homework-2

Name: Brian Deng

```r
library(bis557)
```

Name: Brian Deng (BIS557 HW2)

Question 1

Assume there are n observations. Then, let Y be an $n \times 1$ matrix, X be an $n \times 2$ matrix, and β be a 2×1 matrix.

Then, we have:

$$
\begin{aligned}
\hat{\beta} &= \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} \\
&= (X^T X)^{-1} X^T Y \\
&= \begin{pmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum y_i \\ \sum x_i y_i \end{pmatrix} \\
&= \frac{1}{n \sum x_i^2 - n^2 \bar{X}^2} \begin{pmatrix} \sum x_i^2 & -\sum x_i \\ -\sum x_i & n \end{pmatrix} \begin{pmatrix} \sum y_i \\ \sum x_i y_i \end{pmatrix} \\
&= \frac{1}{n \sum x_i^2 - n^2 \bar{X}^2} \begin{pmatrix} n\overline{X^2} & -n\bar{X} \\ -n\bar{X} & n \end{pmatrix} \begin{pmatrix} n\bar{Y} \\ \sum x_i y_i \end{pmatrix} \\
&= \frac{1}{\sum x_i^2 - n\bar{X}^2} \begin{pmatrix} \overline{X^2} & -\bar{X} \\ -\bar{X} & 1 \end{pmatrix} \begin{pmatrix} n\bar{Y} \\ \sum x_i y_i \end{pmatrix} \\
&= \frac{1}{\sum x_i^2 - 2\bar{X} \sum x_i + n\bar{X}^2} \begin{pmatrix} n\overline{X^2}\bar{Y} - \bar{X} \sum x_i y_i \\ -n\bar{X}\bar{Y} + \sum x_i y_i \end{pmatrix} \\
&= \frac{1}{\sum (x_i - \bar{X})^2} \begin{pmatrix} \bar{Y} \sum x_i^2 - n\bar{X}^2\bar{Y} + n\bar{X}^2\bar{Y} - \bar{X} \sum x_i y_i \\ \sum x_i y_i - \bar{Y} \sum x_i - \bar{X} \sum y_i + n\bar{X}\bar{Y} \end{pmatrix} \\
&= \frac{1}{\sum (x_i - \bar{X})^2} \begin{pmatrix} \bar{Y}(\sum x_i^2 - n\bar{X}^2) - \bar{X}(-n\bar{X}\bar{Y} + \sum x_i y_i) \\ \sum (x_i - \bar{X})(y_i - \bar{Y}) \end{pmatrix} \\
&= \frac{1}{\sum (x_i - \bar{X})^2} \begin{pmatrix} \bar{Y} \sum (x_i - \bar{X})^2 - \bar{X} \sum (x_i - \bar{X})(y_i - \bar{Y}) \\ \sum (x_i - \bar{X})(y_i - \bar{Y}) \end{pmatrix} \\
&= \begin{pmatrix} \frac{\bar{Y} \sum (x_i - \bar{X})^2}{\sum (x_i - \bar{X})^2} - \bar{X} \frac{\sum (x_i - \bar{X})(y_i - \bar{Y})}{\sum (x_i - \bar{X})^2} \\ \frac{\sum (x_i - \bar{X})(y_i - \bar{Y})}{\sum (x_i - \bar{X})^2} \end{pmatrix} \\
&= \begin{pmatrix} \bar{Y} - \hat{\beta}_1 \bar{X} \\ \frac{\sum (x_i - \bar{X})(y_i - \bar{Y})}{\sum (x_i - \bar{X})^2} \end{pmatrix}.
\end{aligned}
$$

Therefore, we have:

$$\hat{\beta}_1 = \frac{\sum(x_i - \bar{X})(y_i - \bar{Y})}{\sum(x_i - \bar{X})^2}; \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1\bar{X}.$$

Question 2

The function will be called `bis557::ols_gd_hw2b()`.

Here, we will compare this function to the OLS model `lm()`.

```
# Use formula to calculate coefficients and penalty
data(iris)
b_gd <- bis557::ols_gd_hw2b(form = Sepal.Length ~ ., d = iris[,-5],
                            b_0 = rep(1e-9,4), learn_rate = 2,
                            max_iter = 2e4)
print(cbind("lm()" = lm(Sepal.Length ~ ., iris[,-5])$coefficients,
            "ols_gd_hw2b()" = b_gd$coefficients))
#>                    lm() ols_gd_hw2b()
#> (Intercept)   1.8559975     0.9699853
#> Sepal.Width   0.6508372     0.8794688
#> Petal.Length  0.7091320     0.8138332
#> Petal.Width  -0.5564827    -0.7471755
cat("\n")
print(paste0("CV_penalty_MSE = ", sprintf("%.4f", b_gd$CV_penalty_MSE)))
#> [1] "CV_penalty_MSE = 0.1529"
```

Here, we see that the penalty based on the out-of-sample 10-fold CV MSE is estimated to be around 0.16.

The estimated coefficients using gradient descent vs. the true coefficients are slightly different, since there are many local minima to be optimized.

Question 3

We will use the function `bis557::ridge_hw2c()` for ridge regression, where the penalty L equals:

$$L = \frac{1}{2n}||Y - X\beta||_2^2 + \lambda||\beta||_2^2$$

From the textbook, we solve using the formula:

$$\hat{\beta}_{ridge} = (X^T X + \lambda I_p)^{-1} X^T Y$$

Remember that for SVD, we have $X = U\Sigma V^T$. Then (from the textbook), another way to write the estimated coefficients is:

$$\hat{\beta}_{ridge} = V \cdot \text{Diag}\left(\frac{\sigma_1}{\sigma_1^2 + \lambda}, \cdots\right) U^T Y$$

We show that as $\lambda \to \infty$, then $\hat{\beta}_{ridge} \to 0$.

```
# Show ridge regularization
data(iris)
b_ridge <- bis557::ridge_hw2c(form = Sepal.Length ~ ., d = iris,
                              lambda_val = 10)
print(cbind("lm()" = lm(Sepal.Length ~ ., iris)$coefficients,
            "lam=10" = b_ridge$coefficients))
#>                          lm()
#> (Intercept)        2.1712663  0.5321332
#> Sepal.Width        0.4958889  1.0159230
#> Petal.Length       0.8292439  0.6328160
#> Petal.Width       -0.3151552 -0.1712601
#> Speciesversicolor -0.7235620  0.1008790
#> Speciesvirginica  -1.0234978 -0.1099248
cat("\n")

# Show that ridge regression works for colinear regression variables
data(lm_patho)
b_patho <- bis557::ridge_hw2c(form = y ~ ., d = lm_patho,
                              lambda_val = 1)
print(cbind("lm()" = lm(y ~ ., lm_patho)$coefficients,
            "lam=1" = b_patho$coefficients))
#>                      lm()
#> (Intercept) 1.003095e-05  5.00000e-06
#> x1          1.000000e+00  1.00000e+00
#> x2                    NA -9.50015e-10
```

Question 4

The method for optimizing the ridge parameter λ is below. We will use the function `bis557::ridge_hw2d()`.

```
# Show the most optimal lambda
data(iris)
b_ridge <- bis557::ridge_hw2d(form = Sepal.Length ~ ., d = iris[,-5],
                              lambda_vals = 10^seq(-3, 2, length = 200))
print(b_ridge)
#> $min_loss
#> [1] 0.0994243
#>
#> $min_lambda
#> [1] 0.03612343
cat("\n")
b_best <- bis557::ridge_hw2c(form = Sepal.Length ~ ., d = iris[,-5],
                              lambda_val = b_ridge$min_lambda)
print(cbind("lm()" = lm(Sepal.Length ~ ., iris[,-5])$coefficients,
            "best_lambda" = b_best$coefficients))
#>                     lm()
#> (Intercept)   1.8559975  1.8222266
#> Sepal.Width   0.6508372  0.6597411
#> Petal.Length  0.7091320  0.7114400
#> Petal.Width  -0.5564827 -0.5586203
```

Question 5

Here, we let j be a predictor. Here, Y is a column vector with length n, and β is a column vector with length p. Also, X is a matrix with dimension $n \times p$. For notation purposes, let X_j be the j-th column of X.

Then, we would minimize (using the gradient w.r.t. β):

$$L(\beta) = \frac{1}{2n}||Y - X\beta||_2^2 + \lambda||\beta||_1 = \frac{1}{2n}||Y - \sum_{j=1}^{p} \beta_j X_j||_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$

So, for each partial derivative, we have (separate cases for $\beta_j > 0$, $\beta_j < 0$, and $\beta_j = 0$):

$$\begin{aligned} \frac{\partial L}{\partial \beta_j} &= 0 \\ &= \frac{1}{n}(X_j^T X_j \beta_j - X_j^T Y) + \lambda \cdot \text{sign}(\beta_j). \end{aligned}$$

For $\beta_j > 0$, we have:

$$\hat{\beta}_j^{LASSO} = \left(\frac{X_j^T X_j}{n}\right)^{-1} \left(\frac{X_j^T Y}{n} - \lambda\right).$$

For $\beta_j < 0$, we have:

$$\hat{\beta}_j^{LASSO} = \left(\frac{X_j^T X_j}{n}\right)^{-1} \left(\frac{X_j^T Y}{n} + \lambda\right).$$

These both work only when $\left|\frac{X_j^T Y}{n}\right| > \lambda$, so that the signs are consistent. We know that the norm $X_j^T X_j \geq 0$. Otherwise, we have $\beta_j = 0$ when $\left|\frac{X_j^T X_j}{n}\right| \leq \lambda$.

Therefore, we can say that the solution is:

$$\hat{\beta}_j^{LASSO} = \text{sign}\left(\frac{X_j^T Y}{n}\right) \cdot \left(\frac{X_j^T X_j}{n}\right)^{-1} \cdot \max\left(\left|\frac{X_j^T Y}{n}\right| - \lambda, 0\right).$$

Thus, the statement "$\hat{\beta}_j^{LASSO} = 0$ if $|X_j^T Y| \leq n\lambda$" is true.