Machine Leaning Block 1 Lab #2

Group A13: Arash Haratian, Connor Turner, Yi Hung Chen

2022-11-27

Statement of Contribution: *Assignment 1 was contributed by Yi Hung, Assignment 2 was contributed by Arash, and Assignment 3 was contributed by Connor. Each assignment was then discussed in detail with the group before piecing together each secton of the final report.*

Assignment 1. Explicit regularization

For this assignment, we are given data "tecator.csv" that contains the results of study aimed to investigate whether a near infrared absorbance spectrum can be used to predict the fat content of samples of meat.

Q1. We begin by divide data randomly to train and test set(50/50). We then use the training data to train a linear regression model with all the absorbance characteristics (Channels) as features. Below is the underlying probabilistic model of linear regression

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_{100} x_{100} + \epsilon$$
$$y|x \sim N(\theta^T x, \sigma^2)$$

where

$$y = Fat$$
$$\theta = \theta_0, ..., \theta_{100}$$
$$x = \{1, Channel_1, Channel_2, ..., Channel_{100}\}$$
$$\epsilon \ is \ the \ error \ term$$

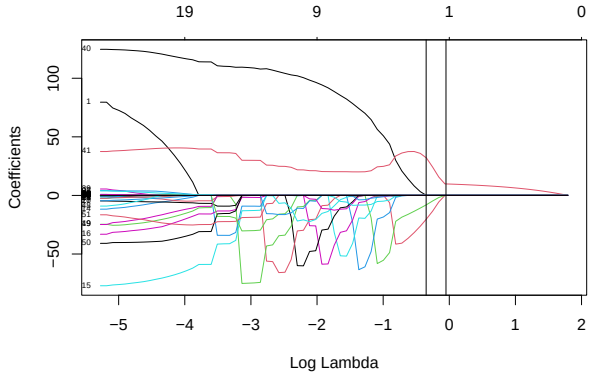
We estimate mean squared errors(MSE) for both training data and testing data. We obtain 0.0057091 for training data and 722.4294193 for testing data. The large MSE on testing data indicates the linear regression model is overfitting. To improve this, we are going to use Lasso and Ridge regression in this assignment.

Q2. We then switch to use LASSO regression model. Which the cost function that should be optimized is.

$$\hat{\theta^{lasso}} = \{\frac{1}{n}\sum_{i=1}^{n}(y_i - \theta_0 - \sum_{j=1}^{p}\theta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}|\theta_j|\}$$

Q3. We fit the LASSO regression model to the training data. The dependency of the regression coefficients on the $\log(\lambda)$ is shown below in Figure 1. As the plot shown, the LASSO will drop some coefficients as λ increase (set the values to 0). Interestingly, when λ change, some coefficient will reappear.

Figure 1:
Dependence of Lasso Regression coefficients on log(lambda)

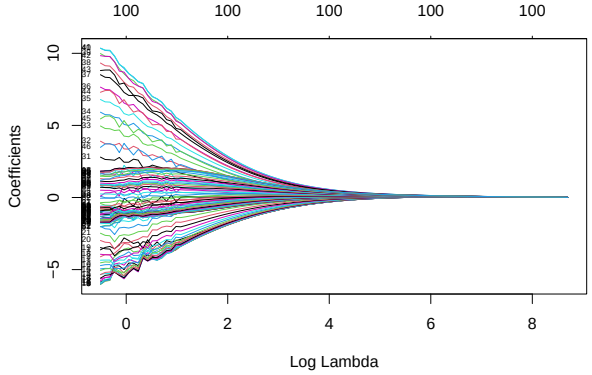


If we want to select a model with only three features, we can choose λ around 0.9512294 to 0.7046881.(Note: The range is choosen by dierectly observed with the plot(vertical lines). Additionally, by using default lambdas in glmnet, we calculated $\lambda = 0.8530452, 0.777263, 0.7082131$ will gives coefficient for only three features. However, as the plot shown, the lambdas should be in a range not district values.)

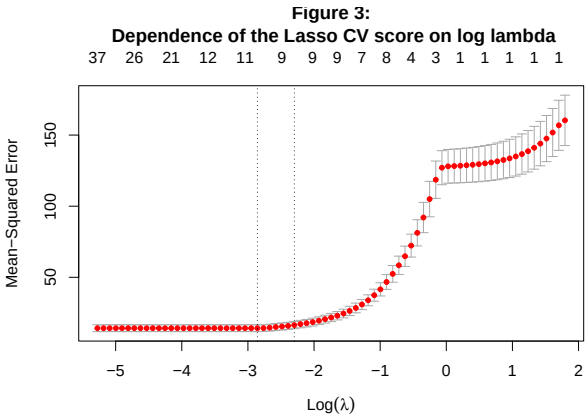
Q4. We then repeat the privious step, but instead of using Lasso Regression, we use Ridge Regression to compare both method.

As the Figure 2 shown. Compare to Lasso Regression, the penalty term of Ridge Regression will not "collapse" to zero. Also, the coefficient value of Ridge Regression scale slower when lambda increase compared to Lasso Regression.

Figure 2:
Dependence of Ridge Regression coefficients on log(lambda)



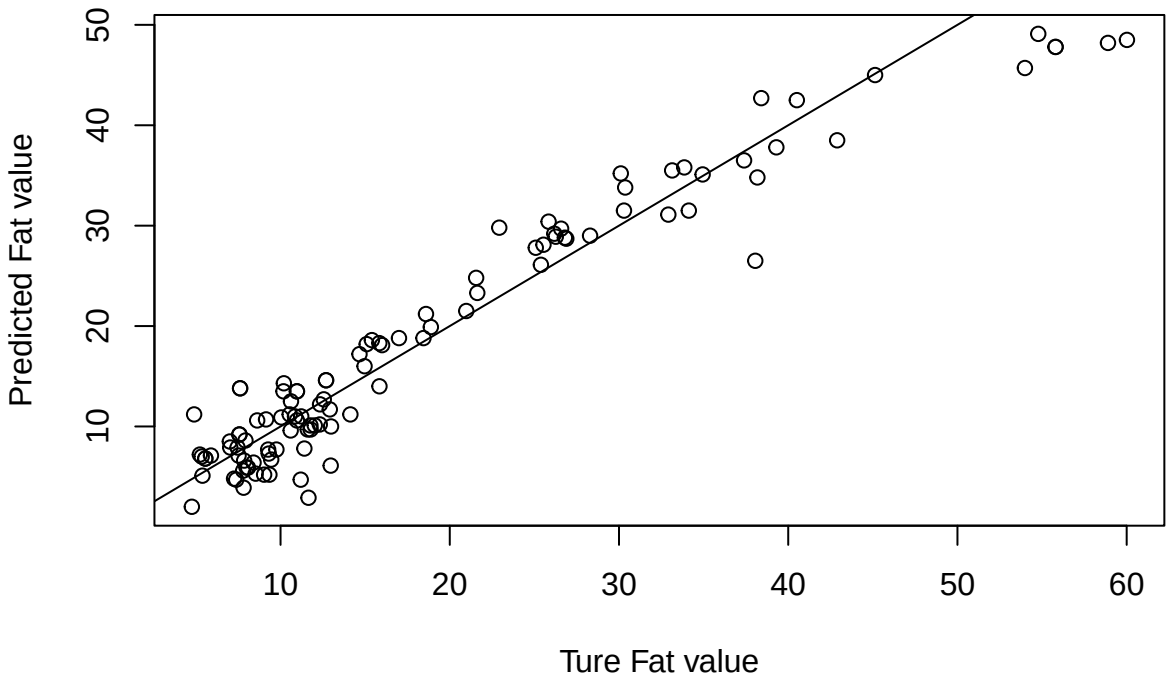
Q5. We use cv.glmnet function to do cross-validation for Lasso model. We do not specify how many folds being use, instead we use the default number of folds to compute. As the Figure 3 shown, the MSE start increasing dramatically as $\log(\lambda)$ approach -2, which means the the model become less usable, this might result from too many coefficient being ditched by Lasso regression as lambda increase.



We obtain the best λ =0.0574453 with 7 variables chosen(Channel13, Channel14, Channel15, Channel16, Channel40, Channel41, Channel51). According to Figure 3, the optimal λ ($\log(\lambda)$ =-2.8569213) is not statistically significantly better than $\log\lambda$ = -4, as they have similar MSE.

Finally, we plot the scatter plot (Figure 4) with true "Fat" value from test data on the X-axis and predict value on the Y-axis. We can observed that the prediction is much better compare to linear regression(MSE = 13.2997953). Note: We also include a line x=y which indicate where the perfect prediction should lay on.

Figure 4:Prediction of using Optimal Lasso on True test data



Appendix

```
#===== Assignment 1=====

  #==Q1==
  rm(list=ls())
  library(glmnet)
  # Split and Prepare Data
  # ------------------------------------------------
  # Split the Data Into Training and Testing Data
  # (50/50):
  tecator <- read.csv("tecator.csv")
  set.seed(12345)
  n = nrow(tecator)
  id = sample(1:n, floor(n * 0.5))
  train = tecator[id, ]
  test = tecator[-id, ]

  #=====Fit Linear Regression=====
  lm_tecator <- lm(formula=Fat~.-Protein-Moisture-Sample,data=train)

  train_fitvalues <- lm_tecator$fitted.values
  test_predict <- predict(lm_tecator, test)


  MSE_train <- mean(lm_tecator$residuals^2)
  MSE_test <- mean((test$Fat-test_predict)^2)


  #==Q2,Q3==
  #=====Lasso=====
  #Make argument in the format that glmnet can use (data matrix for x)
  x_name <- colnames(tecator)[-1]
  x_name <- x_name[-102]
  x_name <- x_name[-102]
  x_name <- x_name[-101]
  x <- data.matrix(train[, x_name])
  y <- train$Fat

  #Train Lasso
  lasso_tecator <- glmnet(x=x,y=y, alpha = 1)

  plot(lasso_tecator,xvar="lambda",label = TRUE,
    main = "Figure 1:\n\ Dependence of Lasso Regression coefficients on log(lambda) \n\n")+
    abline(v = -0.05)+
    abline(v = -0.35)

  #pick lambda for 3 features
  coef_matrix <- as.matrix(coef(lasso_tecator))
  coef_matrix <- coef_matrix!=0
  coef_matrix <- colSums(coef_matrix)
  lamda_index <- which(coef_matrix == 4) # we use 4 because there is alway intercept with in the matric
  lambda_3features <- lasso_tecator$lambda[lamda_index] # the lamda for 3 features
```

```
  #==Q4==
  #=====Ridge======
  ridge_tecator <- glmnet(x, y, alpha = 0)
  plot(ridge_tecator,xvar="lambda",label = TRUE,
      main = "Figure 2:\n\ Dependence of Ridge Regression coefficients on log(lambda) \n\n")

  #==Q5==
  cv_lasso_tecator <- cv.glmnet(x=x,y=y, alpha = 1) #Use cv.glmnet to do cross validation

  plot(cv_lasso_tecator,
     main = "Figure 3:\n Dependence of the Lasso CV score on log lambda\n\n")

  best_lambda <- cv_lasso_tecator$lambda.min
  best_model <- glmnet(x, y, alpha = 1, lambda = best_lambda)
  variables <- coef(best_model)
  variables <- as.vector(variables!= 0)
  coef_index <- which(match(variables,TRUE) == 1)
  coef_index <- coef_index[-1]

  coef_get_select <- rownames(coef(best_model))[coef_index]

#===== END Assignment 1=====
```