BA810 individual assignement

Qiaoling Huang (U20421641)

10/5/2019

Setup

```
install.packages("glmnet")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(ggplot2)
library(ggthemes)
library(glmnet)
```

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

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

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

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
```

```
##     accumulate, when
```

```
## Loaded glmnet 2.0-18
```

```
theme_set(theme_bw())
```

Load the CA housing Dataset

```
getwd()
```

```
## [1] "/cloud/project"
```

```
dd <- read_csv("housing.csv")
```

```
## Parsed with column specification:
## cols(
##   longitude = col_double(),
##   latitude = col_double(),
##   housing_median_age = col_double(),
##   total_rooms = col_double(),
##   total_bedrooms = col_double(),
##   population = col_double(),
##   households = col_double(),
##   median_income = col_double(),
##   median_house_value = col_double(),
##   ocean_proximity = col_character()
## )
```

```
glimpse(dd)
```

```
## Observations: 20,640
## Variables: 10
## $ longitude          <dbl> -122.23, -122.22, -122.24, -122.25, -122.25...
## $ latitude           <dbl> 37.88, 37.86, 37.85, 37.85, 37.85, 37.85, 3...
## $ housing_median_age <dbl> 41, 21, 52, 52, 52, 52, 52, 52, 42, 52, 52,...
## $ total_rooms        <dbl> 880, 7099, 1467, 1274, 1627, 919, 2535, 310...
## $ total_bedrooms     <dbl> 129, 1106, 190, 235, 280, 213, 489, 687, 66...
## $ population         <dbl> 322, 2401, 496, 558, 565, 413, 1094, 1157, ...
## $ households         <dbl> 126, 1138, 177, 219, 259, 193, 514, 647, 59...
## $ median_income      <dbl> 8.3252, 8.3014, 7.2574, 5.6431, 3.8462, 4.0...
## $ median_house_value <dbl> 452600, 358500, 352100, 341300, 342200, 269...
## $ ocean_proximity    <chr> "NEAR BAY", "NEAR BAY", "NEAR BAY", "NEAR B...
```

The total_bedrooms column has missing data that we are going to impute.

```
total_bedrooms_median <- median(dd$total_bedrooms, na.rm = TRUE)
dd <- dd %>%
replace_na(list("total_bedrooms" = total_bedrooms_median))
dd
```

```
## # A tibble: 20,640 x 10
##    longitude latitude housing_median_~ total_rooms total_bedrooms
##        <dbl>    <dbl>            <dbl>       <dbl>          <dbl>
##  1     -122.     37.9               41         880            129
##  2     -122.     37.9               21        7099           1106
##  3     -122.     37.8               52        1467            190
```

```
##  4     -122.     37.8              52        1274            235
##  5     -122.     37.8              52        1627            280
##  6     -122.     37.8              52         919            213
##  7     -122.     37.8              52        2535            489
##  8     -122.     37.8              52        3104            687
##  9     -122.     37.8              42        2555            665
## 10     -122.     37.8              52        3549            707
## # ... with 20,630 more rows, and 5 more variables: population <dbl>,
## #   households <dbl>, median_income <dbl>, median_house_value <dbl>,
## #   ocean_proximity <chr>
```

Split dataset in train and test.

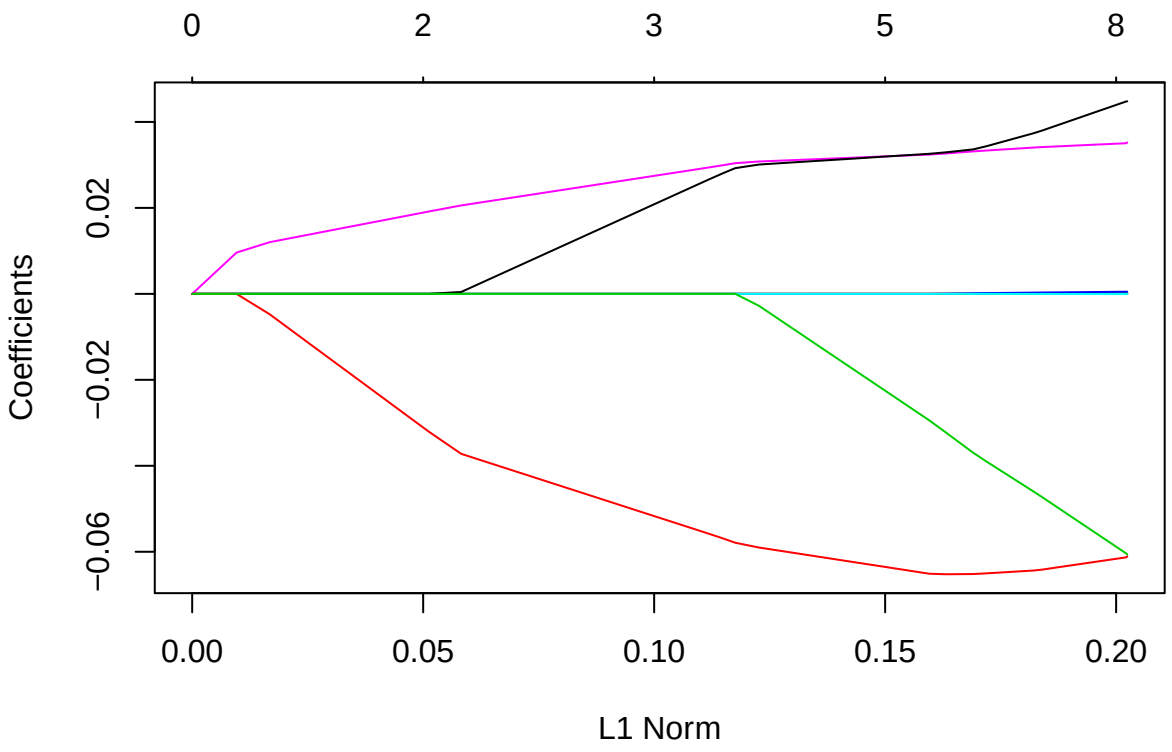
```
# use this piece of code as is
train_offsets <- seq(5000)
test_offsets <- 15000 + seq(3000)
x_data <- model.matrix( ~ -1 + total_rooms + total_bedrooms +
households + housing_median_age +
population + median_income + ocean_proximity, dd)
# outcome is median house value in millions
y_data <- dd$median_house_value / 1e6
x_train <- x_data[train_offsets, ]
y_train <- y_data[train_offsets]
x_test <- x_data[test_offsets, ]
y_test <- y_data[test_offsets]
```

Run Lasso regression

```
est <- glmnet(x_train, y_train, alpha = 1, nlambda = 100)
est$lambda
```

```
##  [1] 7.264200e-02 6.618868e-02 6.030866e-02 5.495101e-02 5.006931e-02
##  [6] 4.562129e-02 4.156842e-02 3.787560e-02 3.451083e-02 3.144499e-02
## [11] 2.865150e-02 2.610618e-02 2.378698e-02 2.167381e-02 1.974837e-02
## [16] 1.799398e-02 1.639544e-02 1.493892e-02 1.361178e-02 1.240255e-02
## [21] 1.130074e-02 1.029682e-02 9.382075e-03 8.548597e-03 7.789163e-03
## [26] 7.097195e-03 6.466700e-03 5.892216e-03 5.368768e-03 4.891821e-03
## [31] 4.457245e-03 4.061276e-03 3.700484e-03 3.371743e-03 3.072206e-03
## [36] 2.799280e-03 2.550600e-03 2.324011e-03 2.117553e-03 1.929435e-03
## [41] 1.758029e-03 1.601851e-03 1.459547e-03 1.329885e-03 1.211742e-03
## [46] 1.104094e-03 1.006009e-03 9.166380e-04 8.352064e-04 7.610090e-04
## [51] 6.934030e-04 6.318030e-04 5.756754e-04 5.245340e-04 4.779358e-04
## [56] 4.354773e-04 3.967907e-04 3.615409e-04 3.294226e-04 3.001576e-04
## [61] 2.734925e-04 2.491961e-04 2.270582e-04 2.068870e-04 1.885077e-04
## [66] 1.717612e-04 1.565024e-04 1.425992e-04 1.299311e-04 1.183884e-04
## [71] 1.078711e-04 9.828809e-05 8.955645e-05 8.160050e-05
```

```
plot(est)
```



Predict response

Next, we will use each of these 100 models to create predictions for both train and test.

```
y_train_hat<- predict(est, s = est$lambda, newx = x_train)
y_test_hat <- predict(est, s = est$lambda, newx = x_test)
```

Compute MSEs

```
mse_train=vector()
mse_test=vector()
for (i in 1:length(est$lambda)) {
  mse_train[i] <- mean((y_train - y_train_hat[,i])^2)
  mse_test[i] <- mean((y_test - y_test_hat[,i])^2)
}
mse_train
```

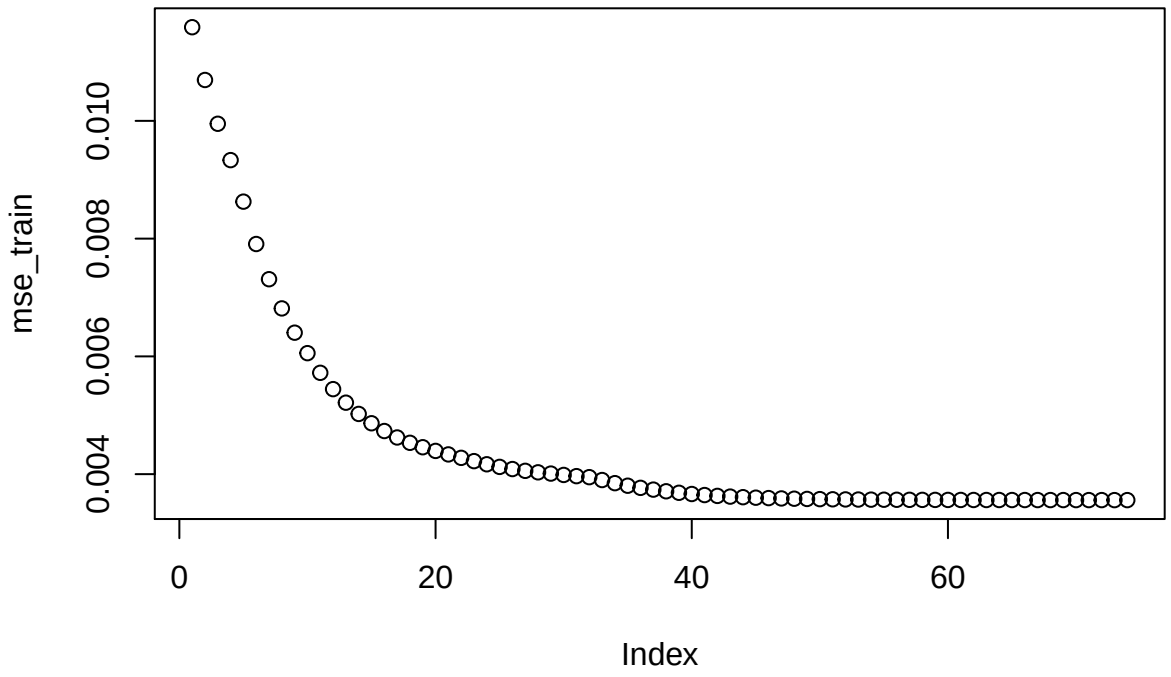
```
##  [1] 0.011590426 0.010694508 0.009950701 0.009333180 0.008628351
##  [6] 0.007908480 0.007310832 0.006814653 0.006402717 0.006054776
## [11] 0.005720891 0.005443702 0.005213575 0.005022519 0.004863902
## [16] 0.004732214 0.004622885 0.004532118 0.004456762 0.004394295
## [21] 0.004335298 0.004276605 0.004220875 0.004167545 0.004123280
## [26] 0.004086531 0.004055979 0.004030654 0.004009631 0.003986281
## [31] 0.003965869 0.003948875 0.003899676 0.003846876 0.003803384
## [36] 0.003767057 0.003737093 0.003708590 0.003683304 0.003662336
## [41] 0.003644911 0.003630450 0.003618445 0.003608477 0.003600200
## [46] 0.003593262 0.003587542 0.003582798 0.003578807 0.003575529
## [51] 0.003572811 0.003570513 0.003568634 0.003567041 0.003565744
```

```
## [56] 0.003564671 0.003563752 0.003563009 0.003562370 0.003561834
## [61] 0.003561405 0.003561035 0.003560740 0.003560483 0.003560280
## [66] 0.003560101 0.003559940 0.003559755 0.003559578 0.003559419
## [71] 0.003559279 0.003559159 0.003559057 0.003558984
```

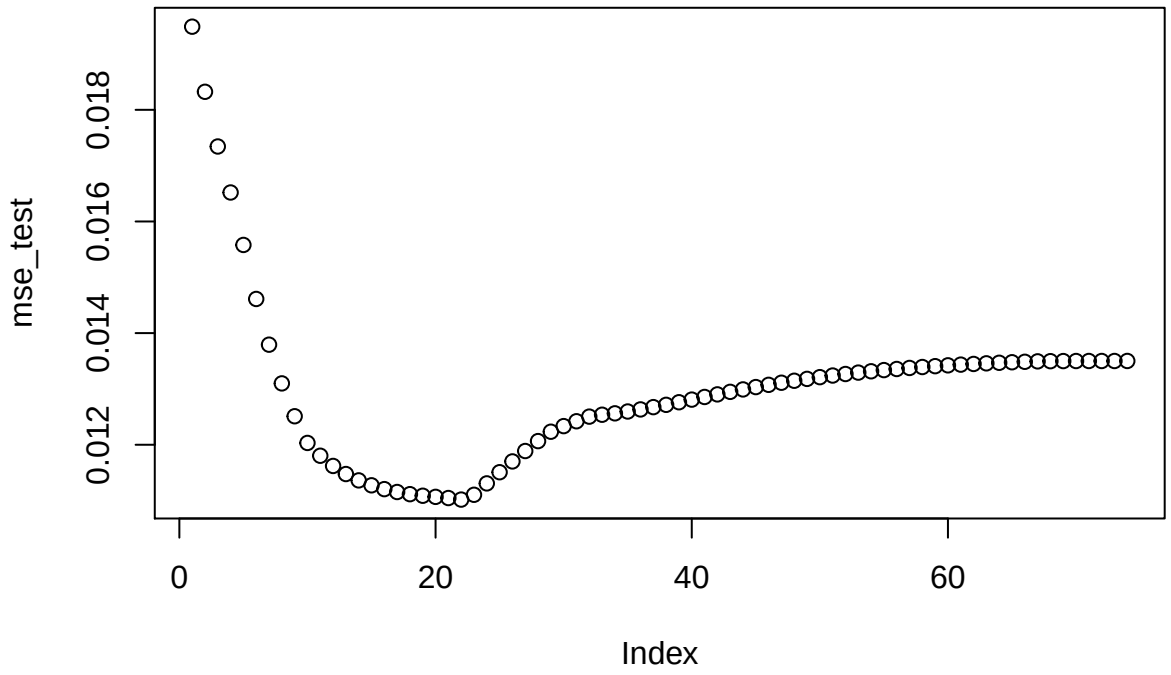
```
mse_test
```

```
##  [1] 0.01948911 0.01832295 0.01734312 0.01651902 0.01557935 0.01461175
##  [7] 0.01379305 0.01309932 0.01251059 0.01203278 0.01180332 0.01162053
## [13] 0.01147582 0.01136210 0.01127353 0.01120533 0.01115357 0.01111502
## [19] 0.01108704 0.01106681 0.01104580 0.01101823 0.01110399 0.01130825
## [25] 0.01150812 0.01170173 0.01188815 0.01206549 0.01223363 0.01233397
## [31] 0.01242118 0.01250428 0.01253797 0.01256289 0.01259566 0.01263312
## [37] 0.01267409 0.01271543 0.01276170 0.01280880 0.01285596 0.01290248
## [43] 0.01294782 0.01299156 0.01303344 0.01307339 0.01311114 0.01314667
## [49] 0.01318013 0.01321133 0.01324038 0.01326754 0.01329263 0.01331602
## [55] 0.01333751 0.01335728 0.01337568 0.01339247 0.01340808 0.01342250
## [61] 0.01343555 0.01344768 0.01345861 0.01346879 0.01347792 0.01348644
## [67] 0.01349399 0.01349737 0.01349917 0.01350006 0.01350043 0.01350052
## [73] 0.01350049 0.01350118
```

```
plot(mse_train)
```




```
plot(mse_test)
```



Choose the lowest MSE for train and test

```
lambda_min_mse_train <- est$lambda[which.min(mse_train)]
lambda_min_mse_test <- est$lambda[which.min(mse_test)]
lambda_min_mse_train
```

```
## [1] 8.16005e-05
```

```
lambda_min_mse_test
```

```
## [1] 0.01029682
```

Aggregate all MSEs in a single dataset

Create a tibble of train MSEs and lambdas

```
dd_mse <- tibble(
  lambda = est$lambda,
  mse = mse_train,
  dataset = "Train"
)

dd_mse <- rbind(dd_mse, tibble(
  lambda = est$lambda,
  mse = mse_test,
  dataset = "Test"
))

dd_mse
```

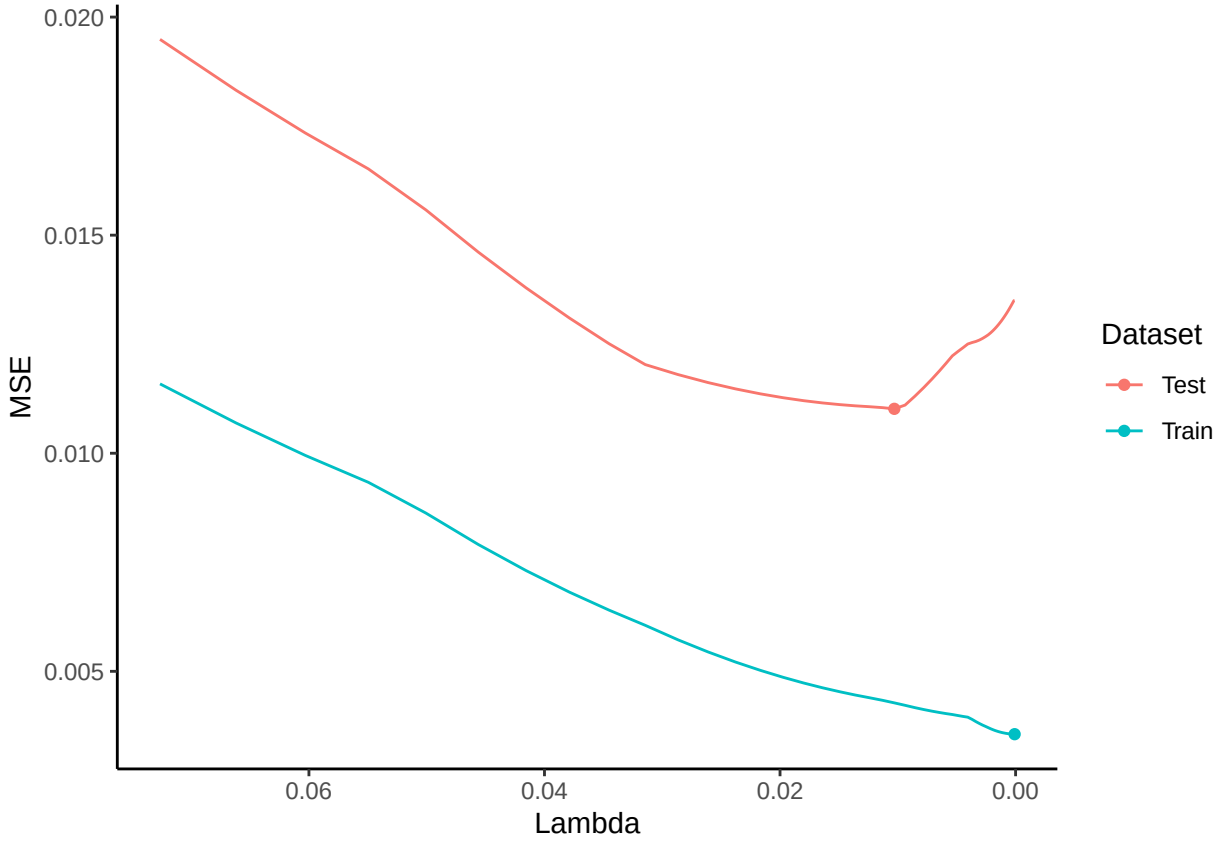
```
## # A tibble: 148 x 3
##    lambda     mse dataset
##     <dbl>   <dbl> <chr>
##  1 0.0726 0.0116  Train
```

```
##  2 0.0662 0.0107  Train
##  3 0.0603 0.00995 Train
##  4 0.0550 0.00933 Train
##  5 0.0501 0.00863 Train
##  6 0.0456 0.00791 Train
##  7 0.0416 0.00731 Train
##  8 0.0379 0.00681 Train
##  9 0.0345 0.00640 Train
## 10 0.0314 0.00605 Train
## # ... with 138 more rows
```

Plot the MSEs

```
mins <- dd_mse %>%
  group_by(dataset) %>%
  filter(mse == min(mse))

dd_mse %>%
  ggplot(aes(lambda, mse, color = dataset))+
  geom_line()+
  scale_x_reverse()+
   labs(color = "Dataset",
        x = "Lambda",
        y = "MSE")+
  geom_point(data = mins, aes(x = lambda, y = mse))+
  theme_classic()
```



Discuss the results of the best fitting model

```
print(lambda_min_mse_test)
```

```
## [1] 0.01029682
```

```
coef(est, s = lambda_min_mse_test)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                       1
## (Intercept)                8.041209e-02
## total_rooms                .
## total_bedrooms             3.268843e-06
## households                 .
## housing_median_age         .
## population                 .
## median_income              3.040308e-02
## ocean_proximity<1H OCEAN   2.925072e-02
## ocean_proximityINLAND     -5.790602e-02
## ocean_proximityISLAND      .
## ocean_proximityNEAR BAY    .
## ocean_proximityNEAR OCEAN  .
```

If I am considering investing in CA real estate, I will mainly focus on median_income, ocean_proximity<1H ocrean, and ocean_proximityINLAND as the coefficients shows these variables have significant correlation. Although total_bedrooms has correlation, the coefficient is too small. Therefore, I think total_bedrooms is not the variable that I should mainly focus on.

Collabration statement

I completed the majority of the assignment myself by reading the text book from page 251 to page 255. Meanwhile, Zhang Hang from cohort A taught me to use for loop to write the code to create a vector contains 100 MSE. And Shangkun Zuo from cohort B reminded me to fix the correct code for lambda_min_mse_test.