Stats 102B HW 4

Bryan Mui - UID 506021334

Due Wed, June 4, 11:00 pm

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
train <- read_csv("train_data.csv")
```

```
Rows: 600 Columns: 601
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (601): X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15,...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
val <- read_csv("validation_data.csv")
```

```
Rows: 200 Columns: 601
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (601): X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15,...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Problem 1

Consider the function

$$f(x) = \frac{1}{4}x^4 - x^2 + 2x$$

Part (α)

Using the pure version of Newton's algorithm report x_k for $k = 20$ (after running the algorithm for 20 iterations) based on the following 5 initial points:

1. $x_0 = -1$
2. $x_0 = 0$
3. $x_0 = 0.1$
4. $x_0 = 1$
5. $x_0 = 2$

Newton's pure algorithm is as follows:

1. Select $x_0 \in \mathbb{R}^n$
2. While stopping criterion > tolerance do:
    1. $x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$
    2. Calculate value of stopping criterion($|f(x_{k+1}) - f(x_k)| \leq \epsilon$)

Gradient: $\nabla f(x) = \frac{\partial}{\partial x} = f'(x) = x^3 - 2x + 2$

Hessian: $\nabla^2 f(x) = \frac{\partial^2}{\partial x^2} = f''(x) = 3x^2 - 2$

```
# params
max_iter <- 20
starting_points <- c(-1, 0, 0.1, 1, 2)
stopping_tol <- 1e-6


# algorithm
newton_pure_alg <- function(max_iter, starting_point, stopping_tol) {
  beta <- starting_point
  iterations_ran <- 0
  betas_vec <- c(beta)

  obj <- function(x) {
  return(1/4 * x^4 - x^2 + 2*x)
  }
  grad <- function(x) {
    x^3 - 2*x + 2
  }
  hessian <- function(x) {
    3*x^2 - 2
  }

  for(i in 1:max_iter) {
    beta_new <- beta - (grad(beta) / hessian(beta))
    betas_vec[i+1] <- beta_new
    if(abs(beta_new - beta) <= stopping_tol) { break }
    beta <- beta_new
```

```r
  }
  iterations_ran <- i
  return(list(iterations=iterations_ran, betas=betas_vec))
}

# running the alg
for (starting_point in starting_points) {
  result <- newton_pure_alg(max_iter, starting_point, stopping_tol)
  cat("Starting Point:", starting_point, "\nIterations:", result$iterations, "\nBetas:",
   ↪  result$betas,"\n", "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~", "\n")
}
```

```
Starting Point: -1
Iterations: 8
Betas: -1 -4 -2.826087 -2.146719 -1.842326 -1.772848 -1.769301 -1.769292 -1.769292
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Starting Point: 0
Iterations: 20
Betas: 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Starting Point: 0.1
Iterations: 20
Betas: 0.1 1.014213 0.07965577 1.009099 0.05222653 1.003965 0.02332944 1.000804 0.004806795 1.000035 0.
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Starting Point: 1
Iterations: 20
Betas: 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Starting Point: 2
Iterations: 9
Betas: 2 1.4 0.8989691 -1.288779 -2.105767 -1.8292 -1.771716 -1.769297 -1.769292 -1.769292
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Part (i) What do you observe?

Part (ii) How can you fix the issue reported in (i)?

Problem 2

Consider the data in the train data.csv file. The first 600 columns correspond to the predictors and the last column to the response y.

Part (i) Implement that proximal gradient algorithm for Lasso regression, by modifying appropriately your code from Homework 1.

To select a good value for the regularization parameter λ use the data in the validation data.csv to calculate the sum-of-squares error validation loss.

Part(ii) Show a plot of the training and validation loss as a function of iterations. Report the number of regression coefficients estimated as zero based on the best value of λ you selected.