

# Assignment1

Farid Musayev, Kristina Levina, and Wuhao Wang

02 December, 2021

## Question 1

The results below shows the error in training and test data.

```
## [1] "using mean square error"

## [1] "error for train: 0.00570911701090834"

## [1] "error for test: 772.498785033593"
```

The model is good in training data, but not very good in test data. So, in general, this model is not good.

## Question 2

The original cost function should be

$$\min_{(\beta_0, \beta) \in R_{p+1}} \left[ \frac{1}{N} \sum_{i=1}^N w_i l(y_i, \beta_0 + x^T \beta)^2 + \lambda P_\alpha(\beta) \right]$$

where  $N$  is the number of observations,  $w_i$  is the weight of each observation,  $l(p, q)$  is the minus log-likelihood function,  $\lambda$  is penalty factor,  $\beta_0$  and  $\beta$  are the coefficient of liner model ( $\beta_0$  is intercept) and  $P_\alpha(\beta)$  is:

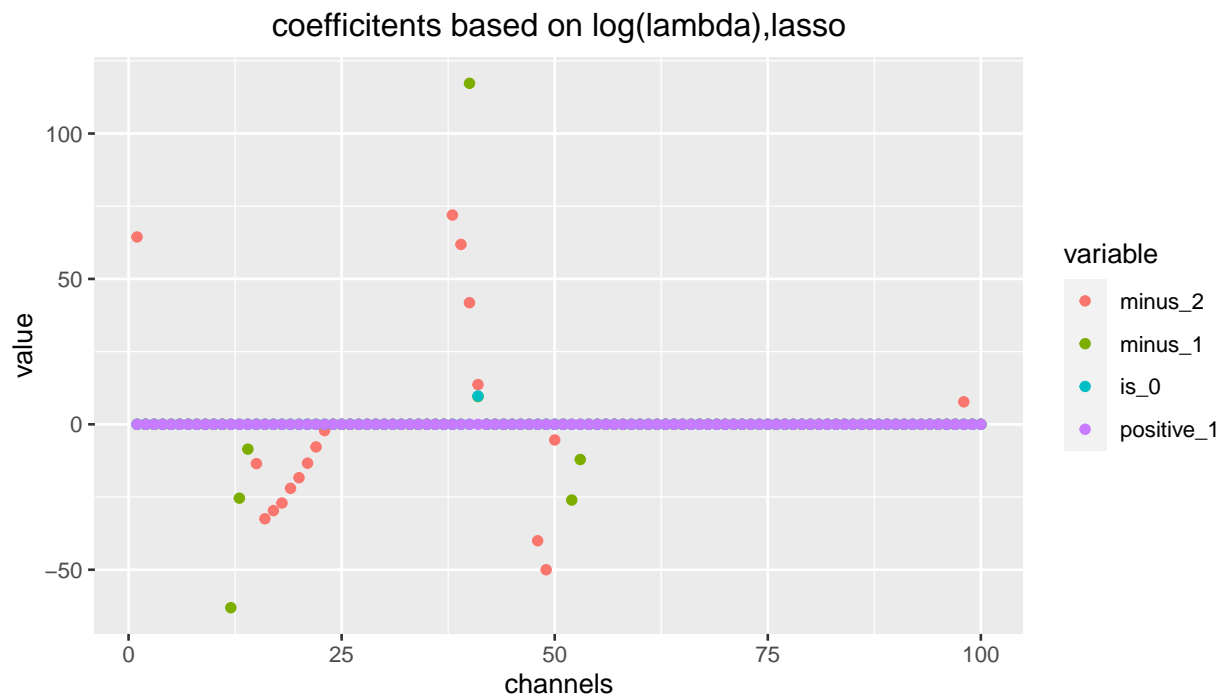
$$(1 - \alpha) \frac{\|\beta\|_{l_2}^2}{2} + \alpha \|\beta\|_{l_1}$$

Noticing that  $\alpha$  is the control parameter, if  $\alpha = 1$ , then it is LASSO penalty. Here we are going to implement LASSO regression model, so we set  $\alpha = 1$ . Then the cost function for LASSO regression is:

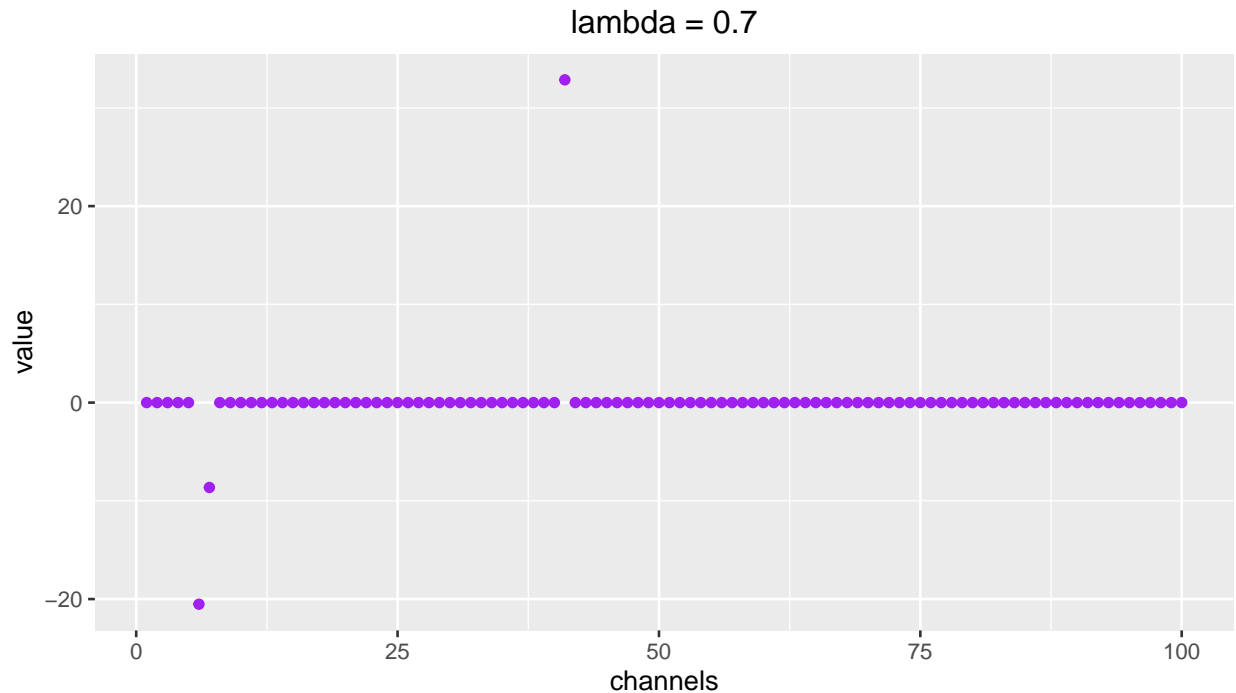
$$\min_{(\beta_0, \beta) \in R_{p+1}} \left[ \frac{1}{N} \sum_{i=1}^N w_i l(y_i, \beta_0 + x^T \beta)^2 + \lambda \alpha \|\beta\|_1 \right]$$

### Question 3

We implement 4 lasso regression models, and pick lambda from (0.01, 0.1, 1, 10) to see how coefficients change.



From the plot, we can see with  $\log(\lambda)$  increases, the number of non-zero coefficients decreases rapidly ( points with color red green and blue are covered by purple points in  $y=0$ ). So if we only want three coefficients, we will get a interval for  $\lambda$  , and can use  $\lambda = 0.7$ .



## Question 4

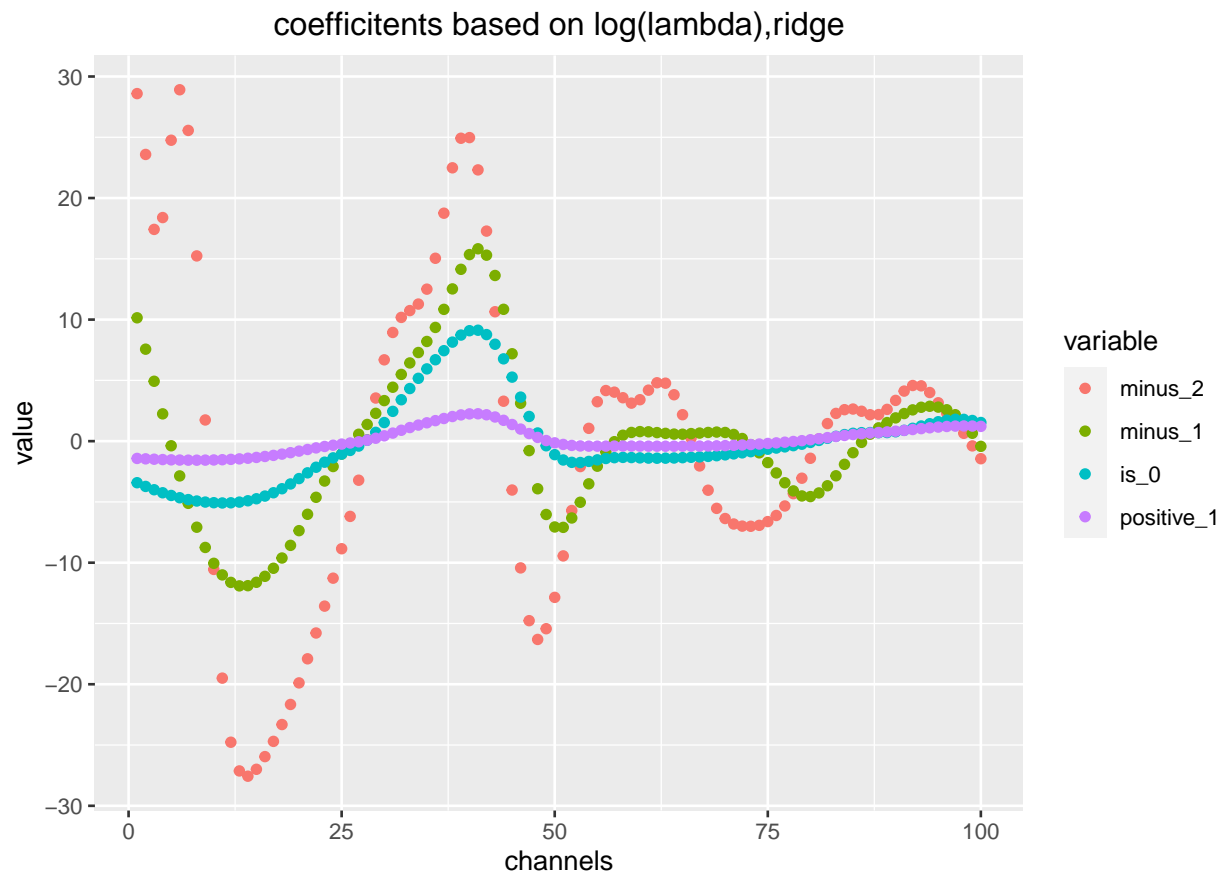
In this question, we set  $\alpha = 0$  to use ridge regression model.

Repeating step 3, we find that in ridge regression, all the features are used, although some of them seems highly correlated, which means we can set those coefficients into pairs, and the sum of each pair are closed to zero (we can see the plot below and find that the every curve seems roughly has mean value 0).

Besides, even we run penalty factor from 100 to 200, we can still not find lambda which can make model have only 3 features, so we guess we can not find that lambda.

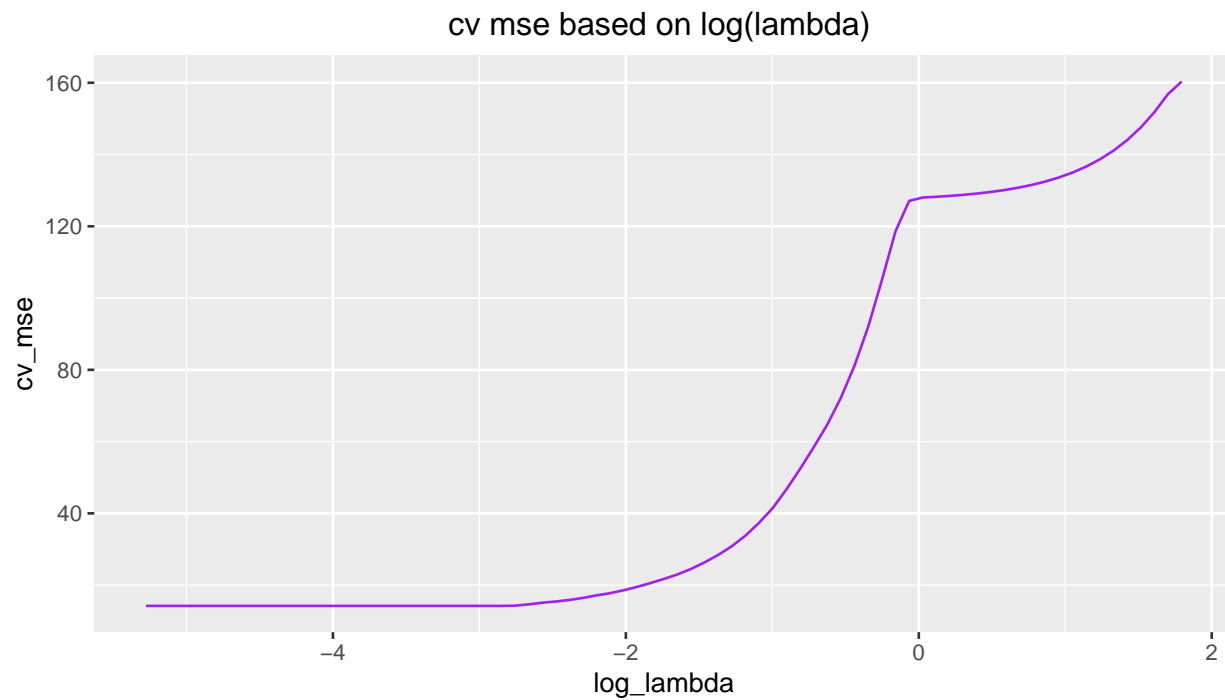
Conclusion:

LASSO model should be applied when features are highly correlated, since it will set corresponding coefficients to 0, thus, simplifying the model.



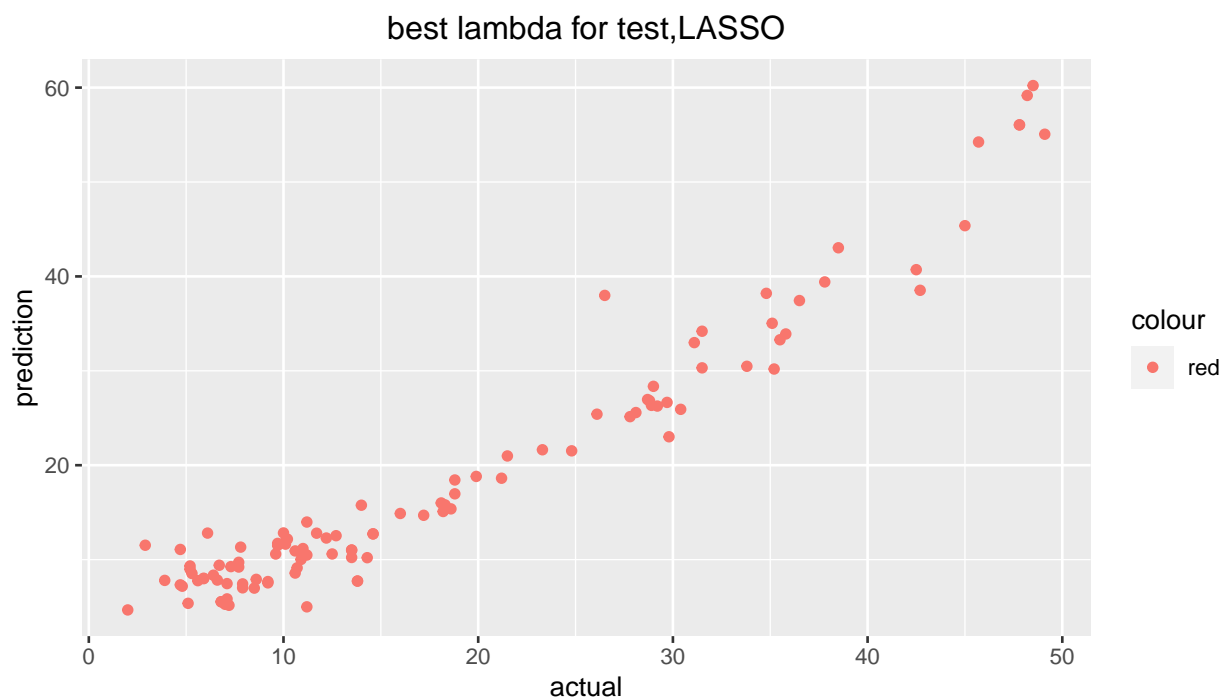
## Question 5

```
## [1] "the best lambda is 0.0574453477182168 log(lambda) = -2.85692125763191"
```



```
## [1] "the number of features: 7"
```

The cv score keeps stable first, and then increases rapidly when  $\log(\lambda)$  between  $(-3,0)$ . After that, the curve shows similar trend again. According to the plot, the  $\lambda_{opt}$  shows no statistical difference with  $\log\lambda=-4$ .



```
## [1] "mse : 13.3135363399641"
```

According to the plot and mse, we can see there is a huge progress compared with simple liner regression. In general, this model prediction is good.

## APPENDIX

*hints: (some codes related to print are set '## expression')*

```
library(ggplot2)
library(reshape2)
library(glmnet)

data <- read.csv('tecator.csv')
data <- data[,-1]
data <- data[,c(-102,-103)]
#####
set.seed(12345)
n <- dim(data)[1]
id <- sample(1:n, floor(n*0.5))
train <- data[id,]
id1 <- setdiff(1:n, id)
test <- data[id1,]
rm(id,id1)
#####
#Question 1
#####
lm_q1 <- lm(Fat~.,data = train)
pre_train <- predict.lm(lm_q1,train)
pre_test <- predict.lm(lm_q1,test)
error_train <- sum((pre_train-train[, 'Fat'])**2)/length(pre_train)
error_test <- sum((pre_test-test[, 'Fat'])**2)/length(test)
##print('using mean square error')
##print(paste0('error for train: ',error_train))
##print(paste0('error for test: ',error_test))
#####
#Question 2
#####
#in the Rmarkdown
#in the Rmarkdown
#in the Rmarkdown
#in the Rmarkdown
#in the Rmarkdown
#in the Rmarkdown
#####
#Question 3
#####
gl_0.01 <- glmnet(train[, -101],train[, 101],lambda = 0.01)
gl_0.1 <- glmnet(train[, -101],train[, 101],lambda = 0.1)
gl_1 <- glmnet(train[, -101],train[, 101],lambda = 1)
gl_10 <- glmnet(train[, -101],train[, 101],lambda = 10)
df_lambda <-data.frame(channels =1:100,
                        minus_2 = gl_0.01[['beta']][,1],
                        minus_1 = gl_0.1[['beta']][,1],
                        is_0 = gl_1[['beta']][,1],
```

```

        positive_1 = gl_10[['beta']][,1]
    )
df_melt <- reshape2::melt(df_lambda,id.vars='channels')
p1 <-ggplot(df_melt,aes(x=channels,y=value))+
  geom_point(aes(color=variable))+
  ggtitle('coefficitensts based on log(lambda),lasso')+
  theme(plot.title = ggplot2::element_text(hjust=0.5))
## print(p1)
# find lambda which can let model have 3 coefficients being non-zero.
lambda_for_three_coefficients <- 0
lambda_list <- 1:10/10
for (i in lambda_list)
{
  gl <- glmnet(train[,-101],train[,101],lambda = i)
  if(length(which(gl[['beta']][,1]!=0))==3)
  {
    lambda_for_three_coefficients <- i
    df_lambda <-data.frame(channels =1:100,
                          value = gl[['beta']][,1])
    p1 <-ggplot(df_lambda,aes(x=channels,y=value))+
      geom_point(color='purple')+
      ggtitle(paste0('lambda = ', i))+
      theme(plot.title = ggplot2::element_text(hjust=0.5))
    ## print(p1)
    break
  }
}
#####
#Question 4
#####
rg_0.01 <- glmnet(train[,-101],train[,101],lambda = 0.01,alpha = 0)
rg_0.1 <- glmnet(train[,-101],train[,101],lambda = 0.1,alpha = 0)
rg_1 <- glmnet(train[,-101],train[,101],lambda = 1,alpha = 0)
rg_10 <- glmnet(train[,-101],train[,101],lambda = 10,alpha = 0)
df_lambda <-data.frame(channels =1:100,
                      minus_2 = rg_0.01[['beta']][,1],
                      minus_1 = rg_0.1[['beta']][,1],
                      is_0 = rg_1[['beta']][,1],
                      positive_1 = rg_10[['beta']][,1]
)
df_melt <- reshape2::melt(df_lambda,id.vars='channels')
p1 <-ggplot(df_melt,aes(x=channels,y=value))+
  geom_point(aes(color=variable))+
  ggtitle('coefficitensts based on log(lambda),ridge')+
  theme(plot.title = ggplot2::element_text(hjust=0.5))
## print(p1)
# find lambda which can let model have 3 coefficients being non-zero.
lambda_for_three_coefficients <- 0
lambda_list <- 100:200
for (i in lambda_list)
{
  rg <- glmnet(train[,-101],train[,101],lambda = i,alpha = 0)
  if(length(which(rg[['beta']][,1]!=0))==3)

```

```

{
  lambda_for_three_coefficients <- i
  df_lambda <- data.frame(channels = 1:100,
                          value = rg[['beta']][,1])
  p1 <- ggplot(df_lambda, aes(x=channels, y=value)) +
    geom_point(color='purple') +
    ggtitle(paste0('lambda = ', i)) +
    theme(plot.title = ggplot2::element_text(hjust=0.5))
  ## print(p1)
  break
}
}

#####
#Question 5
#####
cv <- cv.glmnet(as.matrix(train[,-101]), as.matrix(train[,101]), type.measure = 'mse')
best_lambda <- cv$lambda.min
##print(paste0('the best lambda is ', best_lambda, ' log(lambda) = ', log(best_lambda)))
df <- data.frame(log_lambda = log(cv$lambda), cv_mse = cv$cvm)
p1 <- ggplot(data = df, aes(x = log_lambda, y = cv_mse)) +
  geom_line(color='purple') +
  ggtitle('cv mse based on log(lambda) ') +
  theme(plot.title = ggplot2::element_text(hjust=0.5))
##print(p1)
gl <- glmnet(train[,-101], train[,101], lambda = best_lambda)

pre_test <- predict(gl, as.matrix(test[,-101]))[,1]

actual_test <- as.numeric(as.character(test[,101]))

df_test <- data.frame(
  actual = actual_test,
  prediction = pre_test
)
p2 <- ggplot(df_test, aes(x=actual, y=prediction)) +
  geom_point(aes(color='red')) +
  ggtitle('best lambda for test, LASSO') +
  theme(plot.title = ggplot2::element_text(hjust=0.5))
##print(p2)

##print(paste0('mse : ', mean((pre_test-actual_test)**2)))
##print(paste0('the mse : ', mse))

```