The test error is 1670383

#g - Ridge, Lasso, and PLS provide accurate predictions. PCR has a higher error rate.

rm(list = ls())
setwd("C:/R Studio Files/POLS6394-Machine-Learning/Problem Set 3")

library(MASS)
Boston <- Boston

set.seed(25)

train.size = dim(Boston)[1] / 2
train = sample(1:dim(Boston)[1], train.size)
test = -train
Train.Boston = Boston[train, ]
Test.Boston = Boston[test, ]

model1 <- lm(crim ~ zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black + lstat + med
             data = Train.Boston)


model1.pred <- predict(model1, Test.Boston)
mean((Test.Boston[, "crim"] - model1.pred)^2)
```

```
## [1] 41.32216
```

```r
#Test RSS is 41.32216

library(glmnet)

mat.train <- model.matrix(crim ~ zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black
mat.test <- model.matrix(crim ~ zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black +
grid <- 10 ^ seq(4, -2, length=100)
model.ridge <- cv.glmnet(mat.train, Train.Boston[, "crim"], alpha=0, lambda=grid, thresh=1e-12)
lambda1 <- model.ridge$lambda.min
lambda1
```

```
## [1] 0.1873817
```

```r
model.ridge.pred <- predict(model.ridge, newx=mat.test, s=lambda1)
mean((Test.Boston[, "crim"] - model.ridge.pred)^2)
```

```
## [1] 41.52214
```

```r
#Test error is 41.52214

model.lasso <- cv.glmnet(mat.train, Train.Boston[, "crim"], alpha=1, lambda=grid, thresh=1e-12)
lambda2 <- model.lasso$lambda.min
lambda2
```

```
## [1] 0.06135907
```

```r
model.lasso.pred <- predict(model.lasso, newx=mat.test, s=lambda2)
mean((Test.Boston[, "crim"] - model.lasso.pred)^2)
```

```
## [1] 41.26353
```

```r
#Test error is 41.26353
#Coefficients:

model.lasso <- glmnet(model.matrix(crim ~ zn + indus + chas + nox + rm + age + dis + rad + tax + ptrati
predict(model.lasso, s=lambda2, type="coefficients")
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) 11.968350618
## (Intercept)  .
## zn           0.035212862
## indus       -0.066930199
## chas        -0.570351234
## nox         -6.292517825
## rm           0.187608148
## age          .
## dis         -0.748763104
## rad          0.510293154
## tax          .
## ptratio     -0.171162416
## black       -0.007553759
## lstat        0.123966707
## medv        -0.151139614
```

```r
library(pls)

model.pcr <- pcr(crim ~ zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black + lstat +
validationplot(model.pcr, val.type="MSEP")
```

problem-set-3--Tom-Hanna_files/figure-latex/unnamed-chunk-1-3.pdf

```r
model.pcr.pred <- predict(model.pcr, Test.Boston, ncomp = 10)
mean((Test.Boston[, "crim"] - model.pcr.pred)^2)
```

```
## [1] 43.96352
```

```r
#Test error is 43.96352

model.pls <- plsr(crim ~ zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black + lstat
validationplot(model.pls, val.type="MSEP")
```

problem-set-3--Tom-Hanna_files/figure-latex/unnamed-chunk-1-4.pdf

```r
model.pls.pred <- predict(model.pls, Test.Boston, ncomp=10)
mean((Test.Boston[, "crim"] - model.pls.pred)^2)
```

```
## [1] 41.34152
```

```r
#11 -b

library (leaps)
```

```
## Warning: package 'leaps' was built under R version 4.0.3
```

```r
regfit.full=regsubsets (crim~. ,data = Train.Boston)

reg.summary =summary(regfit.full)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(crim ~ ., data = Train.Boston)
## 13 Variables  (and intercept)
##         Forced in Forced out
## zn          FALSE      FALSE
## indus       FALSE      FALSE
## chas        FALSE      FALSE
## nox         FALSE      FALSE
## rm          FALSE      FALSE
## age         FALSE      FALSE
## dis         FALSE      FALSE
## rad         FALSE      FALSE
## tax         FALSE      FALSE
## ptratio     FALSE      FALSE
## black       FALSE      FALSE
## lstat       FALSE      FALSE
## medv        FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          zn  indus chas nox rm  age dis rad tax ptratio black lstat medv
## 1  ( 1 ) " " " "   " "  " " " " " " " " "*" " " " "     " "   " "   " "
## 2  ( 1 ) " " " "   " "  " " " " " " " " "*" " " " "     " "   "*"   " "
## 3  ( 1 ) " " " "   " "  " " " " " " " " "*" " " " "     "*"   "*"   " "
## 4  ( 1 ) "*" " "   " "  " " " " " " " " "*" " " " "     "*"   "*"   " "
## 5  ( 1 ) "*" " "   " "  " " " " " " "*" "*" " " " "     "*"   "*"   " "
## 6  ( 1 ) "*" " "   " "  "*" " " " " "*" "*" " " " "     "*"   "*"   " "
## 7  ( 1 ) "*" " "   " "  "*" " " " " "*" "*" " " " "     "*"   "*"   "*"
## 8  ( 1 ) "*" " "   " "  "*" " " " " "*" "*" " " "*"     "*"   "*"   "*"
```

```r
reg.summary$rsq
```

```
## [1] 0.4224482 0.4560671 0.4648444 0.4682357 0.4704521 0.4757076 0.4811685
## [8] 0.4840637
```

```r
par(mfrow =c(2,2))
plot(reg.summary$rss ,xlab=" Number of Variables ",ylab=" RSS",
       type="l")
plot(reg.summary$adjr2 ,xlab =" Number of Variables ",
       ylab=" Adjusted RSq",type="l")
plot(reg.summary$bic ,xlab =" Number of Variables ",
     ylab="BIC",type="l")

##BIC is minimized with 2 variables. With 2 features, and "rad" and "lstat" produce the best model.
```

```
#C - No. BIC was minimized with 2 features.

##Chapter 7
#3

x = -2:2
y = 1 + 1*x + -2 * (x-1)^2 * I(x>=1)
plot(x, y)
```

problem-set-3--Tom-Hanna_files/figure-latex/unnamed-chunk-1-5.pdf

```
#4

x=seq(-2,2,by=0.01)
y2 = function(x){
    # b1(X) = I(0  X  2)  (X 1)I(1  X  2)
    # b2(X) = (X 3)I(3  X  4)+I(4 < X  5)
    1 + (I(x>=0 && x<=2) - ((x-1)*I(x>=1 && x<=2))) + 3*((x-3)*I(x>=3 && x<=4) + I(x>4 && x<=5))
}
yp = sapply(x,FUN=y2)
plot(x,yp,xlim=c(-2,2),ylim=c(0,2.5))
#points(x=x,y=y(x),pch=19,col="red")

#9
#A -
model1 <- lm(nox ~ poly(dis, 3), data = Boston)
model1
```

```
##
## Call:
## lm(formula = nox ~ poly(dis, 3), data = Boston)
##
## Coefficients:
##   (Intercept)  poly(dis, 3)1  poly(dis, 3)2  poly(dis, 3)3
##        0.5547        -2.0031         0.8563        -0.3180
```

```
#B -

model2 <- lm(nox ~ poly(dis,2), data = Boston)
model3 <- lm(nox ~ poly(dis,3), data = Boston)
model4 <- lm(nox ~ poly(dis,4), data = Boston)
model5 <- lm(nox ~ poly(dis,5), data = Boston)
model6 <- lm(nox ~ poly(dis,6), data = Boston)
model7 <- lm(nox ~ poly(dis,7), data = Boston)
model8 <- lm(nox ~ poly(dis,8), data = Boston)
model9 <- lm(nox ~ poly(dis,9), data = Boston)
model10 <- lm(nox ~ poly(dis,10), data = Boston)

anova(model1)["Residuals", "Sum Sq"]
```

```
## [1] 1.934107
```

```r
anova(model2)["Residuals", "Sum Sq"]
```

```
## [1] 2.035262
```

```r
anova(model3)["Residuals", "Sum Sq"]
```

```
## [1] 1.934107
```

```r
anova(model4)["Residuals", "Sum Sq"]
```

```
## [1] 1.932981
```

```r
anova(model5)["Residuals", "Sum Sq"]
```

```
## [1] 1.91529
```

```r
anova(model6)["Residuals", "Sum Sq"]
```

```
## [1] 1.878257
```

```r
anova(model7)["Residuals", "Sum Sq"]
```

```
## [1] 1.849484
```

```r
anova(model8)["Residuals", "Sum Sq"]
```

```
## [1] 1.83563
```

```r
anova(model9)["Residuals", "Sum Sq"]
```

```
## [1] 1.833331
```

```r
anova(model10)["Residuals", "Sum Sq"]
```

```
## [1] 1.832171
```

```r
#c -
library(ISLR)
set.seed=(45)
train=sample (405,101)

model1 <- lm(nox ~ poly(dis,1), data = Boston, subset =train)
model2 <- lm(nox ~ poly(dis,2), data = Boston, subset =train)
model3 <- lm(nox ~ poly(dis,3), data = Boston, subset =train)
model4 <- lm(nox ~ poly(dis,4), data = Boston, subset =train)
model5 <- lm(nox ~ poly(dis,5), data = Boston, subset =train)
model6 <- lm(nox ~ poly(dis,6), data = Boston, subset =train)
model7 <- lm(nox ~ poly(dis,7), data = Boston, subset =train)
model8 <- lm(nox ~ poly(dis,8), data = Boston, subset =train)
model9 <- lm(nox ~ poly(dis,9), data = Boston, subset =train)
model10 <- lm(nox ~ poly(dis,10), data = Boston, subset =train)

mean((Boston$nox - predict(model1,Boston))[-train]^2)
```

```
## [1] 0.006026416
```

```r
mean((Boston$nox - predict(model2,Boston))[-train]^2)
```

```
## [1] 0.004310486
```

```r
mean((Boston$nox - predict(model3,Boston))[-train]^2)
```

```
## [1] 0.004150605
```

```r
mean((Boston$nox - predict(model4,Boston))[-train]^2)
```

```
## [1] 0.004337492
```

```r
mean((Boston$nox - predict(model5,Boston))[-train]^2)
```

```
## [1] 0.004282151
```

```r
mean((Boston$nox - predict(model6,Boston))[-train]^2)
```

```
## [1] 0.004308835
```

```r
mean((Boston$nox - predict(model7,Boston))[-train]^2)
```

```
## [1] 0.004376609
```

```r
mean((Boston$nox - predict(model8,Boston))[-train]^2)
```

```
## [1] 0.01291827
```

```r
mean((Boston$nox - predict(model9,Boston))[-train]^2)
```

```
## [1] 0.0579271
```

```r
mean((Boston$nox - predict(model10,Boston))[-train]^2)
```

```
## [1] 0.06482916
```

```r
#The validation error is lowest with the 3rd degree polynomial, model3.

#d -

summary(Train.Boston$dis)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.130   1.993   3.099   3.688   5.212  10.586
```

```r
library (splines )

model.spline <- lm(nox ~ bs(dis,df=4,knots =c(2.197,3.272,5.117)), data = Train.Boston)
pred <- predict(model.spline,newdata=Test.Boston,se=T)
```

```
## Warning in bs(dis, degree = 3L, knots = c(2.197, 3.272, 5.117), Boundary.knots
## = c(1.1296, : some 'x' values beyond boundary knots may cause ill-conditioned
## bases
```

```r
attach(Test.Boston)
par(mfrow =c(2,2))
```

problem-set-3--Tom-Hanna_files/figure-latex/unnamed-chunk-1-6.pdf

```r
plot(dis,nox,col="gray")
lines(Test.Boston$dis,pred$fit,lwd=2)
lines(Test.Boston$dis,pred$fit +2* pred$se.fit,lty="dashed")
lines(Test.Boston$dis,pred$fit -2* pred$se.fit,lty="dashed")

#I placed the knots at the quartiles for dis


model.spline6 <- lm(nox ~ bs(dis,df=6), data = Train.Boston)
pred6 <- predict(model.spline6,newdata=Test.Boston,se=T)
```

```
## Warning in bs(dis, degree = 3L, knots = c(`25%` = 1.9929, `50%` = 3.0992, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases
```

```r
plot(dis,nox,col="gray")
lines(Test.Boston$dis,pred6$fit,lwd=2)
lines(Test.Boston$dis,pred6$fit +2* pred$se.fit,lty="dashed")
lines(Test.Boston$dis,pred6$fit -2* pred$se.fit,lty="dashed")

model.spline8 <- lm(nox ~ bs(dis,df=8), data = Train.Boston)
pred8 <- predict(model.spline8,newdata=Test.Boston,se=T)
```

```
## Warning in bs(dis, degree = 3L, knots = c(`16.66667%` = 1.8629, `33.33333%` =
## 2.2222, : some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```r
plot(dis,nox,col="gray")
lines(Test.Boston$dis,pred8$fit,lwd=2)
lines(Test.Boston$dis,pred8$fit +2* pred$se.fit,lty="dashed")
lines(Test.Boston$dis,pred8$fit -2* pred$se.fit,lty="dashed")


model.spline10 <- lm(nox ~ bs(dis,df=10), data = Train.Boston)
pred10 <- predict(model.spline10,newdata=Test.Boston,se=T)
```

```
## Warning in bs(dis, degree = 3L, knots = c(`12.5%` = 1.74435, `25%` = 1.9929, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```r
plot(dis,nox,col="gray")
lines(Test.Boston$dis,pred10$fit,lwd=2)
lines(Test.Boston$dis,pred10$fit +2* pred$se.fit,lty="dashed")
lines(Test.Boston$dis,pred10$fit -2* pred$se.fit,lty="dashed")
```

problem-set-3--Tom-Hanna_files/figure-latex/unnamed-chunk-1-7.pdf

```r
anova(model.spline)["Residuals", "Sum Sq"]
```

```
## [1] 1.029811
```

```r
anova(model.spline6)["Residuals", "Sum Sq"]
```

```
## [1] 1.02421
```

```r
anova(model.spline8)["Residuals", "Sum Sq"]
```

```
## [1] 0.9942262
```

```r
anova(model.spline10)["Residuals", "Sum Sq"]
```

```
## [1] 0.9792116
```

```r
#The model with 10 degrees of freedom had the lowest RSS.

#10A

rm(list = ls())
setwd("C:/R Studio Files/POLS6394-Machine-Learning/Problem Set 3")
library(leaps)

College <- read.csv("C:/R Studio Files/POLS6394-Machine-Learning/Problem Set 3/College.csv")
names(College)
```

```
##  [1] "X"           "Private"     "Apps"        "Accept"      "Enroll"
##  [6] "Top10perc"   "Top25perc"   "F.Undergrad" "P.Undergrad" "Outstate"
## [11] "Room.Board"  "Books"       "Personal"    "PhD"         "Terminal"
## [16] "S.F.Ratio"   "perc.alumni" "Expend"      "Grad.Rate"
```

```r
attach(College)

set.seed(25)
train <- sample(length(Outstate), length(Outstate)/2)
test <- -train
Train.College <- College[train, ]
Test.College <- College[test, ]
reg.fit <- regsubsets(Outstate ~ ., data = Train.College, nvmax = 17, method = "forward")
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 17 linear dependencies found
```

```r
reg.summary <- summary(reg.fit)

par(mfrow = c(2,1))

plot(reg.summary$bic, xlab = "Variables", ylab = "BIC", type = "l")
min.bic <- min(reg.summary$bic)
std.bic <- sd(reg.summary$bic)
abline(h = min.bic + 0.2 * std.bic, col = "red", lty = 2)
abline(h = min.bic - 0.2 * std.bic, col = "red", lty = 2)

plot(reg.summary$adjr2, xlab = "Variables", ylab = "Adjusted R2",
     type = "l", ylim = c(0.4, 0.84))
max.adjr2 <- max(reg.summary$adjr2)
std.adjr2 <- sd(reg.summary$adjr2)
abline(h = max.adjr2 + 0.2 * std.adjr2, col = "red", lty = 2)
abline(h = max.adjr2 - 0.2 * std.adjr2, col = "red", lty = 2)
```

problem-set-3--Tom-Hanna_files/figure-latex/unnamed-chunk-1-8.pdf