P8106_hw1_xy2395

Jack Yan

2/27/2019

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(caret)
library(glmnet)
library(pls)
set.seed(123123)
```

Introduction

In this homework, 4 regression methods (i.e. least squares, ridge, lasso, and PCR) are implemented to predict the solubility of compounds using their chemical structures. The test errors of the 4 models are compared.

Data Entry

```
train_df <-
  read_csv("./data/solubility_train.csv") %>%
  janitor::clean_names()

test_df <-
  read_csv("./data/solubility_test.csv") %>%
  janitor::clean_names()
```

Model Implementation

Least Squares

```
fit_ls = lm(solubility ~ ., data = train_df)

test_df_ls =
  modelr::add_predictions(test_df, fit_ls) %>%
  mutate(error = solubility - pred)
mse_ls = mean(test_df_ls$error^2)
mse_ls
```

```
## [1] 0.5558898
```

The test mean square error for the least square model is 0.5558898.

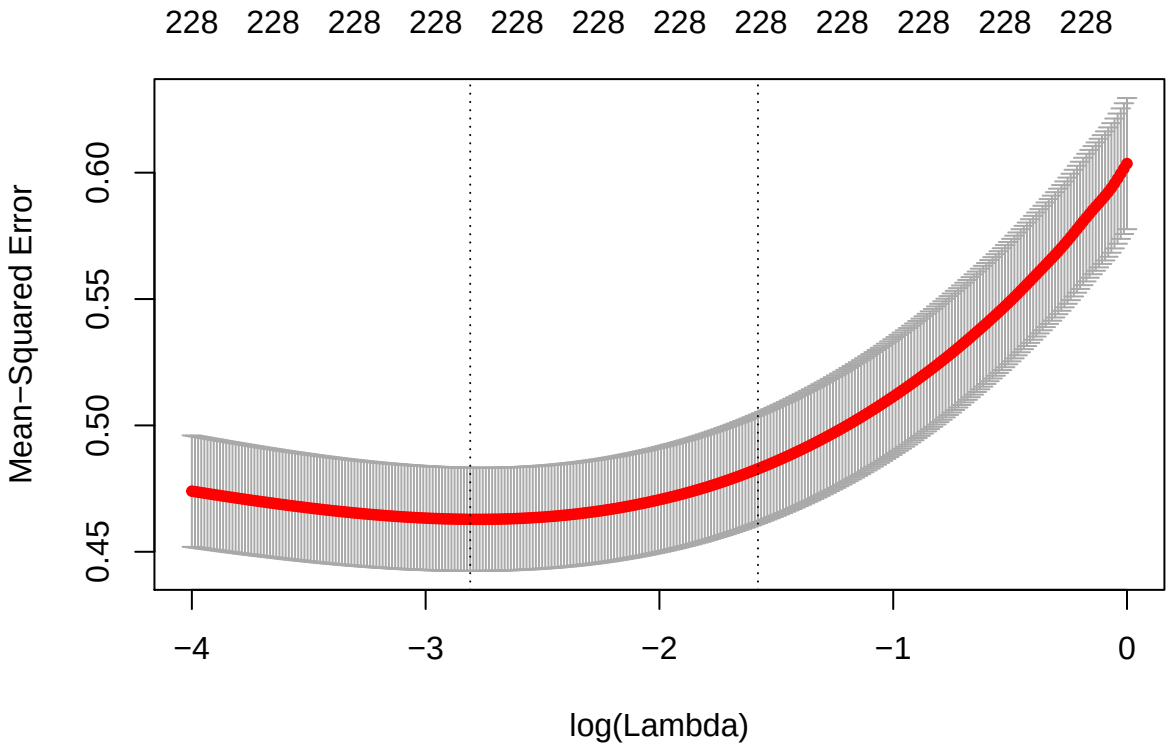
Ridge Regression

```
x = model.matrix(solubility~., train_df)[,-1]
y = train_df %>% pull(solubility)
ridge.mod <- glmnet(x, y, alpha = 0, lambda = exp(seq(-2, 0, length = 300)))
coef(ridge.mod) %>% dim()
```

```
## [1] 229 300
```

```
cv.ridge <- cv.glmnet(x, y,
                      alpha = 0,
                      nfolds = 10,
                      lambda = exp(seq(-4, 0, length = 300)),
                      type.measure = "mse")

plot(cv.ridge)
```




```
best_lambda_ridge <- cv.ridge$lambda.min
best_lambda_ridge
```

```
## [1] 0.06024326
```

The lambda corresponding to the lowest training MSE is 0.0602433.

```
new_x = model.matrix(solubility~., test_df)[,-1]
test_df_ridge =
  test_df %>%
  mutate(pred = predict(ridge.mod, s = best_lambda_ridge, newx = new_x, type = "response")) %>%
  mutate(error = pred - solubility)

mse_ridge = mean(test_df_ridge$error^2)
mse_ridge
```

```
## [1] 0.5138249
```

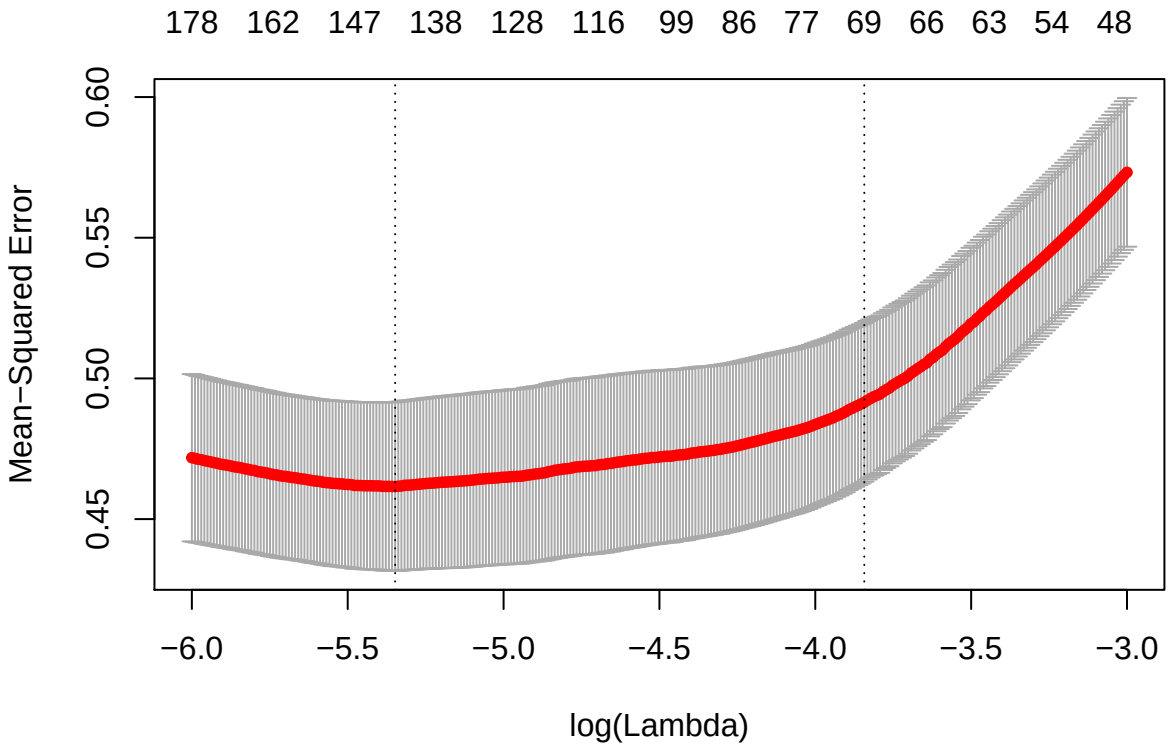
The test mean square error for the ridge model is 0.5138249.

Lasso Regression

```
x = model.matrix(solubility~., train_df)[,-1]
y = train_df %>% pull(solubility)
lasso_mod <- glmnet(x, y, alpha = 1, lambda = exp(seq(-4, 0, length = 300)))
```

```
cv_lasso <- cv.glmnet(x, y,
                      alpha = 1,
                      nfolds = 10,
                      lambda = exp(seq(-6, -3, length = 300)),
                      type.measure = "mse")

plot(cv_lasso)
```




```
best_lambda_lasso <- cv_lasso$lambda.min
best_lambda_lasso
```

```
## [1] 0.004758484
```

The lambda corresponding to the lowest training MSE is 0.0047585.

```
new_x = model.matrix(solubility~., test_df)[,-1]
test_df_lasso =
  test_df %>%
  mutate(pred = predict(lasso_mod, s = best_lambda_lasso, newx = new_x, type = "response")) %>%
  mutate(error = pred - solubility)

mse_lasso = mean(test_df_lasso$error^2)
mse_lasso
```

```
## [1] 0.5333447
```

The test mean square error for the lasso model is 0.5333447.

```
n_nonzero_coef =
  glmnet(x, y, alpha = 1, lambda = best_lambda_lasso) %>%
  coef %>%
  as.matrix() %>%
  as.tibble() %>%
  filter(s0 != 0) %>%
  nrow()

n_nonzero_coef
```

```
## [1] 143
```

There are 143 non-zero coefficient estimates if we use the 'best' lambda 0.0047585.

Principal Component Regression

```
pcr_mod <- pcr(solubility~.,
               data = train_df,
               scale = TRUE,
               validation = "CV")
# find the number of components with the lowest MSEP
class(pcr_mod)
```

```
## [1] "mvr"
```

```
msep_sort =
  pcr_mod %>%
  MSEP %>% # extract the object VALIDATION: RMSEP
  .[[1]] %>% # extract the array from the object
  .[2,,] %>% # extract the CV MESP(numeric) from the array
  as.list %>% as.tibble() %>% # coerce to tibble
  gather(key = 'ncomp', value = 'msep', `(Intercept)`: `228 comps`) %>%
  arrange(msep) # sort by MSEP to find the best M
msep_sort
```

```
## # A tibble: 229 x 2
##    ncomp      msep
##    <chr>     <dbl>
##  1 157 comps 0.490
##  2 154 comps 0.491
##  3 158 comps 0.491
##  4 153 comps 0.492
##  5 160 comps 0.492
##  6 149 comps 0.492
##  7 159 comps 0.492
##  8 155 comps 0.492
##  9 148 comps 0.493
## 10 150 comps 0.493
## # ... with 219 more rows
```

The number of M is 157 comps.

```
ncomp = msep_sort[1,] %>% pull(ncomp) %>% str_remove(' comps') %>% as.numeric()
pcr_pred = predict(pcr_mod, test_df, ncomp = ncomp)
```

```
mse_pcr = mean((pcr_pred - test_df$solubility)^2)
mse_pcr
```

```
## [1] 0.549917
```

The test MSE for the PCR model is 0.549917, with M = 157.

Discussion

The MSE's for the 4 models are summarized below.

Model	Test MSE
Least Squares	0.5558898
Ridge	0.5138249
Lasso	0.5333447
PCR	0.549917

With this data set, the Ridge regression has the lowest test MSE, and as expected, the ordinary Least squares regression has the highest test MSE. The Ridge, Lasso and PCR regression use regularization or dimension reduction techniques to decrease the variability in coefficients, so they perform better than the ordinary least squares regression.

We use cross-validation extensively throughout the homework. It is a powerful tool in selecting tuning parameters as well as measuring model predictability.