Homework8

3/6/2019

Contents

Question 11.1

build a regression model using: 1. Stepwise regression 2. Lasso 3. Elastic net

Stepwise Regression

```
set.seed(42)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(MASS)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-16
```

```
uscrime <- read.delim("~/Documents/R/GeorgiaTech/DataPreparation/uscrime.txt")
# AIC uses forward and backward
# first start with all predictors
full.model <- lm(Crime ~., data = uscrime)
# R^2 = 0.71, RSE = 209.1
summary(full.model)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -395.74  -98.09   -6.69  112.99  512.67
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## M            8.783e+01  4.171e+01   2.106 0.043443 *
## So          -3.803e+00  1.488e+02  -0.026 0.979765
## Ed           1.883e+02  6.209e+01   3.033 0.004861 **
## Po1          1.928e+02  1.061e+02   1.817 0.078892 .
## Po2         -1.094e+02  1.175e+02  -0.931 0.358830
## LF          -6.638e+02  1.470e+03  -0.452 0.654654
## M.F          1.741e+01  2.035e+01   0.855 0.398995
## Pop         -7.330e-01  1.290e+00  -0.568 0.573845
## NW           4.204e+00  6.481e+00   0.649 0.521279
## U1          -5.827e+03  4.210e+03  -1.384 0.176238
## U2           1.678e+02  8.234e+01   2.038 0.050161 .
## Wealth       9.617e-02  1.037e-01   0.928 0.360754
## Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
## Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
## Time        -3.479e+00  7.165e+00  -0.486 0.630708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 209.1 on 31 degrees of freedom
## Multiple R-squared:  0.8031, Adjusted R-squared:  0.7078
## F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
```

```
# now apply Stepwise in both directions
step.model <- stepAIC(full.model, direction = "both", trace = FALSE)
# R^2 = 0.744 with 8 predictors and RSE = 195.5
summary(step.model)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob,
##     data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -444.70 -111.07    3.03  122.15  483.30
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6426.10    1194.61  -5.379 4.04e-06 ***
## M              93.32      33.50   2.786  0.00828 **
## Ed            180.12      52.75   3.414  0.00153 **
## Po1           102.65      15.52   6.613 8.26e-08 ***
## M.F            22.34      13.60   1.642  0.10874
## U1          -6086.63    3339.27  -1.823  0.07622 .
## U2            187.35      72.48   2.585  0.01371 *
## Ineq           61.33      13.96   4.394 8.63e-05 ***
## Prob        -3796.03    1490.65  -2.547  0.01505 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 195.5 on 38 degrees of freedom
## Multiple R-squared:  0.7888, Adjusted R-squared:  0.7444
## F-statistic: 17.74 on 8 and 38 DF,  p-value: 1.159e-10
```

```r
# now we can remove the predictors with p-values higher than 0.5
# remove M.F and U1.
step.model_pruned <- lm(Crime ~ M + Ed + Po1 + U2+ Ineq + Prob, data = uscrime)
# R^2 = 0.74
summary(step.model_pruned)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.68  -78.41  -19.68  133.12  556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
## M             105.02      33.30   3.154  0.00305 **
## Ed            196.47      44.75   4.390 8.07e-05 ***
## Po1           115.02      13.75   8.363 2.56e-10 ***
## U2             89.37      40.91   2.185  0.03483 *
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## Prob        -3801.84    1528.10  -2.488  0.01711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200.7 on 40 degrees of freedom
## Multiple R-squared:  0.7659, Adjusted R-squared:  0.7307
## F-statistic: 21.81 on 6 and 40 DF,  p-value: 3.418e-11
```

```r
# now apply cv
train.control <- trainControl(method = "cv", number = 10)
step.model_pruned <- train(Crime ~ M + Ed + Po1 + U2+ Ineq + Prob, data = uscrime,
                    method = "lmStepAIC",
                    trControl = train.control,
                    trace = FALSE
                    )
# Model accuracy
step.model_pruned$results
```

```
##   parameter    RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      none 226.321 0.7235632 180.3997 91.88231  0.2561513 66.88053
```

```r
# Final model coefficients
step.model_pruned$finalModel
```

```
##
## Call:
## lm(formula = .outcome ~ M + Ed + Po1 + U2 + Ineq + Prob, data = dat)
##
## Coefficients:
## (Intercept)            M           Ed          Po1           U2
##    -5040.50       105.02       196.47       115.02        89.37
##        Ineq         Prob
##       67.65     -3801.84
```

```r
# Summary of the model
summary(step.model_pruned$finalModel)
```

```
##
## Call:
## lm(formula = .outcome ~ M + Ed + Po1 + U2 + Ineq + Prob, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.68  -78.41  -19.68  133.12  556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
## M             105.02      33.30   3.154  0.00305 **
## Ed            196.47      44.75   4.390 8.07e-05 ***
## Po1           115.02      13.75   8.363 2.56e-10 ***
## U2             89.37      40.91   2.185  0.03483 *
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## Prob        -3801.84    1528.10  -2.488  0.01711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200.7 on 40 degrees of freedom
## Multiple R-squared:  0.7659, Adjusted R-squared:  0.7307
## F-statistic: 21.81 on 6 and 40 DF,  p-value: 3.418e-11
```

Discussion Stepwise

stepAIC chooses the best model by AIC. It has an option named direction, which can take the following values: i) "both" (for stepwise regression, both forward and backward selection); "backward" (for backward selection) and "forward" (for forward selection). It return the best final model.

In this case forward selection means adding predictors starting from 1 predictor and backwards selection means removing predictor by predictor starting from all available predictors in the linear regression. Stepwise uses the combination of this two techinques.

In this problem I started with all predictors with a reported adjusted $R^2 = 0.71$. Then with StepAIC applied the reported adjusted R^2 was 0.744. This is better than the previous model and uses fewer predictors (8). Now use the p-values to eliminate the predictors with higher p-value 0.05. So remove predictor M.F and U1.

This results in a final lm model with adjusted $R^2 = 0.74$. To report the final quality of the model apply cross-validation to the model, which results in a $R^2 = 0.73$

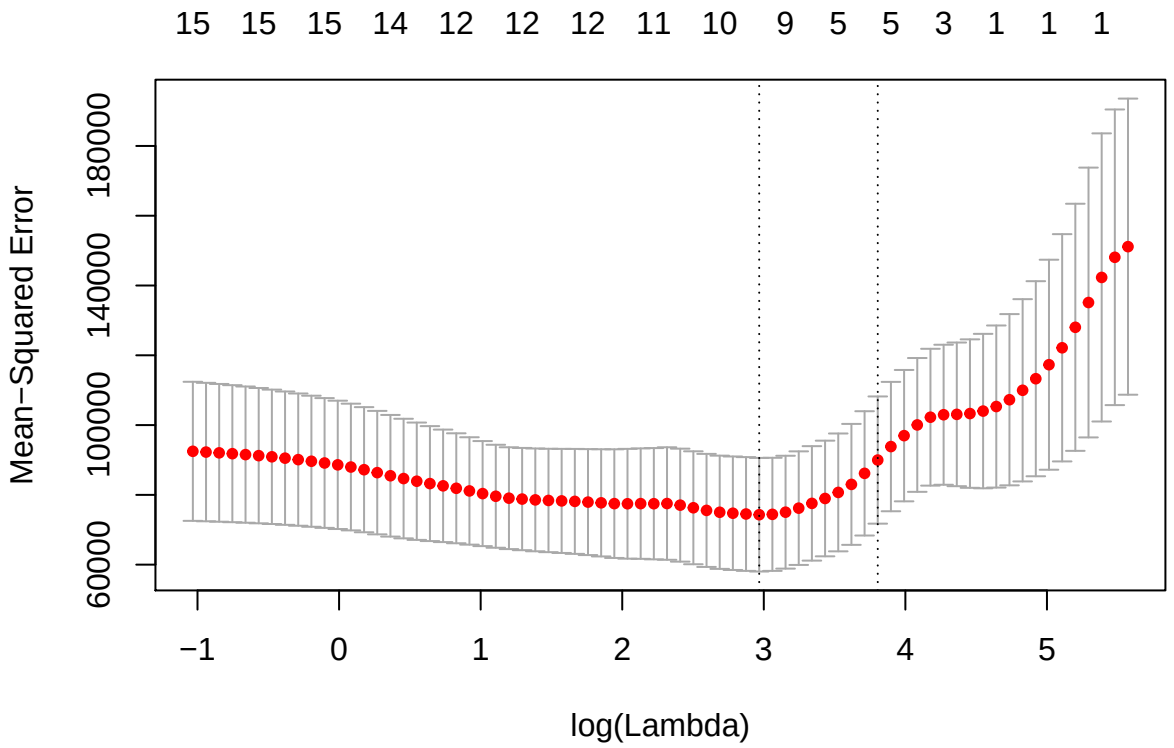
The resulting regression is `y = 105M + 196.47Ed + 115Po1 + 89U2 + 67Ineq - 3801Prob`.

Lasso Regression

```r
# scale data
uscrimeScaled = as.data.frame(scale(uscrime[,c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)]))
uscrimeScaled <- cbind(uscrimeScaled, uscrime[,16]) # add column response
colnames(uscrimeScaled)[16] <- "Crime"

# use the cv glmnet
```

```
cv.lasso=cv.glmnet(x=as.matrix(uscrimeScaled[,-16]),y=as.matrix(uscrimeScaled$Crime),alpha = 1, nfolds =
plot(cv.lasso)
```




```
cv.lasso$lambda.min
```

```
## [1] 19.44465
```

```
# display coefficients

coef(cv.lasso, s=cv.lasso$lambda.min)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) 905.085106
## M            66.662502
## So           11.547912
## Ed           67.734367
## Po1         304.125333
## Po2           .
## LF            .
## M.F          50.349502
## Pop           .
## NW            3.241285
## U1            .
## U2           16.917069
## Wealth        .
## Ineq        142.037145
## Prob        -69.399577
## Time          .
```

```
model <- lm(Crime ~M+Po1+M.F+Ineq+Prob, data = uscrimeScaled)
summary(model)
```

```
##
## Call:
## lm(formula = Crime ~ M + Po1 + M.F + Ineq + Prob, data = uscrimeScaled)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -516.2 -117.8   22.6  110.8  445.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      32.15  28.152  < 2e-16 ***
## M              77.27      43.12   1.792  0.08053 .
## Po1           365.04      44.02   8.292 2.64e-10 ***
## M.F            96.53      33.24   2.904  0.00591 **
## Ineq          172.17      49.63   3.469  0.00124 **
## Prob          -95.67      38.09  -2.511  0.01606 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 220.4 on 41 degrees of freedom
## Multiple R-squared:  0.7105, Adjusted R-squared:  0.6752
## F-statistic: 20.13 on 5 and 41 DF,  p-value: 4.316e-10
```

```
# reported R^2 = 0.67
```

Discussion

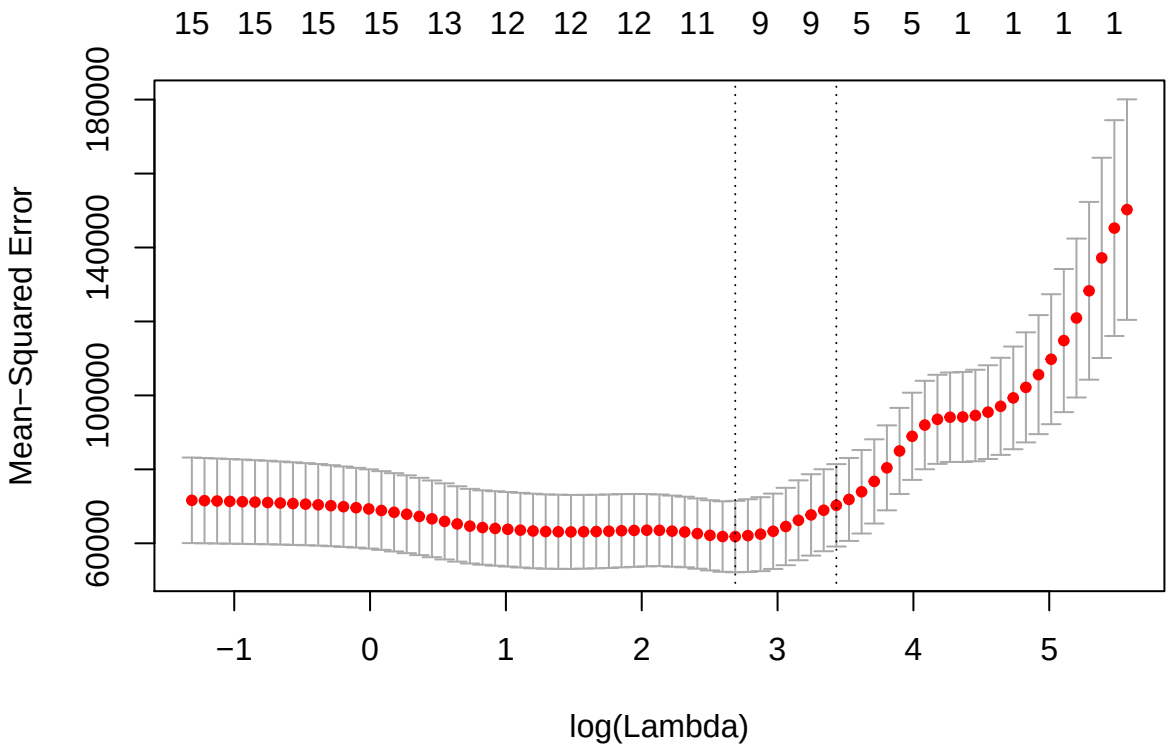
The plot displays the cross-validation error according to the log of lambda. The left dashed vertical line indicates that the log of the optimal value of lambda is approximately -5, which is the one that minimizes the prediction error. This lambda value will give the most accurate model. The chosen lambda for this problem was 2.7562

This model quality was adjusted $R^2 = 0.67$. Which es less than the first problem but it also uses less predictors (simpler model) and the difference of quality is not big.

Elastic Net

```
acum <- c()
for (i in seq(0.1,1,0.1)) {
# use the cv glmnet and test multiple alpha
  elastic.net=cv.glmnet(x=as.matrix(uscrimeScaled[,-16]),y=as.matrix(uscrimeScaled$Crime),alpha = i, nf
  acum = cbind(acum,elastic.net$glmnet.fit$dev.ratio[which(elastic.net$glmnet.fit$lambda == elastic.net

}
alpha = (which.max(acum)-1)/10
plot(elastic.net)
```




```
elastic.net$lambda.min
```

```
## [1] 14.70916
```

```
coef(elastic.net, s=elastic.net$lambda.min)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) 905.085106
## M            77.631433
## So           21.724531
## Ed           98.184745
## Po1         311.196314
## Po2           .
## LF            2.888766
## M.F          46.955309
## Pop           .
## NW            2.653843
## U1            .
## U2           29.301947
## Wealth        .
## Ineq        164.141688
## Prob        -77.051716
## Time          .
```

```
model <- lm(Crime ~M+So+Po1+Po2+LF+M.F+NW+U1+U2+Ineq+Prob, data = uscrimeScaled)
summary(model)
```

```
##
## Call:
## lm(formula = Crime ~ M + So + Po1 + Po2 + LF + M.F + NW + U1 +
##     U2 + Ineq + Prob, data = uscrimeScaled)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -490.30 -142.50   -1.61  136.86  411.28
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      33.69  26.865   <2e-16 ***
## M              85.65      56.22   1.523   0.1366
## So             38.53      75.76   0.509   0.6142
## Po1           352.78     322.52   1.094   0.2815
## Po2           -17.34     330.75  -0.052   0.9585
## LF             42.98      60.02   0.716   0.4787
## M.F            99.08      57.77   1.715   0.0952 .
## NW            -18.96      67.69  -0.280   0.7811
## U1            -61.58      79.90  -0.771   0.4460
## U2             83.90      71.93   1.167   0.2513
## Ineq          138.09      71.90   1.921   0.0629 .
## Prob          -93.59      45.20  -2.071   0.0458 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 231 on 35 degrees of freedom
## Multiple R-squared:  0.7287, Adjusted R-squared:  0.6434
## F-statistic: 8.545 on 11 and 35 DF,  p-value: 4.615e-07
```

```r
# remove by p-values
model <- lm(Crime ~M + Ed + Po1 + U2+ Ineq + Prob, data = uscrimeScaled)
summary(model)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = uscrimeScaled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.68  -78.41  -19.68  133.12  556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      29.27  30.918  < 2e-16 ***
## M             131.98      41.85   3.154  0.00305 **
## Ed            219.79      50.07   4.390 8.07e-05 ***
## Po1           341.84      40.87   8.363 2.56e-10 ***
## U2             75.47      34.55   2.185  0.03483 *
## Ineq          269.91      55.60   4.855 1.88e-05 ***
## Prob          -86.44      34.74  -2.488  0.01711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200.7 on 40 degrees of freedom
## Multiple R-squared:  0.7659, Adjusted R-squared:  0.7307
## F-statistic: 21.81 on 6 and 40 DF,  p-value: 3.418e-11
```

Discussion

The alpha parameter for the elastic net is between 0 and 1. To find the best alpha in this model, it is necessary to iterate through some possible alpha values, from 0.10 to 1 in steps of 0.10. So after this is accomplished, the predictors are discarded by coeficients and after that the p-values are used to leave only the most relevant predictors. The reportes R^2 of this model was 0.73. The same as the first one. And it also uses the same predictors.