Frank Miller, `frank.miller@liu.se`,
Department of Computer and Information Science, Linköpings University

March 17, 2023

Perform the solutions individually and send your report **until April 3** by email to me. Try to keep this deadline. However, if you have problems with it, there will be a final deadline on September 30 for all assignments. From the following six problems, you can choose five. It is optional, if you want to do the sixth as well.

Problem 1.1

We have independent data $x_1, \ldots, x_n$ from a Cauchy-distribution with unknown location parameter θ and known scale parameter 1. The log likelihood function is

$$-n \log(\pi) - \sum_{i=1}^{n} \log(1 + (x_i - \theta)^2),$$

and it's derivative with respect to θ is

$$\sum_{i=1}^{n} \frac{2(x_i - \theta)}{1 + (x_i - \theta)^2}.$$

Data of size $n = 12$ is given:
$\mathbf{x} = (0.72, 5.50, -2.21, -0.35, -0.67, 0.16, 23.64, 1.00, 1.06, -495.81, -1.98, 37.72)$.

a. Plot the log likelihood function for the given data within an appropriate range; try to cover all local maxima of this function. Plot the derivative in the same range and check visually how often the derivative is equal to 0.

b. Program the Newton-Raphson method. You need the second derivative and you might use the `R`-function `D` to help. Choose suitable starting values (based on your plots) to identify all local maxima of the likelihood function.

c. How would you use your Newton-Raphson function if you want to determine the global maximum and if you do not want to choose the starting value based on a plot (i.e. the algorithm should run automatised)?

Problem 1.2

Let

$$g(x, y) = -x^2 + 10y - 2y^3 + \frac{1}{2}x^2 y.$$

a. Plot the function with a contour plot or a 3-dimensional plot to visualise the function.

b. Compute the gradient analytically. Set the gradient to 0 and solve the equations analytically to identify candidates for maxima, minima, and saddle points.

c. Compute the Hessian matrix analytically. Determine if it is positive, negative, or indefinite in the candidate points (if you want to calculate eigenvalues, you can use software for it). What does this mean for your candidate points identified in b.?

Problem 1.3

Three doses (0.1, 0.3, and 0.9 g) of a drug and placebo (0 g) are tested in a study. A dose-dependent event is recorded afterwards. The data of $n = 10$ subjects is shown in Table 1; x_i is the dose in gram; $\tilde{x}_i$ is the dose with unit changed to milligram; $y_i = 1$ if the event occurred, $y_i = 0$ otherwise.

x_i	in g	0	0	0	0.1	0.1	0.3	0.3	0.9	0.9	0.9
$\tilde{x}_i$	in mg	0	0	0	100	100	300	300	900	900	900
y_i		0	0	1	0	1	1	1	0	1	1

Table 1: Data for Problem 1.3

You should fit a simple logistic regression

$$p(x) = P(Y = 1|x) = \frac{1}{1 + \exp(-\beta_0 - \beta_1 x)}.$$

to the data, i.e. estimate β_0 and β_1. One can show that the log likelihood is

$$g((\beta_0, \beta_1)) = \sum_{i=1}^{n} \left[y_i \log\{(1 + \exp(-\beta_0 - \beta_1 x_i))^{-1}\} + (1 - y_i) \log\{1 - (1 + \exp(-\beta_0 - \beta_1 x_i))^{-1}\} \right]$$

and the gradient is

$$\mathbf{g}'(\mathbf{b}) = \sum_{i=1}^{n} \left\{ y_i - \frac{1}{1 + \exp(-\beta_0 - \beta_1 x_i)} \right\} \begin{pmatrix} 1 \\ x_i \end{pmatrix}.$$

a. Write a program for an ML-estimator for (β_0, β_1) using the steepest ascent method with a step-size halving line search (back-tracking). Problems could turn up later, e.g. when the log is taken from a value close to 0. If this becomes a problem, try to handle this issue.

b. Plot the log-likelihood function (contour plot) for the observations in Table 1. Compute the ML-estimator with your function both using gram data (x_i, y_i) and using milligram data $(\tilde{x}_i, y_i)$. How many iterations were used in each case? Discuss reasons for that result. Note: to check your ML-solution, you might use the function `glm` in R.

c. Choose the quasi-Newton with BFGS or the Newton algorithm. Write your own optimisation program with the chosen algorithm. Run your program for the dataset in Table 1, both using g-data and mg-data. Report the number of iterations used and compare with the results from b.. If you have chosen the Newton algorithm, you need the Hessian matrix; write how you obtained it or if you used sources for it, please cite appropriately.

Problem 1.4

We consider the quadratic two-dimensional function

$$g(\mathbf{x}) = -\frac{1}{2} \mathbf{x}^\top A \mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^2.$$

with a symmetric and positive definite 2×2-matrix A. The function g has a maximum at $(0,0)^\top$ and the gradient is $\mathbf{g}'(\mathbf{x}) = -A\mathbf{x}$. For A, we consider two different matrices with

$$A_1 = \begin{pmatrix} 8 & 1 \\ 1 & 8 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 0 \\ 0 & 100 \end{pmatrix}.$$

a. Program your own function for accelerated steepest ascent with Polyak's momentum with fixed step size α (without backtracking) and fixed momentum parameter β. These two parameters should be options in your function such that you can test different options. Choose a stopping criterion such that you have correct results up to around 6 digits. Your function should report the number of iterations used.

b. Plot a contour plot for the two cases and compute with `R` the eigenvalues of the matrices A_i and the condition number κ. For both the steepest ascent and the accelerated steepest ascent, compute the optimal parameters and the best convergence rate ρ in the two cases.

c. Run the steepest ascent method (i.e. use $\beta = 0$ in your function) using a starting value $\mathbf{x}^{(0)} = (-4, -2)^\top$. Use the optimal value and several other α-values in a grid of 0.01 or 0.02 around the optimal value. Report if the algorithm successfully found the maximum and how many iterations were needed for each parameter value. Is the performance best for the theoretically best α-value?

d. Run the accelerated steepest ascent method with Polyak's momentum. Use the optimal α, β and some values around them and report convergence and number of iterations. How much does the acceleration improve performance.

Problem 2.1

As in Problem 1.3, we consider ML estimation for simple logistic regression

$$p(x) = P(Y = 1|x) = \frac{1}{1 + \exp(-\beta_0 - \beta_1 x)}.$$

a. Program a stochastic gradient ascent algorithm with a fixed step size α and a predefined total number of iterations T for simple logistic regression. Your program should also plot the computed $(\beta_0^{(t)}, \beta_1^{(t)})$ in each iteration t $(t = 0, 1, \ldots, T)$ such that you can monitor the search path.

b. Analyze the dataset `logist.txt` (homepage; first column is x, second column y) with your algorithm using the starting value $(\beta_0^{(0)}, \beta_1^{(0)}) = (0.2, 0.5)$. Choose the total number of iterations T and the step size α. You might need to test different options first to come to a good choice. Explain why you have chosen these values T and α.

Problem 2.2

Consider a linear regression model
$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$
for a design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ of full rank p with $n > p$. The interest is in the least squares solution, i.e. the function
$$g(\mathbf{b}) = \frac{1}{n} \|\mathbf{X}\mathbf{b} - \mathbf{y}\|_2^2$$
should be minimised. Show that g is L-smooth and m-strongly convex. Present expressions for L and m.