ADA HW6

Liangquan Zhou lz2377

Firday, October 17, 2014

1. Consider the data set `birthwt` in R library MASS. Compare models selected using LASSO and a stepwise procedure to predict 'bwt' birth weight in grams using the following set of predictors:

`age` mother's age in years

`lwt` mother's weight in pounds at last menstrual period

`race` mother's race ('1' = white, '0' = other)

`smoke` smoking status during pregnancy

`ptl` number of previous premature labours

`ht` history of hypertension

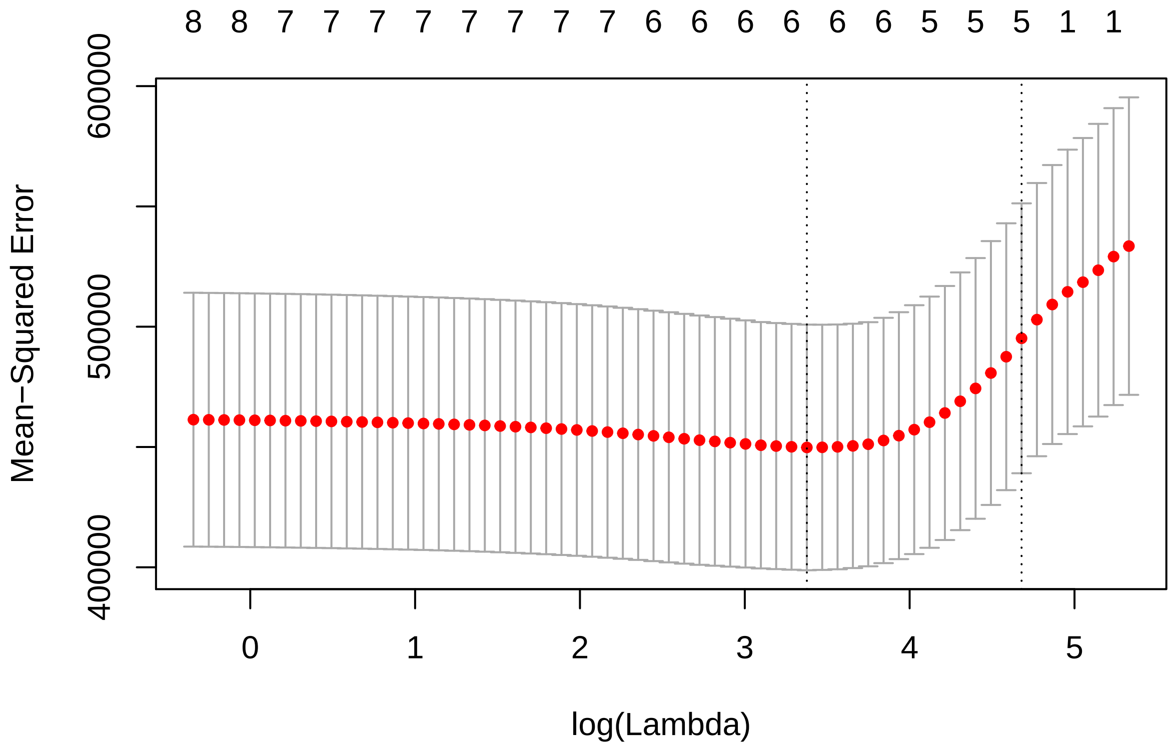
`ui` presence of uterine irritability

`ftv` number of physician visits during the first trimester

```r
library(MASS)
library(glmnet)
data(birthwt)
x = data.matrix(subset(birthwt, select = c(age, lwt, race, smoke, ptl, ht, ui, ftv)))
y = birthwt$bwt

## lasso
fit = glmnet(x, y)
cv.fit = cv.glmnet(x, y)

# plot cv.fit to choose the best lambda
plot(cv.fit)
```



From the plot we can see that the `lambda.min` has a much smaller Mean Squared Error than `lambda.1se`, so we choose the `lambda.min` as the best λ.

So we can get the model coefficients

```
## best lambda value
cv.fit$lambda.min
```

```
## [1] 29.27
```

```
# best  model
model.final <- cv.fit$glmnet.fit
# the best model's coefficients
model.coef <- coef(cv.fit$glmnet.fit, s = cv.fit$lambda.min)
# best model's MSE
pre <-predict(model.final, newx = x,s = cv.fit$lambda.min)
mean((y - pre)^2)
```

```
## [1] 416222
```

We can use `stepAIC` to do the stepwise procedure.

```
## stepwise procedure
birthwt.lm = glm(bwt ~ age + lwt + race + smoke + ptl + ht + ui + ftv, data = birthwt)
# backward
backward = stepAIC(birthwt.lm, direction = c("backward"),trace = F)
```

```
# forward
forward = stepAIC(birthwt.lm, direction = c("forward"),trace = F)
# both
both = stepAIC(birthwt.lm, direction = c("both"),trace = F)
```

We can see that the backward and exhaustive search are keep 5 variables, forward meathod keeps all 8 variables, and lasso keeps 6 variables. But the forward method keeps all the variables.

Check the MSE

```
## lasso
mean((y - pre)^2)
```

```
## [1] 416222
```

```
## backward
mean((y - predict(backward, newx = x))^2)
```

```
## [1] 411855
```

```
## forward
mean((y - predict(forward, newx = x))^2)
```

```
## [1] 411010
```

```
## both
mean((y - predict(both, newx = x))^2)
```

```
## [1] 411855
```

And we can see that the forward model has the best MSE, so this suggests that stepwise models might be over-fitted.

2. For the data set `stackloss` in R, consider the multiple linear regression model of "stack loss" on the other explanatory variables

i). Investigate whether there is any multicollinearity, and suggest remedial measures if appropriate.

Fit the `lm` model, and use

```
dat1 = stackloss
fit1 = lm(stack.loss ~ ., data = stackloss)
library(car)
vif(fit1)
```

```
##   Air.Flow Water.Temp Acid.Conc.
##      2.906      2.573      1.334
```

Since the $VIF > 10$ suggests the multicollinearity, so the model is free from multicollinearity.

ii). Suppose the value of `stack.loss[20]` was changed from 14 to 1500, and those of `Water.Temp[13]` from 18 to 170, and `Acid.Conc.[13]` from 82 to 10.

Change the value

```
dat2 = stackloss
dat2$stack.loss[20] = 1500
dat2$Water.Temp[13] = 170
dat2$Acid.Conc.[13] = 10
```

a).Fit a multiple linear regression model on the new data.

Fit the `lm` model

```
fit2 = lm(stack.loss ~ ., data = dat2)
fit2
```

```
##
## Call:
## lm(formula = stack.loss ~ ., data = dat2)
##
## Coefficients:
## (Intercept)     Air.Flow   Water.Temp   Acid.Conc.
##     1084.67         1.88        -6.28       -11.25
```

b). Identify influential points using `DFFITS`, `DFBETAS`, `Studentized Deleted Residuals` and `Cook's D`.

b). Identify the influential points

```
n = dim(dat2)[1]
p = dim(dat2)[2]
DFFITS = dffits(fit2)
DFBETAS = dfbetas(fit2)
RSTUDENT = rstudent(fit2)
COOK.D = cooks.distance(fit2)

## Use DFFITS to identify influtial points
DFFITS[which(abs(DFFITS) > 2 * sqrt((p+1)/n))]
```

```
##      13      20
##  -2.266 100.752
```

```
## Use DFBETAS to identify influtial points
# for coefficient of Intercept
DFBETAS[,1][DFBETAS[,1] > 2 / sqrt(n)]
```

```
##    20
## 68.43
```

```
# for coefficient of Air.Flow
DFBETAS[,2][DFBETAS[,2] > 2 / sqrt(n)]
```

```
##    20
## 5.985
```

```
# for coefficient of Water.Temp
DFBETAS[,3][DFBETAS[,3] > 2 / sqrt(n)]
```

```
##     17
## 0.6345
```

```
# for coefficient of Acid.Conc.
DFBETAS[,4][DFBETAS[,4] > 2 / sqrt(n)]
```

```
##     17
## 0.6389
```

```
## Use Studentized Deleted Residuals to identify influtial points
RSTUDENT[abs((RSTUDENT)) > abs(qt(0.05/(2*n), n-p-2))]
```

```
##    20
## 327.4
```

```
## Use Cooks Distance to identify influtial points
COOK.D [which(COOK.D>qf(0.05, p+1, n-p-1))]
```

```
##     13     20
## 1.3630 0.4023
```

c). Compare the estimates of the regression coefficients obtained before and after the above changes for each of the following:

OLS

```
formula = stack.loss ~ .
## OLS
fit1 = lm(formula, data = dat1)
fit2 = lm(formula, data = dat2)
fit1$coef
```

```
## (Intercept)    Air.Flow  Water.Temp  Acid.Conc.
##    -39.9197      0.7156      1.2953     -0.1521
```

```
fit2$coef
```

```
## (Intercept)    Air.Flow  Water.Temp  Acid.Conc.
##    1084.671       1.881      -6.281     -11.250
```

Least median of squares regression

```
## Least median of squares regression
fit1 = lmsreg(formula, data = dat1)
fit2 = lmsreg(formula, data = dat2)
fit1$coef
```

```
## (Intercept)    Air.Flow  Water.Temp  Acid.Conc.
##  -3.425e+01   7.143e-01   3.571e-01   3.838e-16
```

```
fit2$coef
```

```
## (Intercept)    Air.Flow  Water.Temp  Acid.Conc.
##   -37.52720     0.76883     0.28452     0.01674
```

Least trimmed squares robust regression

```
## Least trimmed squares robust regression
fit1 = ltsreg(formula, data = dat1)
fit2 = ltsreg(formula, data = dat2)
fit1$coef
```

```
## (Intercept)    Air.Flow  Water.Temp  Acid.Conc.
##    -34.2917      0.7143      0.3571      0.0000
```

```
fit2$coef
```

```
## (Intercept)    Air.Flow  Water.Temp  Acid.Conc.
##   -37.15409     0.75000     0.32075     0.01887
```

M-estimates of regression with Huber weights

```
## M-estimates of regression with Huber weights
fit1 = rlm(formula, data = dat1, psi = psi.huber)
fit1 = rlm(formula, data = dat2, psi = psi.huber)
fit1$coef
```

```
## (Intercept)    Air.Flow  Water.Temp  Acid.Conc.
##   -37.68258     1.11532    -0.08566    -0.11400
```

```
fit2$coef
```

```
## (Intercept)    Air.Flow  Water.Temp  Acid.Conc.
##   -37.15409     0.75000     0.32075     0.01887
```

From the result we can see that the OLS regressiong coefficients changed a lot, since it's not robust. The least median of squares regression is the procedure with the highest breakdown point, and the result shows it's coefficients be affected least. It's the most robust regression. Least trimmed squares of regression has a relatively high breakdown point, so it's not as robust as least median of squares regression. The M-estimates of regression with Huber weights performs similar as the Least trimmed squares regression.