

ETC3250 Lab 11

Souhaib Ben Taieb

Week 11

Purpose

The goal of this lab is to understand ridge regression and the lasso.

Exercise 1

Read Section 6.2 of ISLR and do the exercise 2 in Section 6.8.

- (a) iii is TRUE
- (b) iii is TRUE
- (c) ii is TRUE

Assignment

We will use ridge regression and the lasso to estimate the salary of various baseball players based on several predictor measurements. This data set is taken from the *ISLR* package. Download the file *hitters.Rdata* at <http://bsouhaib.github.io/BusinessAnalytics/labs/hitters.Rdata>. We will use the implementation of these algorithms available in the *glmnet* package.

Question 1

The *glmnet* function, by default, internally scales the predictor variables so that they will have standard deviation 1, before solving the ridge regression or lasso problems. Explain why such scaling is important in our application.

The variables have different variances. If not scaled, variables with higher variance will be penalized more since they will have higher RSS.

Question 2

Run the following commands:

```
library(glmnet)

load("hitters.Rdata")
grid <- 10^seq(10, -2, length=100)
ridge.model <- glmnet(x, y, lambda = grid, alpha = 0)
lasso.model <- glmnet(x, y, lambda = grid, alpha = 1)
```

- a. Using the help page of the *glmnet* function, briefly describe what the previous two lines are doing. In particular, what is *lambda* and *alpha*?

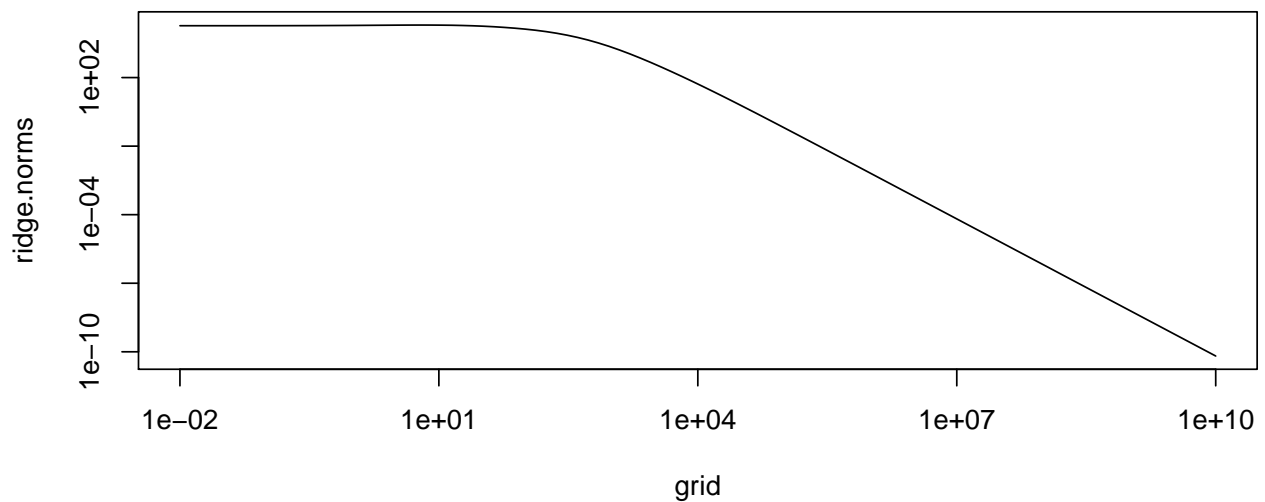
This compute ridge regression and lasso estimates, over the whole sequence of *lambda* values specified by *grid*. The flag *alpha* = 0 notifies glmnet to perform ridge regression, and *alpha*=1 notifies it to perform lasso regression

Question 3

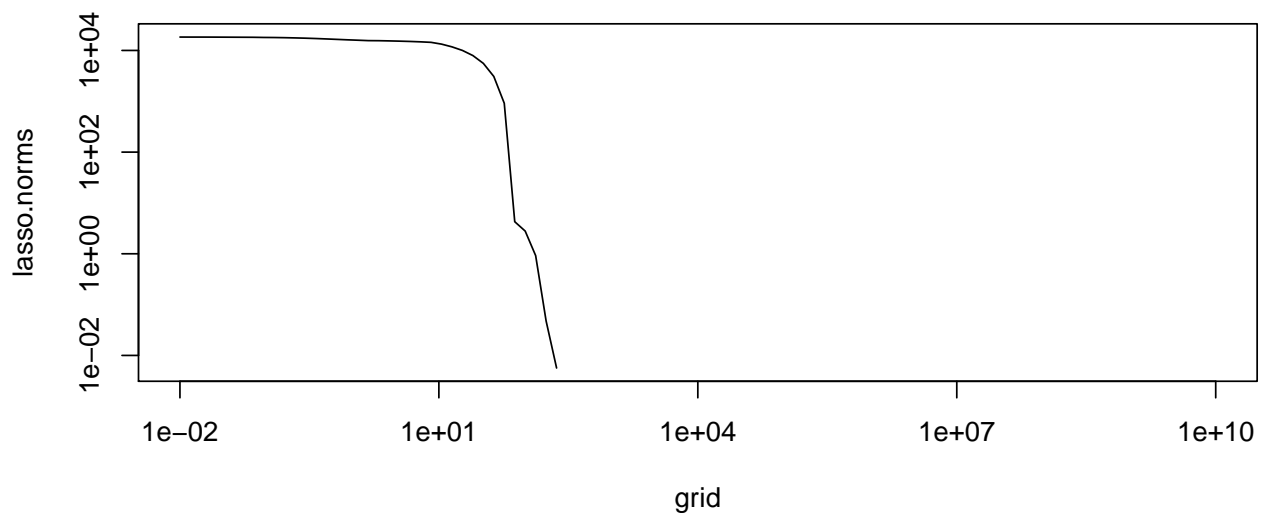
- a. For each model, verify that as λ decreases, the value of the penalty term only increases. In other words, the squared L_2 and the L_1 norm of the coefficients only gets bigger as λ decreases for ridge and the lasso, respectively. The plot should be on a log-log scale.

```
nalpaha <- ncol(ridge.model$beta)
ridge.norms <- lasso.norms <- numeric(nalpaha)

for(ialpha in seq(nalpaha)){
  ridge.norms[ialpha] <- sum(ridge.model$beta[, ialpha]^2)
  lasso.norms[ialpha] <- sum(lasso.model$beta[, ialpha]^2)
}
plot(grid, ridge.norms, type = 'l', log="xy")
```



```
plot(grid, lasso.norms, type = 'l', log="xy")
```



Question 4

- a. For both ridge and the lasso, explain what happens to the coefficients for very small and very large values of λ .

For a very small value of λ , the estimates are very close to the least squares estimate. For a very large value of λ , the estimates approach 0 in all components (except the intercept, which is not penalized by default).

Question 5

- a. For both ridge and lasso, produce a plot of the 5-fold cross-validation error curve as a function of λ , with standard errors drawn, for both the ridge and lasso models. Determine the value of λ that minimize the cross-validation error. You can use `cv.glmnet`.

```
cv.ridge <- cv.glmnet(x, y, nfolds = 5, alpha = 0, lambda = grid)
cv.lasso <- cv.glmnet(x, y, nfolds = 5, alpha = 1, lambda = grid)

idbest_ridge <- match(cv.ridge$lambda.min, grid)
idbest_lasso <- match(cv.lasso$lambda.min, grid)
```

Question 6

- a. For both ridge and lasso, compute the estimates using (1) the best value of λ you obtained in question 5, and (2) the model you fitted in question 2. You can use the `predict` function with `type = "coef"` to compute the estimates for a given model. How do the ridge estimates compare to those from the lasso?

```
best_ridge <- predict(ridge.model, type = "coef")[, idbest_ridge]
best_lasso <- predict(lasso.model, type = "coef")[, idbest_lasso]

print(best_ridge)
# (Intercept)      AtBat      Hits      HmRun      Runs
# 109.2829120 -1.0135354   3.6940734 -0.9796829   0.7058863
#           RBI      Walks      Years      CAtBat      CHits
#   0.5510051   4.0365649 -10.5807895 -0.0138037   0.1611629
#      CHmRun      CRuns      CRBI      CWalks      LeagueN
#   0.7336816   0.4022177   0.2939173 -0.3996671   57.5193908
# DivisionW      PutOuts      Assists      Errors      NewLeagueN
# -124.1385803  0.2714489   0.2139257 -3.8379446 -23.5048569
print(best_lasso)
# (Intercept)      AtBat      Hits      HmRun      Runs
# 146.83733357 -1.84795456   6.36025683   0.53321464  -0.20796869
#           RBI      Walks      Years      CAtBat      CHits
#   0.00000000   5.27692705  -9.37605656  -0.02686293   0.00000000
#      CHmRun      CRuns      CRBI      CWalks      LeagueN
#   0.44385268   0.88762007   0.46369992  -0.66338590  38.73155632
# DivisionW      PutOuts      Assists      Errors      NewLeagueN
# -117.98879627  0.28057238   0.24942119  -2.59606200  -3.40537234
```

We can clearly see that the lasso estimate is sparser than the ridge estimate with 6 coefficients equal to zero.

Question 7

Suppose that you were coaching a young baseball player who wanted to strike it rich in the major leagues. What handful of attributes would you tell this player to focus on?

Look at the lasso coefficients that are non-zero with large positive values.

TURN IN

- Your `.Rmd` file (which should knit without errors and without assuming any packages have been pre-loaded)
- Your Word (or pdf) file that results from knitting the Rmd.
- DUE: 17 October 10am (late submissions not allowed), loaded into moodle