# ST5227 Midterm Assignment Submission

Yeo Ming Jie, Jonathan

2023-03-08

## Question 1: Forward-Backward Selection

For the forward selection variable selection, we start with the null model and sequentially add predictors based on the lowest estimated variance $\sigma^2$ of the model. (Note that the relationship between estimated variance $\hat{\sigma}^2 = \frac{SSE}{n-2}$) We have

$$(\text{null model}) \to (x1, x4) \to (x1, x2, x4) \to (x1, x2, x3, x4)$$

with corresponding $\sigma^2$ given by

$$1.2807 \to 0.6899 \to 0.0899 \to 0.0416 \to 0.0412.$$

Note that the full set of calculated models are:

$$\begin{aligned}(\text{null model}) \to (x1), (x2), (x3), (x4) \to (x1, x2), &(x1, x3), (x1, x4) \\ &\to (x1, x2, x4), (x1, x3, x4) \to (x1, x2, x3, x4)\end{aligned}$$

---

For the backward selection, we start with the full model and sequentially drop predictors based on which predictors (when dropped) leads to the smallest increase in estimated variance. Hence, we have

$$(x1, x2, x3, x4) \to (x1, x2, x4) \to (x1, x4) \to (x1)$$

with corresponding estimated $\sigma^2$ given by

$$0.0412 \to 0.0416 \to 0.0899 \to 0.6899$$

The full set of calculated models are

$$\begin{aligned}(x1, x2, x3, x4) \to (x1, x2, x3), (x1, x2, x3), &(x1, x2, x4), (x2, x3, x4) \\ &\to (x1, x2), (x1, x4), (x2, x4) \to (x1), (x4)\end{aligned}$$

---

The final preferred model depends ultimately on the desired number of predictors to be retained. In view of selecting a final preferred model in which SSE is minimized, however, then the preferred model is $(x1, x2, x3, x4)$ i.e. the full model.

---

## Question 2: Ridge Regression (Theory)

Consider model

$$Y_i = \beta_1 X_i + \varepsilon_i, \quad i = 1, \ldots, n$$

where $X_1, \ldots, X_n$ are non-random values, and $\varepsilon_i, i = 1, \ldots, n$ are IID with $E\varepsilon_i = 0$ and $\mathrm{Var}(\varepsilon_i) = \sigma^2$. Let $\mathbf{X} = (X_1, \ldots, X_n)^{\mathrm{T}}$ and $\mathbf{Y} = (Y_1, \ldots, Y_n)^{\mathrm{T}}$. We estimate $\beta_1$ by minimizing

$$(\mathbf{Y} - \mathbf{X}\beta_1)^{\mathrm{T}} (\mathbf{Y} - \mathbf{X}\beta_1) + \lambda\beta_1^2 = \sum_{i=1}^{n} \{Y_i - \beta_1 X_i\}^2 + \lambda\beta_1^2$$

for some $\lambda \geq 0$.

---

(1) Estimator of $\beta_1$ is given by

$$\hat{\beta}_1(\lambda) = \frac{\sum y_i x_i}{\sum x_i^2 + \lambda}$$

(2) Want to derive $Var(\hat{\beta}_1(\lambda))$. Note that

$$Var(Y_i) = \sigma^2 \Rightarrow Var\left(\sum_{i=1}^{n} y_i x_i\right) = \sum_{i=1}^{n} x_i^2 Var(y_i) = \sigma^2 \sum_{i=1}^{n} x_i^2$$

and hence

$$\begin{aligned} Var(\hat{\beta}_1(\lambda)) &= \left\{\sum_{i=1}^{n} X_i^2 + \lambda\right\}^{-2} Var\left(\sum_{i=1}^{n} y_i x_i\right) \\ &= \frac{\sum x_i^2}{\{\sum X_i^2 + \lambda\}^2} \sigma^2 \end{aligned}$$

(3) Next, note that $Bias(\hat{\beta}_1(\lambda)) = -\lambda(\sum x_i^2 + \lambda)^{-1}\beta_1$. We want to derive $MSE = E(\hat{\beta}_1(\lambda) - \beta_1)^2$.

By the Bias-Variance decomposition, we have

$$\begin{aligned} MSE &= \mathbb{E}\{\hat{\beta}_1(\lambda) - \beta_1\}^2 \\ &= Var(\hat{\beta}_1(\lambda)) + \left\{Bias(\hat{\beta}_1(\lambda))\right\}^2 \\ &= \sigma^2 \left\{\sum_{i=1}^{n} X_i^2 + \lambda\right\}^{-2} \sum_{i=1}^{n} X_i^2 + \lambda^2 \left\{\sum_{i=1}^{n} X_i^2 + \lambda\right\}^{-2} \beta_1^2 \\ &= \frac{\sigma^2 \sum x_i^2 + (\lambda\beta_1)^2}{\{\sum X_i^2 + \lambda\}^2} \end{aligned}$$

(4) Details for choosing $\lambda$ using LOOCV (leave-one-out):

Optimal $\lambda$ st. MSE is minimized. Consider $\lambda \in [0, c]$ for some sufficiently large $c$. For each fixed $\lambda$ in the interval

Compute (where the $j$-th observation is excluded)

$$\hat{\beta}^j_{\text{ridge}}(\lambda) = \left(\sum_{i\neq j} X_i^\top X_i + \lambda I\right)^{-1} \sum_{i\neq j} X_i^\top Y_i.$$

Then define the squared difference between actual and prediction as

$$\text{err}^j(\lambda) = \left(Y_j - X_j \hat{\beta}^j_{\text{ridge}}(\lambda)\right)^2$$

i.e. the prediction error for $(X_j, Y_j)$.

The, by taking average over prediction errors for all entries (left out) we obtain the cross-validation score

$$CV(\lambda) = n^{-1} \sum_{j=1}^{n} err^j(\lambda)$$

Choose $\lambda$ st. $CV(\lambda)$ is minimized.

(5) The effective degrees of freedom is given by

$$\text{df}(\lambda) = \text{trace}\left\{\mathbf{X}\left(\mathbf{X}^{\text{T}}\mathbf{X} + \lambda\right)^{-1}\mathbf{X}^{\text{T}}\right\}$$

Consider

$$\begin{aligned}\text{Cov}(\widehat{\mathbf{Y}}(\lambda), \mathbf{Y})) &= Cov\left(\mathbf{X}\left(\mathbf{X}^{\text{T}}\mathbf{X} + \lambda\right)^{-1}\mathbf{X}^{\text{T}}\mathbf{Y}, \mathbf{Y}\right) \\ &= \mathbf{X}\left(\mathbf{X}^{\text{T}}\mathbf{X} + \lambda\right)^{-1}\mathbf{X}^{\text{T}} \underbrace{Cov(\mathbf{Y}, \mathbf{Y})}_{Var(\mathbf{Y}) = \sigma^2\mathbf{I}}\end{aligned}$$

Then

$$tr\left(\text{Cov}(\widehat{\mathbf{Y}}(\lambda), \mathbf{Y}))\right) = \sigma^2 tr\left(\mathbf{X}\left(\mathbf{X}^{\text{T}}\mathbf{X} + \lambda\right)^{-1}\mathbf{X}^{\text{T}}\right) = \sigma^2\, df(\lambda)$$

and hence

$$\text{df}(\lambda) = \frac{1}{\sigma^2}\,\text{tr}\left(\text{Cov}(\widehat{\mathbf{Y}}(\lambda), \mathbf{Y})\right)$$

(6) Hence

$$\begin{aligned}\text{df}(0) &= \frac{1}{\sigma^2}\,\text{tr}\left(\text{Cov}(\widehat{\mathbf{Y}}(0), \mathbf{Y})\right) \\ &= \frac{1}{\sigma^2} tr\left\{\text{Cov}\left(\mathbf{X}\left(\mathbf{X}^{\text{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\text{T}}\mathbf{Y}, \mathbf{Y}\right)\right\} \\ &= \frac{1}{\sigma^2} tr\left\{\mathbf{X}\left(\mathbf{X}^{\text{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\text{T}} \underbrace{Cov(\mathbf{Y}, \mathbf{Y})}_{Var(\mathbf{Y}) = \sigma^2\mathbf{I}}\right\} = tr\left\{\mathbf{X}^{\text{T}}\mathbf{X}\left(\mathbf{X}^{\text{T}}\mathbf{X}\right)^{-1}\right\} = tr(1) = 1\end{aligned}$$

since $\mathbf{X} = (X_1, \ldots, X_n)^{\text{T}}$ is a vector.

---

## Question 3

(a) For 5 predictors, the two models' coefficients are given by the columns `s3` and `s4` in `out$beta`, with corresponding lambda values 0.3 and 0.4 respectively. Between both models, see that the cross-validation error is minimized where $\lambda = 0.4$.

Hence, estimated model is given by (coefficients from column `s4`)

$$\hat{Y} = 0.3582V_2 - 0.1075V_6 + 0.3535V_{10} - 0.2155V_{14} + 0.4546V_{20}$$

(b) Using 5-fold CV to select the model, the cross-validation error is minimized with value 6.935807 for corresponding lambda value = 0.4. From `out$beta`, this corresponds to column `s4`, i.e. the same 5 predictor variables V2, V6, V10, V14 and V20 are selected.

---

## Question 4

Have the fitted model:

$$\hat{Y}_i = \hat{m}(\mathbf{x}_i) = 1.2 + 1.5\mathbf{x}_i - 0.7\mathbf{x}_i^2 + 0.4(\mathbf{x}_i - 0.3)_+^3 \quad \text{for } i = 1, \ldots, 100$$

The 95% CI for $E\hat{Y}$ i.e. prediction interval is

$$E\hat{Y} \pm 1.96\left\{\mathbf{u}^T\hat{\Sigma}\mathbf{u}\right\}^{1/2} = 2.1372 \pm 1.96(0.03844047) = [2.061857, 2.212543]$$

where $\mathbf{u} = (1, x, x^2, (x-0.3)_+^3)^T$, $\hat{\Sigma} = 0.2^2(\mathbb{X}^T\mathbb{X})^{-1}$ and $E\hat{Y} = m(1.0) = 2.1372$.

The R code for the calculations are given below:

```
Y_hat <- 1.2 + 1.5 - 0.7 + 0.4*(0.7)^3
u <- c(1, 1, 1, 0.7^3)
sigma_hat <- matrix(c(0.0171, -0.0047, -0.0105, 0.0061,
                      -0.0047, 0.0235, 0.0113, -0.0149,
                      -0.0105, 0.0113, 0.0181, -0.0146,
                      0.0061, -0.0149, -0.0146, 0.0178), ncol = 4, byrow = TRUE)
sigma_hat = 0.2^2 * sigma_hat
sqrt(t(u) %*% sigma_hat %*% u)
```

```
##             [,1]
## [1,] 0.03844047
```

---

## Question 5: Handwritten Digit Recognition

```
# Note: Digit ID -> 256 grayscale values for each row (one data entry)
zip_train <- read.table('zip.train')
zip_test <- read.table('zip.test')

# Filtering dataset for digit ID = 2 or 3
bin_test <- zip_test[zip_test[,1] == 2 | zip_test[,1] == 3,]
bin_train <- zip_train[zip_train[,1] == 2 | zip_train[,1] == 3,]
```

**(a) Logistic LASSO regression**

```
library(glmnet)
```

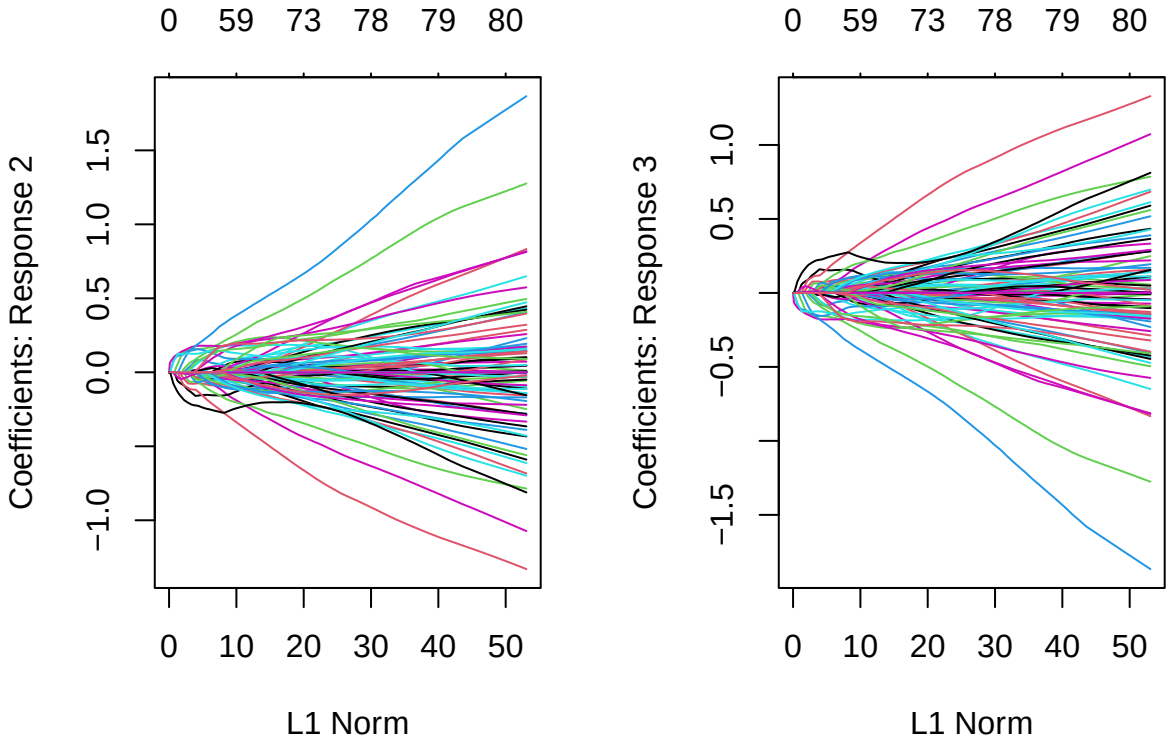
```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-6
```

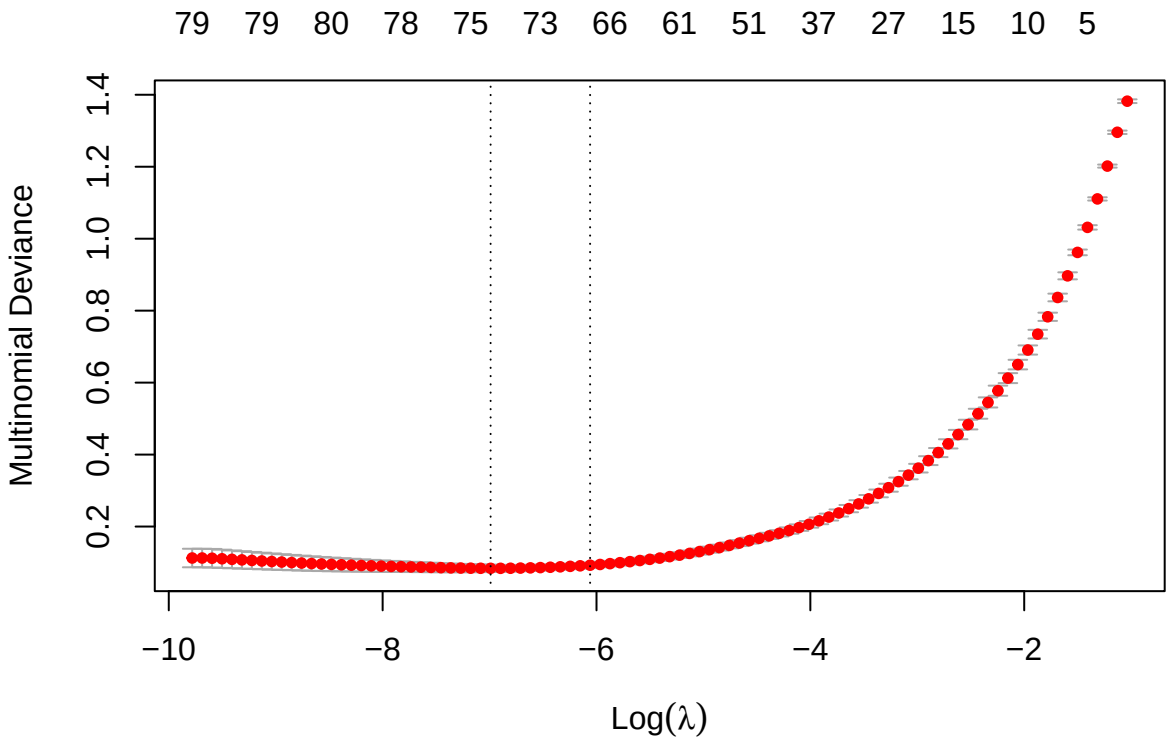
```
# Training and Test Data
y_train <- as.factor(bin_train[,1]); y_test <- bin_test[,1]
x_train <- as.matrix(bin_train[,2:257]); x_test <- as.matrix(bin_test[,2:257])

# Normalize
x_train <- scale(x_train); x_test <- scale(x_test)

# Fit logistic LASSO (solution path)
fm <- glmnet(x_train, y_train, family = 'multinomial')
par(mfrow = c(1,2)); plot(fm)
```




```
# 5-Fold Cross-Validation for Optimal Lambda tuning parameter
set.seed(5227)
cv <- cv.glmnet(x_train, y_train, family = 'multinomial', nfolds = 5)
lambda = cv$lambda.min
par(mfrow = c(1,1)); plot(cv)
```




```
# Fit logistic Lasso (Optimal lambda)
fmLasso <- glmnet(x_train, y_train, family = 'multinomial', lambda = lambda)

# To obtain estimates: (uncomment below)
# fmLasso$a0; fmLasso$beta

# Predicted Probabilities and Classes, Misclassification Rate
yProb = predict(fmLasso, newx = x_test, type = "response")
yClass = predict(fmLasso, newx = x_test, type = "class")
classificationError = mean(yClass != y_test)

# Print Outputs (first 5 values)
head(yProb); head(yClass)
```

```
## , , s0
##
##               2           3
## 3  0.0006203305 9.993797e-01
## 12 0.9999992517 7.483160e-07
## 13 0.7861100437 2.138900e-01
## 16 0.0002485065 9.997515e-01
## 21 0.9999999329 6.708571e-08
## 22 0.9999998835 1.165391e-07
```

```
##      s0
## [1,] "3"
## [2,] "2"
## [3,] "2"
## [4,] "3"
## [5,] "2"
## [6,] "2"
```

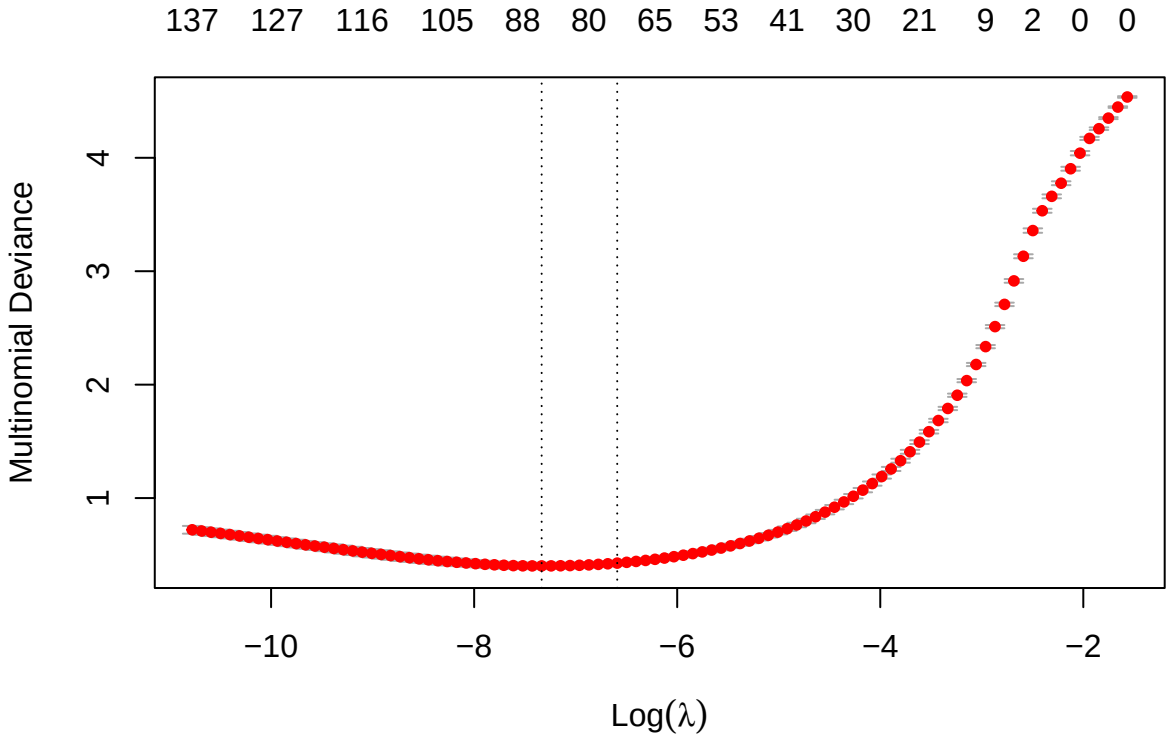
```
classificationError
```

```
## [1] 0.0467033
```

### (b) Multilevel Logistic LASSO regression

```
# Training and Test Data
y_train <- as.factor(zip_train[,1]); y_test <- zip_test[,1]
x_train <- as.matrix(zip_train[,2:257]); x_test <- as.matrix(zip_test[,2:257])

# 5-Fold Cross-Validation for Optimal Lambda tuning parameter
set.seed(5227)
cv <- cv.glmnet(x_train, y_train, family = 'multinomial', nfolds = 5)
lambda = cv$lambda.min
par(mfrow = c(1,1)); plot(cv)
```




```
# LASSO at Optimal Lambda
multi_lasso_est = coef(cv, s = "lambda.min")

# Predicted Probabilities and Classes, Misclassification Rate
yProb = predict(cv, newx = x_test, s = "lambda.min", type = "response")
yClass = predict(cv, newx = x_test, s = "lambda.min", type = "class")
classificationError = mean(yClass != y_test)

# Print Outputs (first 5 values)
round(head(yProb),4); head(yClass)
```

```
## , , 1
##
##           0 1      2      3      4      5      6      7      8      9
## [1,] 0.0000 0 0.0001 0.0003 0.0095 0.0000 0.0000 0.0115 0.0038 0.9749
```

```
## [2,] 0.0013 0 0.0128 0.0000 0.0003 0.0079 0.9777 0.0000 0.0000 0.0000
## [3,] 0.0000 0 0.0019 0.9969 0.0000 0.0000 0.0000 0.0000 0.0012 0.0000
## [4,] 0.3658 0 0.0401 0.0000 0.0008 0.0483 0.5440 0.0009 0.0000 0.0000
## [5,] 0.0029 0 0.0060 0.0000 0.0000 0.0016 0.9895 0.0000 0.0000 0.0000
## [6,] 0.9995 0 0.0001 0.0000 0.0000 0.0000 0.0004 0.0000 0.0001 0.0000
```

```
##      1
## [1,] "9"
## [2,] "6"
## [3,] "3"
## [4,] "6"
## [5,] "6"
## [6,] "0"
```

```
classificationError
```

```
## [1] 0.08669656
```