

LASSO

Shaoming Song 2012011467

2016 年 1 月 15 日

At first I tried LARS algorithm(Least Angle Regression) to solve LASSO problem, after about 10 hours formulas derivation and coding I found that I made a mistake at the very beginning of provment. So I decided to refer Zhengda Zhu's(2012011539) method,which is called cyclical coordinate descent.

```
##import glmnet package and lars package
library(glmnet)

## Warning: package 'glmnet' was built under R version 3.2.3

## Loading required package: Matrix
## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.2.3

## Loaded glmnet 2.0-2

library(lars)

## Warning: package 'lars' was built under R version 3.2.3

## Loaded lars 1.2

##clean data and import input data
rm(list = ls())
setwd('C:/study/大四上/生物信息学/homework3&fin')
raw_data <- read.table(file = "prostate.data")
raw_data <- t(raw_data)
raw_data <- as.data.frame(raw_data)

raw_x <- as.matrix(raw_data[c(1:8),])
raw_y <- as.matrix(raw_data[c(9),])
raw_x <- t(raw_x)
raw_y <- t(raw_y)

##standardize data, normalize x and make mean_y to zero
x <- scale(raw_x)
sd_x <- attr(x,"scaled:scale")
y <- scale(raw_y,scale=FALSE)
```

```

##some necessary variables
status_v <- array(1,8)
threshold <- 0.2
b <- array(0,8)
num_r <- nrow(x)

while(TRUE){
  ##store old b
  b_temp <- b
  ##calculate residence
  r <- y - x[,which(status_v == 1)] %*% b[which(status_v == 1)]
  for(i in which(status_v == 1)){
    ##fix other i and only adjust one i
    b_i_temp <- b[i]
    r_i <- r + b[i] * x[,i]
    b_i <- (t(x[,i]) %*% r_i) / num_r
    ##choose the max value above zero multiplied with sign(b_i)
    b[i] <- sign(b_i) * max((abs(b_i) - threshold) , 0)
    if(b[i] == 0) status_v[i] <- 0
    ##if b_i is no longer larger than threshold, drop it and choose another i
    ##adjust r with new b_i
    r <- r - x[,i] * (b[i] - b_i_temp)
  }
  if(max(abs(b - b_temp)) < 1e-10) break
  ##when b changes little, finish
}

##because b has never been standardized,it needs to be done so
b <- b / sd_x

```

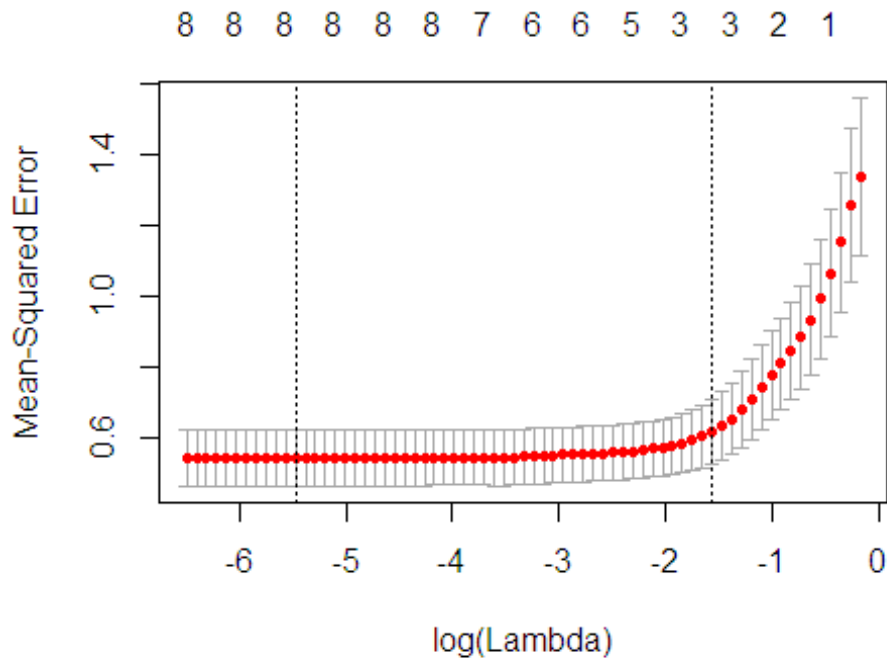
use glmnet package

```

##glmnet solve LASSO by cyclical coordinate descent algorithm
gla <- cv.glmnet(raw_x, raw_y)
la<-gla$lambda.1se

```

The pic of glmnet



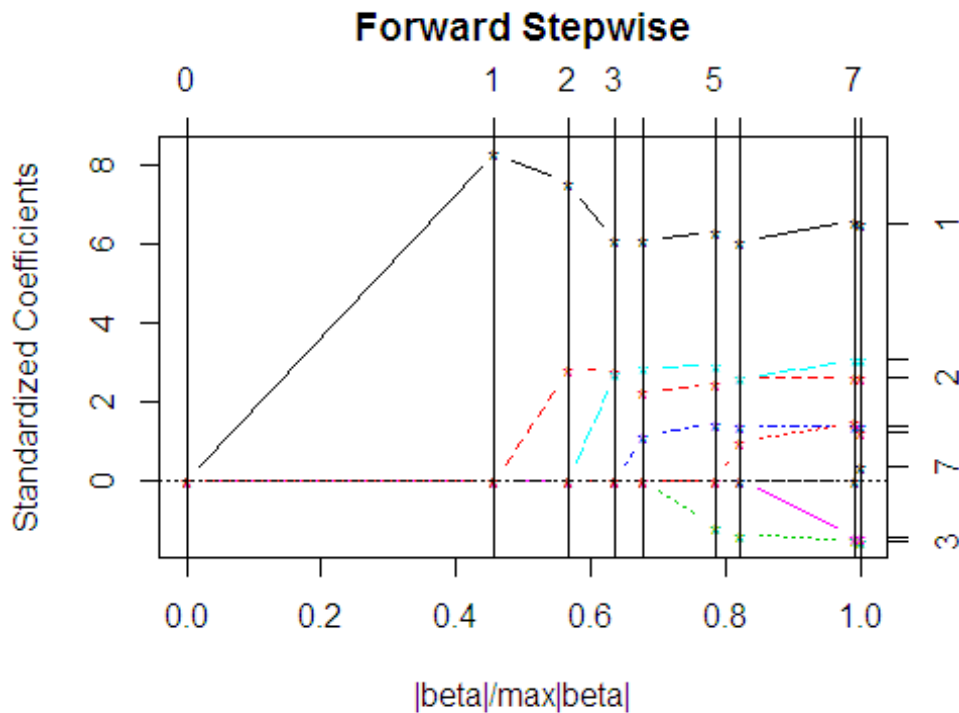
```
##calculate factors  
para<-coef(gla$glmnet.fit, gla$lambda.1se)
```

The answer of my calculation and glmnet

```
## [1] "My answer"  
  
## [1] 0.4458133 0.2953272 0.0000000 0.0000000 0.3555400 0.0000000 0.00  
00000  
## [8] 0.0000000  
  
## [1] "Answer calculated by glmnet"  
  
## 9 x 1 sparse Matrix of class "dgCMatrix"  
##              1  
## (Intercept) 0.7820637  
## lcavol      0.4485189  
## lweight     0.2804033  
## age         .  
## lbph        .  
## svi         0.3383490  
## lcp         .  
## gleason     .  
## pgg45       .
```

use lars package

```
##lars can use stepwise or Lars algorithm to solve LASSO, here I choose  
the former one  
lars_re <- lars(raw_x,raw_y,type = "step")
```



```
coef <- coef.lars(lars_re, mode = "step", s = 4)
```

The answer of LARS

```
## [1] "lars's answer"  
##      lcavol  lweight      age      lbph      svi      lcp  gleaso  
n  
## 0.5258519 0.6617699 0.0000000 0.0000000 0.6656666 0.0000000 0.000000  
0  
##      pgg45  
## 0.0000000
```

Conclusion

We can see that glmnet's answer is almost the same as my own calculation. As about lars, although the number itself is kind of different, we can still figure out that the variables chosen are the same.

Reference

1. Zhengda Zhu's homework(2012011539)

2.

<http://site.douban.com/182577/widget/notes/10567181/note/285262818/>, a webpage explained how to use these two packages

3. PATHWISE COORDINATE OPTIMIZATION, JEROME FRIEDMAN,
TREVOR HASTIE, HOLGER HÖFLING

AND ROBERT TIBSHIRANI