Solutions to Assignment

Rongfei Jin

February 28, 2025

1 Chapter 5 - Conceptual 3

(a) K-fold CV is implemented the following way:

   (a) Split the data into K folds.

   (b) For each fold i:

      i. Use the other $K-1$ folds as training data.

      ii. Train the model on the training data.

      iii. Evaluate the model on the i-th fold.

   (c) Average the evaluation results over all K folds.

(b) Compare to the validation set approach, K-fold CV ensures that all data points are used for both training and validation, which can lead to a more reliable estimate of the model's performance. It also reduces the variance of the performance estimate by averaging over multiple folds.

However, K-fold CV can be computationally more expensive, especially for large datasets or complex models, as it requires training the model K times.

(c) LOOCV is a special case of K-fold CV where K is equal to the number of data points. The K-fold CV where k is less than N is less computationally expensive, as it only requires training the model K times instead of N times. More over, since almost all points are used for training, the variance of error will be greater than that of the K-fold CV, making it more prone to overfitting. However, if a dataset is small, LOOCV can provide a more accurate estimate of the model's performance, as it uses all but one data point for training.

2 Chapter 5 - Conceptual 4

(a) Repeated sample the original with replacement to create B dataset

(b) Train the model on each dataset

(c) Predict the Y with the particular X value

(d) Compute the standard deviation of the predictions

3 Chapter 5 - Applied 5

```r
library(ISLR2)

logi_model <- glm(default ~ income + balance, data = Default, family <- binomial)

summary(logi_model)

# Validation Set Approach
validation_error <- function(seed_num) {
  set.seed(seed_num)
  train <- sample(1:nrow(Default), nrow(Default)/2)
  test <- -train
  train_data <- Default[train, ]
  test_data <- Default[test, ]

  logi_model_val <- glm(default ~ income + balance, data = train_data, family = binomial)
  pred_val <- predict(logi_model_val, newdata = test_data, type = "response")

  # convert the post prob to factor Yes and No
  pred_val <- ifelse(pred_val > 0.5, "Yes", "No")

  # compute and return the validation set error rate
  return(mean(pred_val != test_data$default))
}

validation_error(1)
validation_error(2)
validation_error(3)
# 0.0264

validation_error_2 <- function(seed_num) {
  set.seed(seed_num)
  train <- sample(1:nrow(Default), nrow(Default)/2)
  test <- -train
  train_data <- Default[train, ]
  test_data <- Default[test, ]

  logi_model_val <- glm(default ~ income + balance + student, data = train_data, family = binomial)
  pred_val <- predict(logi_model_val, newdata = test_data, type = "response")

  # convert the post prob to factor Yes and No
  pred_val <- ifelse(pred_val > 0.5, "Yes", "No")

  # compute and return the validation set error rate
  return(mean(pred_val != test_data$default))
}

validation_error_2(1)
validation_error_2(2)
validation_error_2(3)
# 0.0246
```

The difference is small so no needs to add it

4 Chapter 5 - Applied 6

```r
library(ISLR2)
library(boot)

# set the seed
logi_model <- glm(default ~ income + balance, data = Default, family <- binomial)

summary(logi_model)

# Coefficients:
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
# income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
# balance      5.647e-03  2.274e-04  24.836  < 2e-16 ***

boot.fn <- function(data, index) {
  d <- data[index, ]
  logi_model <- glm(default ~ income + balance, data = d, family = binomial)
  return(summary(logi_model)$coef[2:3, 2])
}

boot_results <- boot(Default, boot.fn, 1000, parallel = "multicore", ncpus = 16)

boot_results
# Bootstrap Statistics :
#         original       bias     std. error
# t1* 4.985167e-06 1.503909e-08 1.409532e-07
# t2* 2.273731e-04 1.204649e-06 1.135513e-05

# the income std error from the bootstrap is higher than the original model
# where the balance std error is lower, such change could suggest an underestimation of income std error
# and an overestimation of balance std error
```

5 Chapter 5 - Applied 7

```r
library(ISLR2)
library(boot)

logi_model_1 <- glm(Direction ~ Lag1 + Lag2, data = Weekly, family <- binomial)
summary(logi_model_1)


# All but first observation
logi_model_2 <- glm(Direction ~ Lag1 + Lag2, data = Weekly[2:nrow(Weekly),], family <- binomial)

first_obs_pred <- predict(logi_model_2, Weekly[1,], type = "response")

first_obs_pred <- ifelse(first_obs_pred > 0.5, "Up", "Down")
first_obs_pred == Weekly[1,]$Direction
# NO

err <- c()
for (n in 1:(nrow(Weekly) - 1)) {
  logi_model <- glm(Direction ~ Lag1 + Lag2, data = Weekly[-n,], family <- binomial)
  pred <- predict(logi_model, Weekly[n,], type = "response")
  pred <- ifelse(pred > 0.5, "Up", "Down")
  if (pred == Weekly[n,]$Direction) {
    err <- c(err, 0)
  } else {
    err <- c(err, 1)
  }
}

mean(err)
# 0.4504
# Slightly lower than 50% error rate
```

6 Chapter 6 - Conceptual 3

(a) IV training set does not have bias-variance tradeoff, so it will decrease until reach the OLS solution.

(b) II test set has bias-variance tradeoff, so after the constraint is loose enough, the model becomes too flexible and overfits the test set.

(c) III variance increases with flexibility

(d) IV bias decreases with flexibility

(e) V irreducible error is the error that cannot be reduced by any model, regardless of its flexibility.

7 Chapter 6 - Conceptual 4

(a) I, increase as the model becomes too inflexible

(b) II, decrease as the shrinkage help with overfits but after a point, it underfits

(c) IV Variance decrease as the model becomes too inflexible

(d) III Bias Increase as the model overfits

(e) Constant for the same reason in conceptual 3

8 Chapter 6 - Conceptual 5

(a) The optimization problem is

$$\arg\min_{\beta_1,\beta_2} \sum_{i=1}^{n} \left[y_i^2 - (\beta_1 x_{i1} + \beta_2 x_{i2})\right]^2 + \lambda(\beta_1^2 + \beta_2^2) \tag{1}$$

(b) Since $x_{i1} = x_{i2}$ $i = 1, 2$ We expand (1) to get

$$\sum_{i}^{2} \left[y_i^2 - 2(\beta_1 + \beta_2)x_{i1}y_i + (\beta_1 + \beta_2)^2 x_{i1}^2\right] + \lambda(\beta_1^2 + \beta_2^2)$$

Then we find the partial derivative of (2) with respect to β_1 and β_2 and set them to 0 to get the solution

$$\frac{\partial}{\partial \beta_1} = -2\sum y_i x_{i1} + 2\beta_1 \sum x_{i1}^2 + 2\beta_2 \sum x_{i1}^2 + 2\lambda\beta_1 = 0$$

$$\beta_1 \sum x_{i1}^2 + \beta_2 \sum x_{i1}^2 + \lambda\beta_1 = \sum y_i x_{i1}$$

$$\beta_1 = \frac{\sum y_i x_{i1} - \beta_2 \sum x_{i1}^2}{\sum x_{i1}^2 + \lambda}$$

Similarly, we can get

$$\beta_2 = \frac{\sum y_i x_{i1} - \beta_1 \sum x_{i1}^2}{\sum x_{i1}^2 + \lambda}$$

We denote $a = \sum y_i x_{i1}$, $b = \sum x_{i1}^2$, and $c = \lambda$ to get the solution

$$\hat{\beta}_1 = \frac{a - \beta_2 b}{b + c}$$
$$\hat{\beta}_2 = \frac{a - \beta_1 b}{b + c}$$

We substitute β_2 into the first equation to get

$$\hat{\beta}_1 = \frac{ac}{(b + c)^2 - b^2}$$

Similarly, we can get

$$\hat{\beta}_2 = \frac{ac}{(b + c)^2 - b^2}$$

Therefore $\hat{\beta}_1 = \hat{\beta}_2$

(c) The optimization problem is

$$\arg\min_{\beta_1,\beta_2} \sum_{i=1}^{n} \left[y_i^2 - (\beta_1 x_{i1} + \beta_2 x_{i2})\right]^2 + \lambda(|\beta_1| + |\beta_2|) \tag{2}$$

(d) Since $x_{i1} = x_{i2}$ $i = 1, 2$ We expand (1) to get

$$\sum_{i}^{2} \left[y_i^2 - 2(\beta_1 + \beta_2)x_{i1}y_i + (\beta_1 + \beta_2)^2 x_{i1}^2\right] + \lambda(|\beta_1| + |\beta_2|)$$

Then we find the partial derivative of (2) with respect to β_1 and β_2 and set them to 0 to get the solution

$$\hat{\beta}_1 = \frac{a - \frac{\lambda}{2} sign(\beta_2)}{a} - \beta_2$$

Similarly, we can get

$$\hat{\beta}_2 = \frac{a - \frac{\lambda}{2} sign(\beta_1)}{a} - \beta_1$$

where $a = \sum y_i x_{i1}$

Then we substitute β_2 into the first equation to get

$$\hat{\beta}_1 = \frac{a - \frac{\lambda}{2} sign(\beta_2)}{a} - \frac{a - \frac{\lambda}{2} sign(\beta_1)}{a} + \beta_1$$

We obtain $sign(\beta_1) = sign(\beta_2)$ This means that we have infinitely many solutions

9 Chapter 6 - Applied 6

When $p = 1$, 6.12 can be written as

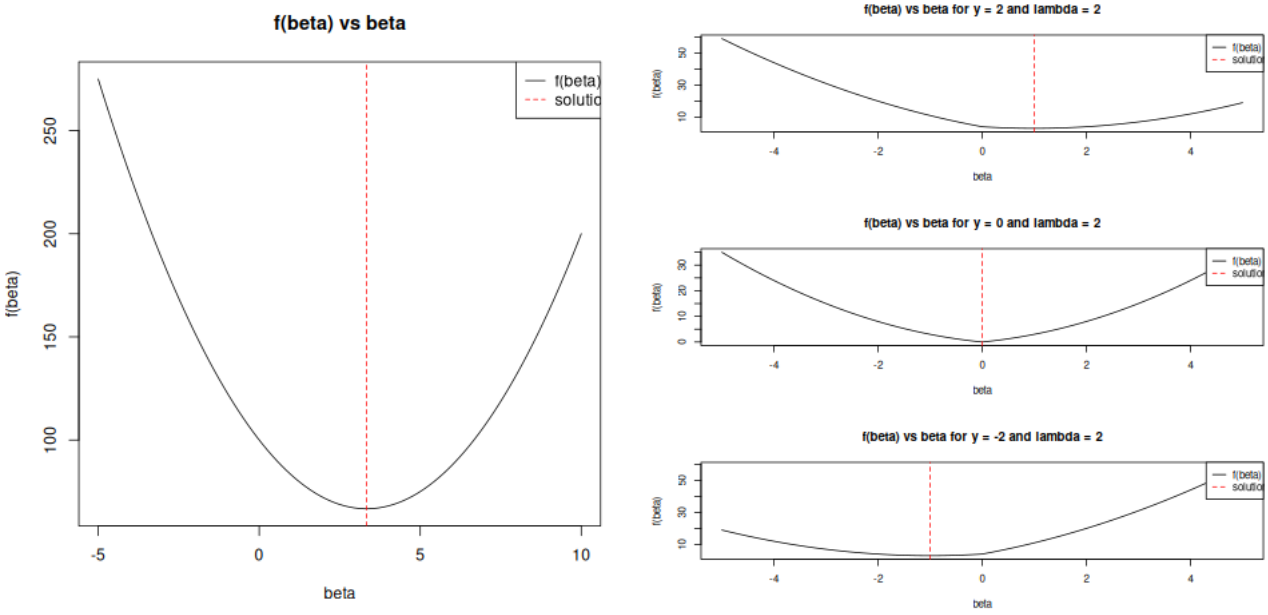
$$(y_1 - \beta_1)^2 + \lambda\beta_1^2$$

Let $y_1 = 10, \lambda = 2$, the plot is shown below

When $p = 1$, 6.13 can be written as

$$(y_1 - \beta_1)^2 + \lambda|\beta_1|$$

The plots verify the results in the book



10 Chapter 6 - Applied 7

(a)

$$L(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X},\sigma) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T[\sigma^2\boldsymbol{I}](\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\right)$$

where the first column of $\boldsymbol{X}$ is 1

(b) Given $p(\beta) = \frac{1}{2b}\exp(-\frac{|\beta|}{b})$ Since independent, we have

$$\begin{aligned} p(\boldsymbol{\beta}) &= \prod_{i=1}^{p} \frac{1}{2b}\exp(-\frac{|\beta_i|}{b}) \\ &= \frac{1}{(2b)^p}\exp(-\frac{\|\boldsymbol{\beta}\|_1}{b}) \end{aligned}$$

Then we have

$$P(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X},b,\sigma) \propto (2\pi\sigma^2)^{-\frac{n}{2}}(2b)^{-p}\exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T[\sigma^2\boldsymbol{I}](\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\right)\exp(-\frac{\|\boldsymbol{\beta}\|_1}{b})$$

To find the mode, we formulate the optimization problem

$$\begin{aligned} \arg\max_{\boldsymbol{\beta}} P(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X},b,\sigma) &= \arg\max_{\boldsymbol{\beta}} \log\left[(2\pi\sigma^2)^{-\frac{n}{2}}(2b)^{-p}\exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T[\sigma^2\boldsymbol{I}](\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\right)\exp(-\frac{\|\boldsymbol{\beta}\|_1}{b})\right] \\ &= \arg\min_{\boldsymbol{\beta}} \frac{(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})}{2\sigma^2} + \frac{\|\boldsymbol{\beta}\|_1}{b} \\ &= \arg\min_{\boldsymbol{\beta}} (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}) + \frac{2\sigma^2}{b}\|\boldsymbol{\beta}\|_1 \\ &= \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n}\left[y_i - \sum_{i=0}^{p}\beta_j x_{ij}\right]^2 + \lambda\sum_{i=1}^{p}|\beta_j| \end{aligned}$$

This is the same as the Lasso problem with $\lambda = \frac{2\sigma^2}{b}$ so the estimate will be the same as the Lasso estimate

(c) since $\boldsymbol{\beta}$ is independently normally distributed, we have

$$P(\boldsymbol{\beta}|c) = \prod_{i=1}^{n} P(\beta_i \in \boldsymbol{\beta}|c) \propto \exp\left[-\frac{1}{2}\boldsymbol{\beta}^T[c\boldsymbol{I}]\boldsymbol{\beta}\right]$$

$$P(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X},c,\sigma^2) \propto \exp[-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}) - \frac{1}{2c}\boldsymbol{\beta}^T\boldsymbol{\beta}]$$

Again we formulate the optimization problem

$$\begin{aligned} \arg\max_{\boldsymbol{\beta}} P(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X},c,\sigma^2) &= \arg\max_{\boldsymbol{\beta}} -\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}) - \frac{1}{2c}\boldsymbol{\beta}^T\boldsymbol{\beta} \\ &= \arg\min_{\boldsymbol{\beta}} (\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}) + \frac{\sigma^2}{c}\boldsymbol{\beta}^T\boldsymbol{\beta} \\ &= \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n}\left[y_i - \sum_{i=0}^{p}\beta_j x_{ij}\right]^2 + \lambda\sum_{i=1}^{p}\beta_j^2 \end{aligned}$$

This is the same as the Ridge problem with $\lambda = \frac{\sigma^2}{c}$ so the estimate will be the same as the Ridge estimate

(d) Since the ridge estimate is the solution to the optimization problem, the estimate is the mode of the posterior distribution

Now we to find the mode, we first show the posterior distribution follows normal distribution

$$\begin{aligned} P(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, c, \sigma^2) &\propto \exp[-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) - \frac{1}{2c}\boldsymbol{\beta}^T\boldsymbol{\beta}] \\ &= \exp[-\frac{1}{2\sigma^2}(\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}) - \frac{1}{2c}\boldsymbol{\beta}^T\boldsymbol{\beta}] \\ &= \exp[-\frac{1}{2\sigma^2}(\boldsymbol{y}^T\boldsymbol{y}) + \frac{1}{2}\boldsymbol{\beta}^T(\frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{X} + \frac{1}{c}\boldsymbol{I})\boldsymbol{\beta} - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta}] \\ &= \exp[-\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{d})^T\boldsymbol{\Omega}^{-1}(\boldsymbol{\beta} - \boldsymbol{d})] \end{aligned}$$

where $\boldsymbol{d} = (\boldsymbol{X}^T\boldsymbol{X} + \sigma^2/c\boldsymbol{I})\boldsymbol{X}^T\boldsymbol{y}$, $\boldsymbol{\Omega} = \boldsymbol{X}^T\boldsymbol{X} + \frac{\sigma^2}{c}\boldsymbol{I}$

This kernel shows that the posterior follows the normal distribution with mean $\boldsymbol{d}$ and covariance $\boldsymbol{\Omega}^{-1}$

And mode and mean are the same for normal distribution, so the ridge estimate is also the mode of the posterior distribution