Homework 2

Stats 102B Lec 1 and 2

Spring 2024

General Guidelines

Show all your work, including any and all relevant code and output. Any and all collaboration must adhere to **Level 1** collaboration described in the Stats 102B Collaboration Policy.

Please submit your homework as a single file, in PDF format only. Name your assignment file with the convention `123456789_stats102b_hw0.pdf`, where `123456789` is replaced with your UID and `hw0` is updated to the actual homework number. Include your first and last name and UID in your assignment as well.

All R code is expected to follow the Tidyverse Style Guide: https://style.tidyverse.org/. If you scan or take a picture of any written work, please make sure the resolution is high enough that your work is clear and legible. Submissions with severe style or formatting issues may receive a penalty. Any submission that cannot be properly read will not be graded.

Question 1

(a)

Show that $S = \{x \in \mathbb{R}^n : ||x||_2 \leq r\}$ is a convex set, where $r > 0$ is a constant and $||x||_2 = \sqrt{x^T x}$ is the Euclidean norm of x.

(b)

Show that $f(x) = |x|$ is a convex function.

(c)

Show that $f(x) = x^2$ is a convex function.

(d)

For $f : \mathbb{R}^2 \to \mathbb{R}$, show that

$$f(x) = \frac{1}{2}x^T \begin{bmatrix} 2 & 4 \\ 0 & 6 \end{bmatrix} x + x^T \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 1$$

is convex.

Hint: You have multiple approaches for showing a function is convex. Also, the triangle inequality may be helpful.

Question 2

(a)

Write a function `grad_desc(g, grad_g, alpha, w0, K)` that runs the one-dimensional version of gradient descent with the following parameters:

- `g` is a function $g : \mathbb{R} \to \mathbb{R}$
- `grad_g` is the gradient of g, a function $\nabla g : \mathbb{R} \to \mathbb{R}$
- `alpha` is the step size
- `w0` is the initial point
- `K` is the number of iterations

The output should be a list object with the following components:

- `$index` should represent w^*, the local minimum
- `$val` should represent $g(w^*)$, the value of the function at the local minimum
- `$coord_matrix` a matrix object with coordinates formatted however you like

Note: You should never print your full `grad_desc()` output object or the full `coord_matrix` component.

(b)

Use your gradient descent function to find the local minimum of

$$g(w) = \tan(w^2 + \sin(w)).$$

Plot the function as well as the coordinates and lines similar to Chapter 3 slide 41 for the following parameters:

- $\alpha = 0.1, w_0 = 0.5, K = 20$
- $\alpha = 0.01, w_0 = 0.5, K = 200$
- $\alpha = 0.8, w_0 = 0.5, K = 20$

(c)

Choose an appropriate step size α, initial point w_0, and maximum iterations K to find the local minimum of

$$g(w) = \frac{1}{4}w^4 - \frac{3}{2}w^3 + \frac{1}{2}w^2 + \sin(w).$$

Plot the function as well as the coordinates and lines similar to Chapter 3 slide 41.

Question 3

(a)

Modify your function in Question 2 to accommodate functions from $\mathbb{R}^n$ to $\mathbb{R}$. More specifically, write a function `grad_desc_n(g, grad_g, alpha, w0, K)` that runs gradient descent with the following parameters:

- `g` is a function $g : \mathbb{R}^n \to \mathbb{R}$
- `grad_g` is the gradient of g, a function $\nabla g : \mathbb{R}^n \to \mathbb{R}^n$
- `alpha` is the step size
- `w0` is the initial point
- `K` is the number of iterations

The output should be a list object with the following components:

- `$index` should represent w^*, the local minimum
- `$val` should represent $g(w^*)$, the value of the function at the local minimum

(b)

Verify that your function works by finding the local minimum of

$$g(w_1, w_2) = (w_1 - 2)^2 + (w_2 - 4)^2 - 1.$$

You should be able to find the local minimum (in this case it is also the global minimum via convexity) by inspection. Choose an appropriate step size α, w_0, and K that convinces you that it works, because you will use this function in later questions.

(c)

Verify that your function works by finding the local minimum of

$$g(w_1, w_2, w_3) = \sin(w_1 + 2) + \cos(w_2 - 4) + w_3^2.$$

Choose a couple of initial points and try to identify the formula for all local minima. (An example of a formula might be $(w_1 = k, w_2 = 2 + k', x_3 = 3)$ where $k, k' \in \mathbb{R}$).

Question 4

Recall that the probability density function of a gamma distribution with parameters $\alpha, \beta > 0$ is defined by

$$f(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x},$$

where $\Gamma(\alpha) = \displaystyle\int_0^\infty x^{\alpha-1} e^{-x} dx$ is the gamma function.

To compute the maximum likelihood estimator, you may be tempted to compute the log-likelihood and take partial derivatives with respect to α and β and attempt to solve a system of equations as you usually do. However, it turns out that there is no closed-form solution. You may be tempted to try gradient descent, but try calculating the gradient of a function that has the gamma function. Instead, we will try to compute a maximum likelihood estimate by using gradient descent, but by numerically calculating the gradient at each step.

(a)

Let $x_1, x_2, \ldots, x_n$ be independent and identically distributed values from a Gamma(α, β) distribution. Show that the negative log-likelihood of α and β can be written as

$$-\log L(\alpha, \beta) = -\left[n\left(\alpha \log \beta - \log \Gamma(\alpha)\right) + (\alpha - 1) \sum_{i=1}^{n} \log(x_i) - \beta \sum_{i=1}^{n} x_i \right]$$

(b)

Read in the `.rds` object from Bruin Learn which contains 100 observed values from an unknown gamma distribution. First write a function for the negative log-likelihood above, then modify your function in Question 3 by eliminating the `grad_g` parameter, adding an `h` parameter, and instead numerically calculating the gradient at each step. (See Chapter 3 slide 37 if you are confused). Run your algorithm on the negative log-likelihood with $\alpha = 0.005,\ w_0 = (1.5, 1)^T,\ K = 200,\ h = 0.0001$ and print the `$index` of your output list.

Hint: You can use the `readRDS()` function to read in `.rds` files, and the `gamma()` function to compute $\Gamma(\alpha)$.

(c)

Implement the momentum-based gradient descent with $\alpha = 0.01,\ w_0 = (1.5, 1)^T,\ K = 50,\ h = 0.0001,$ and $\beta \in \{0.1, 0.3, 0.5, 0.7, 0.9, 0.99\}$, and print the `$index` of your output list for each time. Describe what happens as β varies. It may also help to print the `$value`.

Question 5

Consider the function

$$g(w_1, w_2) = w_1^8 + w_2^8$$

Modify your function from Question 4(b) to run the normalized gradient descent with a numeric gradient calculation and $\varepsilon = 0.0001$ for $\alpha = 0.005,\ w_0 = (1.5, 1)^T,\ K = 2000,\ h = 0.0001$ and print the `$index` of your output list. Run your original function from 4b with the same set of parameters (but no ε) and output the `$index`. What can you conclude and why?