

Computational Statistics Computer Lab 2 (Group 7)

Qinyuan Qi(qinqi464)

Satya Sai Naga Jaya Koushik Pilla (satpi345)

2023-11-07

Question 1: Optimisation of a two-dimensional function

Consider the function

$$g(x, y) = -x^2 - x^2y^2 - 2xy + 2x + 2$$

It is desired to determine the point $(x, y), x, y \in [-3, 3]$ where the function is maximized.

- Derive the gradient and the Hessian matrix in dependence of x, y . Produce a contour plot of the function g .
- Write an own algorithm based on the Newton method in order to find a local maximum of g .
- Use different starting values: use the three points $(x, y) = (2, 0), (-1, -2), (0, 1)$ and a fourth point of your choice.
- What would be the advantages and disadvantages when you would run a steepest ascent algorithm instead of the Newton algorithm?

Answer:

Question 2: Optimisation of a two-dimensional function

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; $y_i = 1$ 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
$-y_i-$		0	0	1	0	1	1	1	0	1	1

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

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

where $b = (\beta_0, \beta_1)^T$ and the gradient is

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

Write a function for an ML-estimator for (β_0, β_1) using the steepest ascent method with a step-size reducing line search (back-tracking). For this, you can use and modify the code for the steepest ascent example from the lecture. The function should count the number of function and gradient evaluations.

b. Compute the ML-estimator with the function from a. for the data (x_i, y_i) above. Use a stopping criterion such that you can trust five digits of both parameter estimates for $(\beta_0$ and $\beta_1)$. Use the starting value $(\beta_0, \beta_1) = (-0.2, 1)$. The exact way to use backtracking can be varied. Try two variants and compare number of function and gradient evaluation done until convergence.

c. Use now the function `optim` with both the BFGS and the Nelder-Mead algorithm. Do you obtain the same results compared with b.? Is there any difference in the precision of the result? Compare the number of function and gradient evaluations which are given in the standard output of `optim`.

d. Use the function `glm` in R to obtain an ML-solution and compare it with your results before.

Answer:

Appendix: All code for this report

```
knitr::opts_chunk$set(echo = TRUE)
```