Homework 3 – Part II – Steepest Ascent Algorithm: 30 points

Exercise J-2.2: In this exercise, we assume that we have a set of data $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$ from a p-variate normal distribution with mean $\boldsymbol{\mu} = \left(\mu_1, \mu_2, \cdots, \mu_p\right)^T$ and a p by p covariance matrix $\boldsymbol{\Sigma} = (\sigma_{ij})$.

You are to write a general function to maximize the log-likelihood function with respect to the parameters in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$**.** The log-likelihood function is given by

$$\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma} | \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n) = -\frac{1}{2}\{np\log(2\pi) + n\log|\boldsymbol{\Sigma}| + trace(\Sigma^{-1}C(\boldsymbol{\mu}))\}$$

where

$$C(\boldsymbol{\mu}) = \sum_{i=1}^{n} [(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T].$$

Note that there are p parameters in $\boldsymbol{\mu}$ and $p(p+1)/2$ parameters in $\boldsymbol{\Sigma}$**,** since $\sigma_{ij} = \sigma_{ji}$. Order the parameters as follows: $\boldsymbol{\theta} = \left(\mu_1, \mu_2, \cdots, \mu_p, \sigma_{11}, \sigma_{21}, \sigma_{22}, \sigma_{31}, \sigma_{32}, \sigma_{33}, \cdots,, \sigma_{p1}, \sigma_{p2}, \cdots, \sigma_{pp}\right)^T$.

Write a general code that applies the Steepest ascent method with step-halving to obtain the maximum likelihood estimate of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for a given set of $n \times p$ matrix of data:

Implement the following stopping/convergence criteria:

(i) Modified Relative error = max $\left|\frac{\boldsymbol{\theta}_j^{(n+1)} - \boldsymbol{\theta}_j^{(n)}}{\max(1, |\boldsymbol{\theta}_j^{(n+1)}|)}\right| < tolerr$ where maximum is taken over all $\left[p + \frac{p(p+1)}{2}\right]$ possible values of θ.

(ii) $\left\|\nabla\ell(\theta^{(n+1)}\right\|_2 < tolgrad$

Stop if (i) & (ii) hold or a specified maximum number of iterations (*maxit*) is reached. Note: *maxit, tolerr* and *tolgrad* are parameters to your function.

In writing your code pay attention to the following:
Write separate functions to compute the log-likelihood and the gradient of the log-likelihood. Also write a separate function for the steepest ascent method that calls your gradient and likelihood function. You should write your functions as computationally efficient as possible. Use vectorization as much as possible.

Your output should print the following at each iteration: Iteration number, the value of the log-likelihood and norm of the gradient at initial point of an iteration and intermediate steps within an iteration. It should also print out the halving counts. Once the iterative process is concluded, you will need to print the final estimates for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$**.** For example, your output would look as follows:

Iteration	halving	log-likelihood	||gradient||
1		-20.3273	2.7e0
1	0	NA	NA
1	1	-32.2234	-5.6e0
1	2	-18.6453	1.6e0

2		-18.6453	1.6e0
2	0	-10.9873	7.1e-1

(a) [5 points] Generate 200 data points from a trivariate normal with

$$\text{mean } \boldsymbol{\mu} = (-1, 1, 2)^T \text{ and covariance } \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.7 & 0.7 \\ 0.7 & 1 & 0.7 \\ 0.7 & 0.7 & 1 \end{pmatrix},$$

using the gen() function that was given during the lecture and using **the seed 2024** in the data generation program. Note: I will not accept data generated from other routines or with a different seed. Print the first three rows of your data.

(b) [25 points] Use the data that you generated in part (a) and your steepest ascent function to estimate the parameters in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Start your iterative process with $\boldsymbol{\mu}^{(0)} = (0,0,0)^T$ and $\boldsymbol{\Sigma}^{(\mathbf{0})} = \boldsymbol{I}$ (the identity matrix). Set $maxit = 500, tolerr = 10^{-6}, and\ tolgrad = 10^{-5}$. Print the first two iterations and the last two iterations.