HW 2

Bryan Mui - UID 506021334 - 28 April 2025

Loaded packages: ggplot2, tidyverse (include = false for this chunk)

Reading the dataset:

```r
data <- read_csv("dataset-logistic-regression.csv")
```

```
Rows: 10000 Columns: 101
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (101): y, X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X...

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
head(data, n = 25)
```

```
# A tibble: 25 x 101
       y      X1      X2     X3     X4     X5     X6      X7     X8      X9
   <dbl>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>   <dbl>  <dbl>   <dbl>
 1     1 -0.0895  0.450   1.71   0.657 -0.392  1.24   0.895   1.13  -0.0117
 2     1 -0.0943  0.281  -0.147 -0.701  0.400 -0.210  0.677  -0.440  0.458
 3     0 -0.431  -0.445  -0.777 -0.832 -2.26  -1.62  -1.98   -1.67  -1.15
 4     0  0.644   0.0817 -0.448  0.852 -1.02   0.671  0.299   0.145 -0.205
 5     1 -0.919  -0.0241  0.807 -0.612 -0.498  0.350  1.12    0.242 -0.947
 6     0 -1.89   -1.11   -0.210  0.161 -1.34  -2.04  -0.0135 -1.39  -1.31
 7     0 -1.34   -0.804   0.322 -0.110  0.624 -0.329 -0.432  -0.191  0.171
 8     1  0.329   0.468   0.719  0.588  1.71   1.39   0.603   0.650  0.161
 9     0  0.332   1.42   -0.431  1.02   0.484  0.348  0.474   1.26  -0.479
10     0 -0.311   0.0193  0.168 -0.346  0.626 -0.704 -0.290   0.680 -0.0453
# i 15 more rows
```

```
# i 91 more variables: X10 <dbl>, X11 <dbl>, X12 <dbl>, X13 <dbl>, X14 <dbl>,
#   X15 <dbl>, X16 <dbl>, X17 <dbl>, X18 <dbl>, X19 <dbl>, X20 <dbl>,
#   X21 <dbl>, X22 <dbl>, X23 <dbl>, X24 <dbl>, X25 <dbl>, X26 <dbl>,
#   X27 <dbl>, X28 <dbl>, X29 <dbl>, X30 <dbl>, X31 <dbl>, X32 <dbl>,
#   X33 <dbl>, X34 <dbl>, X35 <dbl>, X36 <dbl>, X37 <dbl>, X38 <dbl>,
#   X39 <dbl>, X40 <dbl>, X41 <dbl>, X42 <dbl>, X43 <dbl>, X44 <dbl>, ...
```

Our data set has 10000 observations, 1 binary outcome variable y, and 100 predictor variables X1-X100

Separating into X matrix and y vector:

```
X <- data %>%
  select(-y)
y <- data %>%
  select(y)
```

Problem 1

Part (α)

The optimization problem is to minimize the log-likelihood function. From there we will get the objective function and gradient function

From the slides in class we have:

$$\min_{\beta}(-\ell(\beta)) = \frac{1}{m}\sum_{i=1}^{m} f_i(\beta)$$

and the equation for $f_i(\beta)$:

$$f_i(\beta) = -y_i(x^\top \beta) + log(1 + exp(x_i^\top \beta))$$

For the objective function, we get:

$$\boxed{f(\beta) = \frac{1}{m}\sum_{i=1}^{m}[-y_i(x^\top \beta) + log(1 + exp(x_i^\top \beta))]}$$

We also have the gradient function:

$$\nabla f(x) = \frac{1}{m}\sum_{i=1}^{m} \nabla f_i(x)$$

and

$$\nabla_\beta f_i(\beta) = [\sigma(x_i^\top \beta) - y_i] \cdot x_i$$

where $\sigma(z) = \frac{1}{1+exp(-z)}$ as the logistic sigmoid function, therefore:

$$\nabla f(x) = \frac{1}{m}\sum_{i=1}^{m} \nabla f_i(x),\ \nabla_\beta f_i(\beta) = [\sigma(x_i^\top \beta) - y_i] \cdot x_i$$
$$\nabla f(\beta) = \boxed{\frac{1}{m}\sum_{i=1}^{m} [\sigma(x_i^\top \beta) - y_i] \cdot x_i}$$

Therefore our gradient descent update step is:

$$\boxed{\beta_{k+1} = \beta_k - \eta \nabla f(\beta_k)}$$

Implement the following algorithms to obtain estimates of the regression coefficients $\boldsymbol{\beta}$:

(1) Gradient descent with backtracking line search

Algorithm; Backtracking Line Search:

Params:

- Set $\eta^0 > 0$(usually a large value ~1),
- Set $\eta_1 = \eta^0$
- Set $\epsilon \in (0, 1), \tau \in (0, 1)$, where ϵ and τ are used to modify step size

Repeat:

- At iteration k, set $\eta_k < -\eta_{k-1}$
    1. Check whether the Armijo Condition holds:

       $$h(\eta_k) \leq h(0) + \epsilon \eta_k h'(0)$$

       where $h(\eta_k) = f(x_k) - \eta_k \nabla f(x_k)$,
       and $h(0) = f(x_k)$,
       and $h'(0) = -||\nabla(x_k)||^2$

2.

- If yes(condition holds), terminate and keep η_k
- If no, set $\eta_k = \tau\eta_k$ and go to Step 1

Stopping criteria: Stop if $||x_k - x_{k+1}|| \leq tol$ (change in parameters is small)

```
# logistic gradient descent w/ bls
log_bls <- function(X, y, eta = NULL, tol = 1e-6, max_iter = 10000, xi = 0.5,
↪  epsilon = 0.5, tau = 0.5) {
  # Initialize
  n <- nrow(X)
  p <- ncol(X)
  beta <- rep(0, p)
  obj_values <- numeric(max_iter)
  eta_values <- numeric(max_iter)  # To store eta values used each iteration
  beta_values <- list() # To store beta values used each iteration
  eta_bt <- 1  # Initial step size for backtracking

  # Objective function: negative log-likelihood
  obj_function <- function(beta) {
    #sum((X %*% beta - y)^2) / (2 * n)
  }

  # Gradient function
  gradient <- function(beta) {
    #t(X) %*% (X %*% beta - y) / n
  }

  # Algorithm:
  for (iter in 1:max_iter) {
    grad <- gradient(beta)
    beta_values[[iter]] <- beta

    # backtracking step
    if (iter == 1) {
      eta_bt <- 1 # Reset only in the first iteration
      y_k <- beta
    }
    else {
      beta_prev <- beta_values[[iter - 1]]
      y_k <- beta + xi * (beta - beta_prev)
    }
    beta_new <- y_k - eta_bt * grad
```

```
    while (obj_function(beta_new) > obj_function(beta) - epsilon * eta_bt *
     ↪  sum(grad^2)) {
      eta_bt <- tau * eta_bt
      beta_new <- beta - eta_bt * grad
    }
    eta_used <- eta_bt

    eta_values[iter] <- eta_used

    obj_values[iter] <- obj_function(beta_new)

    if (sqrt(sum((beta_new - beta)^2)) < tol) {
      obj_values <- obj_values[1:iter]
      eta_values <- eta_values[1:iter]
      break
    }

    beta <- beta_new
  }

  return(list(beta = beta, obj_values = obj_values, eta_values = eta_values,
   ↪  beta_values = beta_values))
}
```

(2) Gradient descent with backtracking line search and Nesterov momentum

Nesterov is simply BLS with a special way to select the momentum ϵ

(3) Gradient descent with AMSGrad-ADAM momentum

(no backtracking line search, since AMSGrad-ADAM adjusts step sizes per parameter using momentum and adaptive scaling)

(4) Stochastic gradient descent with a fixed schedule of decreasing step sizes

(5) Stochastic gradient descent with AMSGrad-ADAM-W momentum

Part (a)

Part (b)