Linear Regression and Feature Selection Tutorial and Assignment

Dominik Hoftych

November 26, 2018

```
rm(list = ls())
require(glmnet)
```

```
## Loading required package: glmnet
```

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

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

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

```
set.seed(2) # Set random seed to make the result reproducible

# Load the data from file
data <- read.csv('./data.csv', header=TRUE)
cols = length(data)
x<-as.matrix(data)[,seq(cols-1)]
y<-as.matrix(data)[,cols]

# Split data to train and test sets
train_size <- floor(0.8 * nrow(data)) # Use 80% of the data for training
train <- sample(seq_len(nrow(data)), size = train_size) # Generate indices for training data
test <- (-train)

# Tested lambda values
lambda_grid <- 10^ seq(10 , -3 , length =200)
```

TASK 1

There is a methodological error in the block of code below. Find it and correct it. Hint: The error causes the variable lasso.coefficients contain values of lesser precision than what we could get from the data.

```
# Fit LS
ls.train_model <- lm(Y ~ ., data=data[train,])
ls.prediction <- predict(ls.train_model, data[test,])

# Fit LASSO
lasso.model <- glmnet(x[train,],y[train],alpha=1, lambda=lambda_grid, standardize=TRUE)
lasso.cv.out <- cv.glmnet(x[train,],y[train],alpha=1)
lasso.lambda <- lasso.cv.out$lambda.min
plot(lasso.cv.out)
```

assignment_2018_files/figure-latex/unnamed-chunk-2-1.pdf

```
lasso.prediction <- predict(lasso.model, s=lasso.lambda, newx=x[test,])
lasso.coefficients <- predict(lasso.model, type="coefficients", s=lasso.lambda)

print("LASSO coefficients:")
```

```
## [1] "LASSO coefficients:"
```

```
print(as.matrix(lasso.coefficients))
```

```
##                         1
## (Intercept)  1.022269e+03
## X1           1.267727e+00
## X2          -1.168905e+00
## X3          -2.806243e-01
## X4           0.000000e+00
## X5           0.000000e+00
## X6          -1.272083e+01
## X7           0.000000e+00
## X8           8.706453e-03
## X9           3.793990e+00
## X10         -7.769316e-01
## X11          0.000000e+00
## X12          0.000000e+00
## X13          0.000000e+00
## X14          1.344075e-01
## X15          3.422416e-01
```

```
print(as.matrix(lasso.coefficients)[seq(2,cols),] != 0)
```

```
##    X1    X2    X3    X4    X5    X6    X7    X8    X9   X10   X11   X12
##  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
##   X13   X14   X15
## FALSE  TRUE  TRUE
```

```
# CORRECTION HERE: in order to obtain better results, use the whole dataset to train the model, which w
# leave us with no data left for testing
lasso.model <- glmnet(x,y,alpha=1, lambda=lambda_grid, standardize=TRUE)
# lasso.prediction <- predict(lasso.model, s=lasso.lambda, newx=x[test,])
lasso.coefficients <- predict(lasso.model, type="coefficients", s=lasso.lambda)

print("LASSO coefficients when trained with whole dataset:")
```

```
## [1] "LASSO coefficients when trained with whole dataset:"
```

```
print(as.matrix(lasso.coefficients))
```

```
##                         1
## (Intercept)  1.076901e+03
## X1           1.289876e+00
## X2          -1.156148e+00
## X3          -5.670120e-01
## X4           0.000000e+00
## X5           0.000000e+00
## X6          -1.254274e+01
## X7          -6.438579e-01
## X8           6.464375e-03
## X9           3.667651e+00
## X10          0.000000e+00
## X11          0.000000e+00
## X12          0.000000e+00
## X13          0.000000e+00
## X14          1.930310e-01
## X15          1.488120e-01
```

```
print(as.matrix(lasso.coefficients)[seq(2,cols),] != 0)
```

```
##    X1    X2    X3    X4    X5    X6    X7    X8    X9   X10   X11   X12
##  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE
##   X13   X14   X15
## FALSE  TRUE  TRUE
```

TASK 2

Implement analogous fitting method for Ridge regression. Compute the Mean Squared Error for Ridge regression, LS and LASSO and compare them.

```
rr.model <- glmnet(x[train,],y[train],alpha=0, lambda=lambda_grid, standardize=TRUE)
rr.cv.out <- cv.glmnet(x[train,],y[train],alpha=0)
rr.lambda <- rr.cv.out$lambda.min
plot(rr.cv.out)
```

assignment_2018_files/figure-latex/unnamed-chunk-3-1.pdf

```
rr.prediction <- predict(rr.model, s=rr.lambda, newx=x[test,])
rr.coefficients <- predict(rr.model, type="coefficients", s=rr.lambda)

# Display the coefficients and selected variables
print("RIDGE coefficients:")
```

```
## [1] "RIDGE coefficients:"
```

```
print(as.matrix(rr.coefficients))
```

```
##                         1
## (Intercept) 1253.52442638
## X1             1.78448341
## X2            -1.05180521
## X3            -1.00774352
## X4            -5.67434596
## X5           -12.97145181
## X6           -11.86008483
## X7            -1.47547728
## X8             0.01025190
## X9             2.77918058
## X10           -0.95998696
## X11           -0.37967556
## X12           -0.02566503
## X13            0.08798898
## X14            0.18417798
## X15            0.67083615
```

```
print(as.matrix(rr.coefficients)[seq(2,cols),] != 0)
```

```
##   X1   X2   X3   X4   X5   X6   X7   X8   X9  X10  X11  X12  X13  X14  X15
## TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
#Compute the Mean Squared Error for Ridge regression, LS and LASSO and compare them.
mse.ls <- mean((ls.prediction-y[test])^2)
mse.lasso <- mean((lasso.prediction-y[test])^2)
mse.rr <- mean((rr.prediction-y[test])^2)
cat("Least squares MST:", mse.ls)
```

```
## Least squares MST: 1008.999
```

```
cat("\nLASSO MST:", mse.lasso)
```

```
##
## LASSO MST: 1018.965
```

```
cat("\nRidge Regression:", mse.rr)
```

```
##
## Ridge Regression: 1116.057
```

TASK 3

Assume we want LASSO to select exactly 2 variables while still minimizing MSE. What is then the desired parameter lambda (with 1e-1 precission)? What are the variables? What is the MSE? Check if the selected variables are the same as the ones exhaustive subset search would select. You may use the `regsubsets` function from the `leaps` library to do this or implement the search yourself for subsets of size 2.

```
library(leaps)
ess <- regsubsets(Y ~ .,data = data, method = "exhaustive")
summary(ess)
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = data, method = "exhaustive")
## 15 Variables  (and intercept)
##     Forced in Forced out
## X1      FALSE      FALSE
## X2      FALSE      FALSE
## X3      FALSE      FALSE
## X4      FALSE      FALSE
## X5      FALSE      FALSE
## X6      FALSE      FALSE
## X7      FALSE      FALSE
## X8      FALSE      FALSE
## X9      FALSE      FALSE
## X10     FALSE      FALSE
## X11     FALSE      FALSE
## X12     FALSE      FALSE
## X13     FALSE      FALSE
## X14     FALSE      FALSE
## X15     FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          X1  X2  X3  X4  X5  X6  X7  X8  X9  X10 X11 X12 X13 X14 X15
## 1  ( 1 ) " " " " " " " " " " " " " " " " "*" " " " " " " " " " " " "
## 2  ( 1 ) " " " " " " " " " " "*" " " " " "*" " " " " " " " " " " " "
## 3  ( 1 ) " " "*" " " " " " " "*" " " " " "*" " " " " " " " " " " " "
## 4  ( 1 ) " " "*" " " " " " " "*" " " "*" "*" " " " " " " " " " " " "
## 5  ( 1 ) "*" "*" " " " " " " "*" " " "*" "*" " " " " " " " " " " " "
## 6  ( 1 ) "*" "*" " " " " " " "*" " " "*" "*" " " " " " " " " "*" " "
## 7  ( 1 ) "*" "*" " " " " " " "*" " " "*" "*" " " " " "*" "*" " " " "
## 8  ( 1 ) "*" "*" "*" " " " " "*" " " "*" "*" " " " " "*" "*" " " " "
```

```
# create my own lambda_grid in some range
my_min <- min(lambda_grid)
my_max <- max(lambda_grid[lambda_grid < 10^3])
my_grid <- seq(from = my_min,to = my_max,by = 0.1)


my_lasso <- cv.glmnet(x[train,],y[train],alpha=1,lambda=my_grid)
my_lambdas <- my_lasso$lambda
mse <- Inf
my_lambdas_best <- 0

for(x in 1:length(my_lambdas)){
  # select only lambdas with 2 variables
  if(my_lasso$nzero[x] == 2){
      if(my_lasso$cvm[x] < mse){
          mse = my_lasso$cvm[x]
          my_lambdas_best <- my_lasso$lambda[x]
      }
  }
}

lasso.coefficients <- predict(my_lasso, type = "coefficients", s = my_lambdas_best)
print("coefficients:")
```

```
## [1] "coefficients:"
```

```
print(as.matrix(lasso.coefficients))
```

```
##                         1
## (Intercept) 1030.147744
## X1             0.000000
## X2             0.000000
## X3             0.000000
## X4             0.000000
## X5             0.000000
## X6           -10.241587
## X7             0.000000
## X8              0.000000
## X9              1.846969
## X10             0.000000
## X11             0.000000
## X12             0.000000
## X13             0.000000
## X14             0.000000
## X15             0.000000
```

```
print(as.matrix(lasso.coefficients)[seq(2,cols),] != 0)
```

```
##    X1    X2    X3    X4    X5    X6    X7    X8    X9   X10   X11   X12 
## FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE 
##   X13   X14   X15 
## FALSE FALSE FALSE
```

```
# As we can see, both regsubsets function and my subset search found same variables - x6 and x9.
```