Computer Lab 2
Computational Statistics

Linköpings Universitet, IDA, Statistik

2020/11/11

Kurskod och namn:	732A90 Computational Statistics
Datum:	2020/11/10—2020/11/17 (lab session 11 November 2020)
Delmomentsansvarig:	Krzysztof Bartoszek, Martin Beneš, Filip Ekström, Oriol Garrobé Guilera
Instruktioner:	This computer laboratory is part of the examination for the Computational Statistics course Create a group report, (that is directly presentable, if you are a presenting group), on the solutions to the lab as a **.PDF** file. Be concise and do not include unnecessary printouts and figures produced by the software and not required in the assignments. **All R code should be included as an appendix into your report.** A typical lab report should 2-4 pages of text plus some amount of figures plus appendix with codes. In the report reference **ALL** consulted sources and disclose **ALL** collaborations. The report should be handed in via LISAM (or alternatively in case of problems e–mailed to marbe619@student.liu.se, filip.ekstrom@liu.se, origa255@student.liu.se, or krzysztof.bartoszek@liu.se), by **23:59 17 November 2020** at latest. Notice there is a final deadline of **23:59 31 January 2021** after which no submissions nor corrections will be considered and you will have to redo the missing labs next year. The seminar for this lab will take place **25 November 2020**. The report has to be written in English.

Question 1: Optimizing parameters

Finding the minimum or maximum of a function is usually presented as a goal in itself. Here you are asked to use the function `optim()` to create a procedure to approximate another function, through so–called parabolic interpolation. For this exercise let $f(x)$ be a continuous function on the interval $[0, 1]$ and let $x_0, x_1, x_2 \in [0, 1]$ such that $f(x_1) < f(x_0), f(x_2)$. We will approximate the function $f(x)$ with a function that is piecewise $\tilde{f}(x) = a_0 + a_1 x + a_2 x^2$, i.e. a piecewise quadratic function.

1. Write a function that uses `optim()` and finds values of (a_0, a_1, a_2) for which $\tilde{f}$ interpolates f at user provided points x_0, x_1, x_2. Interpolate means $f(x_0) = \tilde{f}(x_0)$, $f(x_1) = \tilde{f}(x_1)$ and $f(x_2) = \tilde{f}(x_2)$. `optim()` should minimize the squared error, i.e. find (a_0, a_1, a_2) that make $(f(x_0) - \tilde{f}(x_0))^2 + (f(x_1) - \tilde{f}(x_1))^2 + (f(x_2) - \tilde{f}(x_2))^2$ as small as possible.

2. Now construct a function that approximates a function defined on the interval $[0, 1]$. Your function should take as a parameter the number of equal–sized intervals that $[0, 1]$ is to be divided into and the function to approximate. The target function is known at the ends of the interval and also at the mid–point of the interval. Independently on each subinterval you should approximate the target function using the parabolic interpolater implemented in the previous part i.e. use the parabolic interpolater to find a_0, a_1, a_2 for each subinterval.

3. Apply your function from the previous item to $f_1(x) = -x(1-x)$ and $f_2(x) = -x\sin(10\pi x)$. Plot $f_1(\cdot)$, $\tilde{f}_1(\cdot)$ and $f_2(\cdot)$, $\tilde{f}_2(\cdot)$. How did your piecewise–parabolic interpolater fare? Explain what you observe. Take the number of subintervals to be at least 100.

Question 2: Maximizing likelihood

The file `data.RData` contains a sample from normal distribution with some parameters μ, σ. For this question read `?optim` in detail.

1. Load the data to `R` environment.

2. Write down the log-likelihood function for 100 observations and derive maximum likelihood estimators for μ, σ analytically by setting partial derivatives to zero. Use the derived formulae to obtain parameter estimates for the loaded data.

3. Optimize the minus log–likelihood function with initial parameters $\mu = 0$, $\sigma = 1$. Try both Conjugate Gradient method (described in the presentation handout) and BFGS (discussed in the lecture) algorithm with gradient specified and without. Why it is a bad idea to maximize likelihood rather than maximizing log–likelihood?

4. Did the algorithms converge in all cases? What were the optimal values of parameters and how many function and gradient evaluations were required for algorithms to converge? Which settings would you recommend?