Project I - Linear Regression, Variable Selection, Ridge Regression and Lasso
STAT 897D (Applied Data Mining) – Fall 2015 by XIANG WANG

Introduction

In this project, the diabetes dataset from R's lars library or Efron et al. (2003) was used to fit five different models which include least square regression, best subset selection using BIC, best subset selection using 10-fold cross-validation, ridge and lasso methods. The evaluation of model fit and accuracy was measured by comparing the mean squared prediction errors and their standard errors.

Analysis and Results

Data preparation and exploration

The data used here is part of R's lars package. There are three variables and we are only interested in x and y. The prediction variable x is a variable matrix consisting of ten columns or variables (age, sex, bmi, map, tc, ldl, hdl, tch, ltg, glu); the response variable y consists of one column with 442 observations or patients. A random seed of 1306 was used to ensure the reproducibility of the project results. The data was then partitioned into two sets of data with a 3:1 ratio (training data, 75% roughly; testing data, 25% roughly).

Model building process

Each of the five models shares the same process of model building with the specific steps below.

a) The same training data was used to generate each model using five methods.
b) Then the models above were used to predict the response variable y using the test data.
c) The mean squared prediction errors and their standard errors were calculated.

Model 1: Least Square Regression

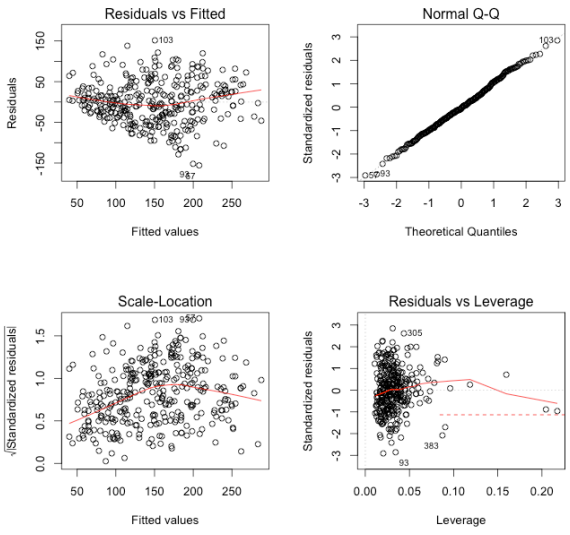
R's lm function was used to generate the least square model on all ten predictors. The following figure shows the coefficient estimates and their corresponding p-values as well as the residuals plots for checking regression assumptions.

```
Call:
lm(formula = y ~ ., data = data.train)

Residuals:
     Min       1Q   Median       3Q      Max
-155.726  -36.065   -2.758   35.039  151.509

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  149.920      2.976  50.382  < 2e-16 ***
age          -66.758     68.946  -0.968  0.33364
sex         -304.651     69.847  -4.362 1.74e-05 ***
bmi          518.663     76.573   6.773 6.01e-11 ***
map          388.111     72.755   5.335 1.81e-07 ***
tc          -815.268    537.549  -1.517  0.13034
ldl          387.604    439.162   0.883  0.37811
hdl          162.903    269.117   0.605  0.54539
tch          323.832    186.803   1.734  0.08396 .
ltg          673.620    206.888   3.256  0.00125 **
glu           94.219     79.590   1.184  0.23737
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 54.05 on 321 degrees of freedom
Multiple R-squared:  0.5213,    Adjusted R-squared:  0.5064
F-statistic: 34.96 on 10 and 321 DF,  p-value: < 2.2e-16
```

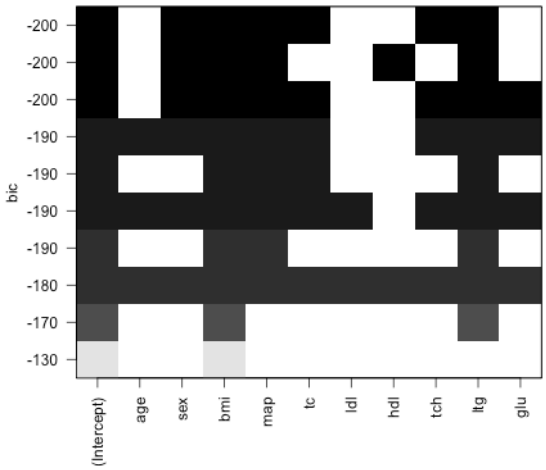


We can see that the four predictors (sex, bmi, map and ltg) are considered significant predictors with p-values smaller than 0.05. The plots also suggest that the residuals are roughly normally distributed. The multiple R-squared is 0.5213, and the adjusted R-squared is 0.5064. The F-statistic is 34.96 on 10 and 321 DF, p-value: < 2.2e-16. The mean squared predicted error is calculated as 3111.3 and its standard error is 361.1.

Model 2: Best Subset Regression using BIC to select the number of predictors

Since there is no built-in predict() function for regsubsets, function of predict.regsubsets() for best subset selection is defined using the code given in textbook.

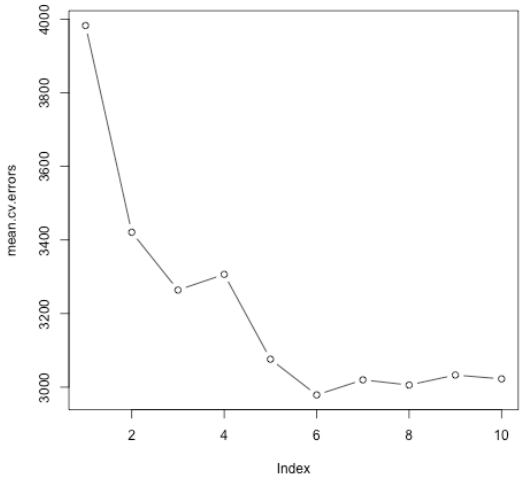
The regsubsets function (part of leaps package) was used to generate models with one through ten predictors. The model showing the lowest BIC (-201.1) has six predictors (also shown in the figure on the right). Their coefficients and estimates are: sex (-306.0), bmi (538.8), map (389.1), tc (-379.0), tch (332.7) and ltg (527.6).



From the predicted responses, the mean squared predicted error is calculated as 3095.5 and its standard error is computed to be 369.8.

Model 3: Best Subset Regression using 10-fold cross-validation to select the number of predictors

The regsubsets function (part of leaps package) was used to generate models with one through ten predictors. Training set is used to perform the 10-fold cross-validation to determine the number of predictors that would produces the lowest training mean cross-validation error. The model that had the lowest cross-validation error is a model with **six** predictors with the lowest mean cross-validation error (2979) (also shown in the figure on the right). This model's coefficients and their estimates are: sex (-306.0), bmi (538.8), map (389.1), tc (- 379.0), tch (332.7) and ltg (527.6).

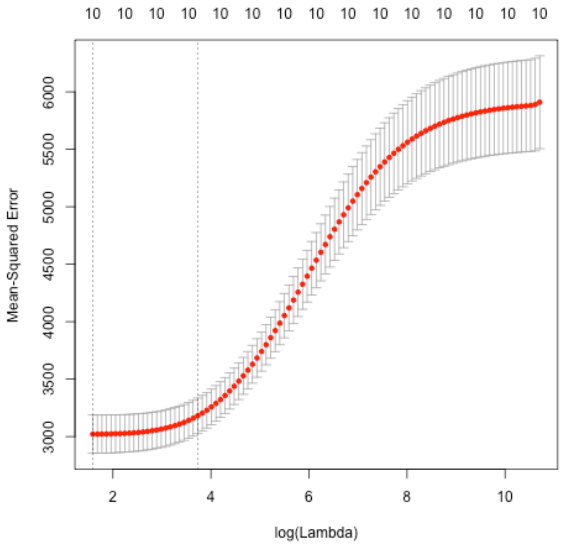


From the predicted responses, the mean squared predicted error is calculated as 3095.5 and its standard error is computed to be 369.8.

Model 4: Ridge Regression using 10-fold cross-validation to select the largest λ value with CV error within 1 SE of min

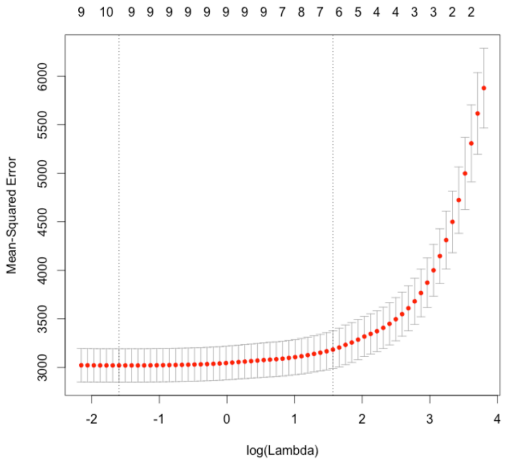
The cv.glmnet function (part of R glmnet package) was used to generate a model that is set to 10-fold CV and alpha=0 to determine the largest λ value with CV error within 1 SE of the

minimum. The model obtained is also shown in the figure on the right.
It is notable that most of the coefficients are smaller than the previous ones, which actually implies the shrinkage method works. This model has ten predictors with λ value to be 41.67. This model's predictor coefficients and their estimates are age (-11.29), sex (-156.90), bmi (374.45), map (264.86), tc (-32.09), ldl (-66.98), hdl (-173.82), tch (124.04), ltg (307.73), and glu (134.52). The model is then used to predict responses on the test dataset. From the predicted responses, the mean squared predicted error is computed to be 3070.9 and its standard error is computed to be 350.6.



Lasso using 10-fold cross-validation to select largest Lambda value with CV error within 1 SE of min

Similarly, the cv.glmnet function (part of R glmnet package) was used to generate a lasso model that is set to 10-fold CV and alpha=1 to determine the largest λ value with CV error within 1 SE of the minimum. This model chosen only contains six predictors with λ value to be 4.79. The model's predictor coefficients and their estimates are sex (-119.65), bmi (501.49), map (270.92), hdl (-180.30), ltg (390.57) and glu (16.61). From the predicted responses, the mean squared predicted error is 2920.1 and its standard error is 346.2.



Summary and Conclusion

The key results of the above five models are summarized in the following table.

Model	Predictors	Number of Predictors	Mean squared prediction error (MSE)	Standard error of MSE
Least Square	All	10	3111.3	361.1
Best Subset (BIC)	sex, bmi, map, tc, tch, ltg	6	3095.5	369.8
Best Subset (10-fold CV)	sex, bmi, map, tc, tch, ltg	6	3095.5	369.8
Ridge Regression	All	10	3070.9	350.6
Lasso	sex, bmi, map, hdl, ltg, glu	6	2920.1	346.2

In general, I think the mean squared prediction errors and their standard errors are fairly close to each other although they vary at some extent. From the points of model complexity and interpretation, BIC and 10-fold cross-validation of best subset models as well as well as the lasso model are the simplest or optimal models. Considering that the Lasso model had the smallest prediction error and standard error, it is the best model I prefer to.

Appendix: R Codes and Results

```
library(lars)                # The data is available in R package "lars"

library(leaps)               # The package "leaps" was used to perform best subset selection by functi
on regsubsets()
library(glmnet)              # The package "glmnet" was used to perform Ridge Regression and Lasso
data(diabetes)
data.all <- data.frame(cbind(diabetes$x, y=diabetes$y))

# Partition the patients into two groups: training (75%) and test (25%)
n <- dim(data.all)[1]          # sample size = 442
set.seed(1306)                 # 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

# Model 1: Least squares regression model using all ten predictors
OLS <- lm(y~., data.train); summary(OLS)   # fit the least squres model using training data
## Call:
## lm(formula = y ~ ., data = data.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -155.726  -36.065   -2.758   35.039  151.509
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  149.920      2.976  50.382  < 2e-16 ***
## age          -66.758     68.946  -0.968  0.33364
## sex         -304.651     69.847  -4.362 1.74e-05 ***
## bmi          518.663     76.573   6.773 6.01e-11 ***
## map          388.111     72.755   5.335 1.81e-07 ***
## tc          -815.268    537.549  -1.517  0.13034
## ldl          387.604    439.162   0.883  0.37811
## hdl          162.903    269.117   0.605  0.54539
## tch          323.832    186.803   1.734  0.08396 .
## ltg          673.620    206.888   3.256  0.00125 **
## glu           94.219     79.590   1.184  0.23737
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 54.05 on 321 degrees of freedom
## Multiple R-squared:  0.5213, Adjusted R-squared:  0.5064
## F-statistic: 34.96 on 10 and 321 DF,  p-value: < 2.2e-16
```

```
pred.OLS <- predict(OLS, data.test)       # predict the response using test data
mean((y.test-pred.OLS)^2)                 # calculate the mean prediction error (MSE)
```

```
## [1] 3111.265
```

```
sd((y.test-pred.OLS)^2)/sqrt(n.test)      # calculate the standard error of MSE
```

```
## [1] 361.0908
```

```
par(mfrow=c(2,2)); plot(OLS)              # Check assumptions using residual and Q-Q plots
```

 # Model 2: Best subset selection using BIC to select the number of predictors

```
BIC <- regsubsets(y~., data=data.train, nvmax=10) # fit best subset model using training data
summary(BIC)                                      # summarize the best subset model
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ ., data = data.train, nvmax = 10)
## 10 Variables  (and intercept)
##     Forced in Forced out
## age     FALSE     FALSE
## sex     FALSE     FALSE
## bmi     FALSE     FALSE
## map     FALSE     FALSE
## tc      FALSE     FALSE
## ldl     FALSE     FALSE
## hdl     FALSE     FALSE
## tch     FALSE     FALSE
## ltg     FALSE     FALSE
## glu     FALSE     FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: exhaustive
##          age sex bmi map tc  ldl hdl tch ltg glu
## 1  ( 1 ) " " " " "*" " " " " " " " " " " " " " "
## 2  ( 1 ) " " " " "*" " " " " " " " " " " "*" " "
## 3  ( 1 ) " " " " "*" "*" " " " " " " " " "*" " "
## 4  ( 1 ) " " " " "*" "*" "*" " " " " " " "*" " "
## 5  ( 1 ) " " "*" "*" "*" " " " " "*" " " "*" " "
## 6  ( 1 ) " " "*" "*" "*" "*" " " " " "*" "*" " "
## 7  ( 1 ) " " "*" "*" "*" "*" " " " " "*" "*" "*"
## 8  ( 1 ) "*" "*" "*" "*" "*" " " " " "*" "*" "*"
## 9  ( 1 ) "*" "*" "*" "*" "*" "*" " " "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

```r
which.min(summary(BIC)$bic) # determine the number of variables with the lowest BIC value
```

```
## [1] 6
```

```r
coef(BIC, id=6)                    # show the coefficient estimates for the model above
```

```
## (Intercept)       sex       bmi       map        tc       tch
##   150.1166  -306.0420   538.8274   389.0673  -379.0379   332.6735
##       ltg
##   527.5658
```

```r
# Create the function to use predict() with regsubsets
predict.regsubsets=function(object,newdata,id,...){
  form=as.formula(object$call[[2]])
  mat=model.matrix(form,newdata)
  coefi=coef(object,id=id)
  xvars=names(coefi)
  mat[,xvars]%*%coefi
}

pred.BIC <- predict(BIC, data.test, id=6)      # predict the response using test data
mean((y.test-pred.BIC)^2)                      # calculate the mean prediction error (MSE)
```

```
## [1] 3095.483
```

```r
sd((y.test-pred.BIC)^2)/sqrt(n.test)           # calculate the standard error of MSE
```

```
## [1] 369.7526
```

```r
plot(BIC, scale='bic')
```

Model 3: Best subset selection using 10-fold cross-validation to select the number of predictors

```r
k=10; set.seed(1306)                                    # set k fold and random seed

folds <- sample(1:k, nrow(data.train), replace=TRUE)    # define each fold
cv.errors <- matrix(NA, k, 10, dimnames=list(NULL, paste(1:10)))

for(j in 1:k){
  best=regsubsets(y~., data=data.train[folds!=j,], nvmax=10)
  for(i in 1:10) {
    pred=predict(best,data.train[folds==j,], id=i)
    cv.errors[j, i]=mean((data.train$y[folds==j]-pred)^2)
  }
}
mean.cv.errors <- apply(cv.errors, 2, mean); mean.cv.errors    # calculate the mean CV error
```

```
##       1       2       3       4       5       6       7       8
## 3982.604 3420.948 3263.854 3306.494 3075.927 2978.907 3019.831 3005.795
##       9      10
## 3032.977 3022.442
```

```
which.min(mean.cv.errors) # 6       # determine the number of variables with the lowest CV error

## 6

best <- regsubsets(y~., data=data.train, nvmax=10) # fit best subset model using training data
coef(best, 6)                                      # show the coefficient estimates for the model above

## (Intercept)       sex       bmi       map        tc       tch
##   150.1166  -306.0420   538.8274   389.0673  -379.0379   332.6735
##       ltg
##   527.5658

pred.best <- predict(best,data.test, 6)            # predict the response using test data
mean((y.test-pred.best)^2)                         # calculate the mean prediction error (MSE)

## [1] 3095.483

sd((y.test-pred.best)^2)/sqrt(n.test)              # calculate the standard error of MSE

## [1] 369.7526

plot(mean.cv.errors, type='b')
```

 # Model 4: Ridge regression using 10-fold cross-validation to select the largest value of λ such that the cross-validation error is within 1 SE of the minimum

```
set.seed(1306)
cv.out <- cv.glmnet(x.train, y.train, alpha=0, nfolds=10)
bestλ <- cv.out$lambda.1se; bestλ

## [1] 41.67209

ridge <- glmnet(x.train, y.train,alpha=0)
pred.ridge <- predict(ridge, s=bestλ, newx=x.test)
predict(ridge, type="coefficients", s=bestλ)[1:11, ]

## (Intercept)       age       sex       bmi       map        tc
##   149.99086  -11.25502 -156.90281  374.44565  264.86245  -32.09103
##       ldl       hdl       tch       ltg       glu
##  -66.97779 -173.82190  124.03502  307.72524  134.51753

mean((y.test-pred.ridge)^2)

## [1] 3070.94

sd((y.test-pred.ridge)^2)/sqrt(n.test)

## [1] 350.5565

plot(cv.out)
```

```
# Model 5: Lasso regression using 10-fold cross-validation to select the largest value of λ such that the cross-validation error is within 1 SE of the minimum
set.seed(1306)
cv.out <- cv.glmnet(x.train, y.train, alpha=1, nfolds=10)
bestλ <- cv.out$lambda.1se; bestλ

## [1] 4.791278

lasso <- glmnet(x.train, y.train, alpha =1)
pred.lasso <- predict(lasso, s=bestλ, newx=x.test, exact=T)

predict(lasso, type="coefficients", s=bestλ)[1:11, ]

(Intercept)       age       sex       bmi       map        tc       ldl       hdl

 149.95300   0.00000 -119.64893  501.48591  270.92404   0.00000   0.00000 -180.30353

      tch       ltg       glu

  0.00000 390.57448  16.61318
mean((y.test-pred.lasso)^2)

## [1] 2920.051

sd((y.test-pred.lasso)^2)/sqrt(n.test)

## [1] 346.2286

plot(cv.out)
```