Stat 897 Project 1

Penn State

October 1, 2017

Linear Regression, Variable Selection, Ridge Regression, Lasso

This project is to be completed individually. You may submit pdf only (Rmd is not needed and you can use another word processing tool if you like).

The diabetes data in Efron et al. (2003) will be used: ten baseline variables: age, sex, body mass index, average blood pressure, and six blood serum measurements were obtained for each of n = 442 diabetes patients, as well as the response of interest, a quantitative measure of disease progression one year after baseline. The data is available in R package lars. Load the data:

```
library(lars)
```

```
## Loaded lars 1.2
```

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.2.4

## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 3.2.5

## Loading required package: foreach

## Loaded glmnet 2.0-5
```

```
data(diabetes)
data.all = data.frame(cbind(diabetes$x, y=diabetes$y))
```

Partition the patients into two groups: training (75%) and test (25%). Please use use the random number generator seed specified below.

```
n = dim(data.all)[1] # sample size = 442
set.seed(38723) # set random number generator seed to enable
# repeatability of results
test = sample(n, round(n/4))  # randomly sample 25% test
data.train = data.all[-test,]
data.test = data.all[test,]
x = model.matrix(y~.,data=data.all)[,-1] # define predictor matrix
# excl intercept col of 1s
x.train = x[-test,] # define training predictor matrix
x.test = x[test,] # define test predictor matrix
y = data.all$y # define response variable
y.train = y[-test] # define training response variable
y.test = y[test] # define test response variable
n.train = dim(data.train)[1] # training sample size
n.test = dim(data.test)[1] # test sample size
```

Project Requirements

Fit the following models to the training set. For each model extract the model coefficient estimates, predict the responses for the test set, and calculate the "mean prediction error" in the test set.

1. Least squares regression model using all ten predictors.

```
fit1   = lm(y ~ ., data = data.train)
coef1  = coef(fit1)
ypred1 = predict(fit1, newdata = data.test)
mse1   = mean((y.test - ypred1)^2)
### coefficients of linear regression
coef1
```

```
## (Intercept)         age         sex         bmi         map          tc
##  151.884956   -6.387052 -257.173483  513.829632  335.714377 -779.357498
##         ldl         hdl         tch         ltg         glu
##  481.739261   85.035712  262.487358  649.499673  117.225546
```

```
### test error of the linear regression model
mse1
```

```
## [1] 2511.981
```

2. Apply best subset selection using BIC to select the number of predictors.

```
library(leaps)

### predict function for regsubsets object
predict.regsubsets = function(object, newdata, id, ...){
form = as.formula(~.)
mat = model.matrix(form, newdata)
coefi = coef(object, id)
xvars = names(coefi)
mat[, xvars] %*% coefi
}

fit2 = regsubsets(y ~., data = data.train, nvmax = 10)
### number of predictors chosen by BIC
df.bic = which.min(summary(fit2)$bic)
coef2 = coef(fit2, id = df.bic)
ypred2 = predict.regsubsets(fit2, newdata = data.test, id = df.bic)
mse2 = mean((y.test - ypred2)^2)
### coefficients of the best subset selection by BIC
coef2
```

```
## (Intercept)         sex         bmi         map         hdl         ltg
##    151.8015   -242.5765    530.5897    346.6307   -353.9840    426.2500
```

```
### test error of the best subset selection by BIC
mse2
```

```
## [1] 2506.565
```

3. Apply best subset selection using 10-fold cross-validation to select the number of predictors (Use a random number seed of 38723 immediately before entering the command

```r
set.seed(38723)
k = 10

# folds = sample(1:k, nrow(data.train), replace = TRUE)

# I realize you used the code above from the book,
# but here's my preferred syntax for getting more even sized k-folds.
# with small data sets like this, the above sampling method from the book
# can give relatively large differences between fold sizes > 50%(!)
 folds = vector("list", k)
 possSet = c(1:nrow(data.train))
 for(a in 1:10){
    # to deal with uneven split
     if(a < 3){
         folds[[a]] = sample(possSet, ceiling(nrow(data.train) / 10), replace = FALSE)
         possSet = setdiff(possSet, folds[[a]])
    } else {
        folds[[a]] = sample(possSet, floor(nrow(data.train) / 10), replace = FALSE)
        possSet = setdiff(possSet, folds[[a]])
    }
}
vmse = rep(0, k)

for(j in 1:k){
   dataj  = data.train[setdiff(c(1:nrow(data.train)), folds[[a]]), ]
   validj = data.train[folds[[a]], ]

  # dataj = data.train[folds != j, ]
  # validj = data.train[folds == j, ]
  fitj   = regsubsets(y ~., data = dataj, nvmax = 10)

  for(i in 1:10){
    yhat = predict.regsubsets(fitj, newdata = validj, id = i)
    vmse[i] = vmse[i] + mean((validj$y - yhat) ^ 2)
  }
}
### number of predictors chosen by CV
df.cv = which.min(vmse)
fit3 = regsubsets(y ~ ., data = data.train, nvmax = 10)
coef3 = coef(fit3, id = df.cv)
ypred3 = predict(fit3, newdata = data.test, id = df.cv)
mse3 = mean((y.test - ypred3)^2)
### coefficients of the best subset selection by CV
coef3
```

```
## (Intercept)         sex         bmi         map          tc         ldl
##    151.8519   -258.9284    512.7940    333.2363   -636.4516    380.6391
##         tch         ltg         glu
##    219.0034    600.8631    117.5116
```

```
### test error of the best subset selection by CV
mse3
```

```
## [1] 2515.08
```

4. Ridge regression model using 10-fold cross-validation to select the largest value of λ such that the cross-validation error is within 1 standard error of the minimum (R functions glmnet and cv.glmnet in package glmnet). Use a random number seed of 1337 immediately before entering the command

```
set.seed(38723)
cv.ridge = cv.glmnet(x.train, y.train, alpha = 0, nfolds = 10)
lam.ridge = cv.ridge$lambda.1se
### lambda chosen by 10-fold CV for ridge regresion
lam.ridge
```

```
## [1] 60.7213
```

```
fit4 = glmnet(x.train, y.train, alpha = 0, lambda = lam.ridge)
coef4 = coef(fit4)
ypred4 = predict(fit4, newx = x.test)
mse4 = mean((y.test - ypred4)^2)
### coefficients of the ridge regression
coef4
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept)  152.103282
## age           29.830187
## sex         -104.518289
## bmi          332.784872
## map          220.833300
## tc             4.914093
## ldl          -20.507126
## hdl         -178.515168
## tch          148.632961
## ltg          255.162238
## glu          133.349908
```

```
### test error of the ridge regression
mse4
```

```
## [1] 2854.064
```

5. Lasso model using 10-fold cross-validation to select the largest value of λ such that the cross-validation error is within 1 standard error of the minimum (R functions glmnet and cv.glmnet in package glmnet). Use a random number seed of 1337 immediately before entering the command

```
set.seed(38723)
cv.lasso = cv.glmnet(x.train, y.train, alpha = 1, nfolds = 10)
lam.lasso = cv.lasso$lambda.1se
### lambda chosen by 10-fold CV for lasso regresion
lam.lasso
```

```
## [1] 7.662159
```

```r
fit5 = glmnet(x.train, y.train, alpha = 1, lambda = lam.lasso)
coef5 = coef(fit5)
ypred5 = predict(fit5, newx = x.test)
mse5 = mean((y.test - ypred5)^2)
### coefficients of the ridge regression
coef5
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## (Intercept)  152.33213
## age            .
## sex            .
## bmi          490.85960
## map          192.15719
## tc             .
## ldl            .
## hdl         -146.84151
## tch            .
## ltg          381.12187
## glu           23.89108
```

```r
### test error of the ridge regression
mse5
```

```
## [1] 2599.724
```

Write up your results in a professional report, like you would present to a client or internal customer for your analysis. The report should be no more than 4 single-spaced pages long and submitted in PDF format.

It should include coefficient estimates for each model and test data mean prediction errors.

Include any other details from your analysis that you feel are worthy of mention.

The report should have sections (e.g., Introduction, Analysis, Results, Conclusion) and provide sufficient details that anyone with a reason able statistics background could understand exactly what you've done and what you concluded.

Consider using tables and figures to enhance your report. You might use the package "pander" if you are using Rmarkdown for nicely formatted tables.

Do not embed R code in the body of your report (if you are using rmarkdown, use {r echo=FALSE} to supress the printing of the r code), but instead attach the code in an appendix. The appendix does not count towards the page limit.

Grading criteria (out of 25)

15 points: fulfilling the project requirements and matching results exactly (this is why you should use the specific random number generator seeds above).

10 points: the quality of your report (including: clarity of writing, organization, and layout; appropriate use of tables and figures; careful proof-reading; adherence to report guidelines