Examination Computational Statistics

Linköpings Universitet, IDA, Statistik

Course:	732A90 Computational Statistics
Date:	2023/01/10, 8–13
Teacher:	Krzysztof Bartoszek
Allowed aids:	Printed books, 100 page computer document,
Provided aids:	material in the zip file **exam_material_732A90.zip**
Grades:	A= $[18-20]$ points
	B= $[16-18)$ points
	C= $[14-16)$ points
	D= $[12-14)$ points
	E= $[10-12)$ points
	F= $[0-10)$ points
Instructions:	Provide a detailed report that includes plots, conclusions and interpretations. If you are unable to include a plot in your solution file clearly indicate the section of R code that generates it. Give motivated answers to the questions. If an answer is not motivated, the points are reduced. Provide all necessary codes in an appendix. In a number of questions you are asked to do plots. Make sure that they are informative, have correctly labelled axes, informative axes limits and are correctly described. Points may be deducted for poorly done graphs. Name your solution files as: **[your exam account id]_[own file description].[format]** If you have problems with creating a pdf you may submit your solutions in text files with unambiguous references to graphics and code that are saved in separate files. There are **TWO** assignments (with sub–questions) to solve. Provide a separate solution file for each assignment. Include all R code that was used to obtain your answers in your solution files. Make sure it is clear which code section corresponds to which question. If you also need to provide some hand–written derivations please number each page according to the pattern: Question number . page in question number i.e. Q1.1, Q1.2, Q1.3,. . ., Q2.1, Q2.2, . . ., Q3.1, ...

NOTE: If you fail to do a part on which subsequent question(s) depend on describe (maybe using dummy data, partial code e.t.c.) how you would do them given you had done that part. You *might* be eligible for partial points.

Assignment 1 (10p)

In this assignment you are asked to implement an acceptance–rejection algorithm for the Kolmogorov distribution, i.e., it density function has support on the positive half–axis and equals for some (unknown $x_0 > 0$)

$$f(x) = \begin{cases} 8x \sum_{k=1}^{\infty} (-1)^{k-1} k^2 \exp\left(-2k^2x^2\right) & \text{if } x > x_0, \\ \frac{\sqrt{2\pi}}{x^2} \sum_{k=1}^{\infty} \left(\frac{(2k-1)^2\pi^2}{4x^2} - 1\right) \exp\left(-\frac{(2k-1)^2\pi^2}{8x^2}\right) & \text{if } 0 < x \le x_0. \end{cases}$$

Of course this infinite sum with an unknown x_0 cannot be used in practice, so a finite sum approximation can be considered

$$f(x) \approx \begin{cases} 8x \sum_{k=1}^{k^*} (-1)^{k-1} k^2 \exp\left(-2k^2x^2\right) & \text{if } x > x_0, \\ \frac{\sqrt{2\pi}}{x^2} \sum_{k=1}^{k^*} \left(\frac{(2k-1)^2\pi^2}{4x^2} - 1\right) \exp\left(-\frac{(2k-1)^2\pi^2}{8x^2}\right) & \text{if } 0 < x \le x_0, \end{cases}$$

where $x_0 \approx 1.207$ and $k^* = 15$.
In the assignment a *gamma distribution* will be taken as the proposal density.
The problem here is based on the recent manuscript P. Onorati and L. Brunero, A Random Number Generator for the Kolmogorov Distribution (2022), arXiv:2208.13598. You have the referenced article provided for you during the exam (file `OnoratiLiseon_RNGKolmogorovDist_2022arXiv.pdf`). However, this file is **ONLY FOR YOUR INTEREST AND REFERENCE**, there is **NO NEED** to use any information from it above to what is provided in the exam questions to solve the problem.

Question 1.1 (2p)

1. Discuss what could be the problems with the target density if the value of k^* in the numerical approximation would be too small. How could such problems exhibit themselves in the acceptance–rejection algorithm, when an approximation with too small k^* is taken for the target density?

2. If time would not be an issue, would larger values of k^* always improve the situation? What should be the order of summation?

Question 1.2 (2p)

Implement an acceptance–rejection algorithm for the Kolmogorov distribution with the gamma distribution as the majorizing density. For now, keep the majorizing constant and parameters of the gamma distribution as free parameters that should be passed to your sampling function. You may use `rgamma()` and `runif()`.

Question 1.3 (3p)

Using some optimization method find the majorizing constant and the parameters of the majorizing gamma density. You **may not** use the information provided below concerning the optimal values.

TIP 1: Remember that the majorizing constant should be as small as possible. You might need to consider separate optimization methods for the majorizing constant and parameters of the gamma density. It might very well be that you might just need to randomly sample the state space.
VERSION OF QUESTION WITH REDUCED POINTS: *IF YOU FAIL AT FINDING ANY REASONABLE PARAMETERS*, you can take $M = 1.123$, and $(9.21, 10.96)$ for the gamma distribution. In this situation **YOUR TASK** will be to establish which value is the shape, and which the scale or rate. Here, you can earn a maximum of **1 point** for this sub–question.

Question 1.4 (3p)

Using your implemented acceptance–rejection method simulate a sample from the Kolmogorov distribution. Provide example calls to your code. What is your acceptance rate? Is it close to the 89.04% reported in the article? Plot a histogram of the number of iterations required to obtain a single sampled value. Compare your sample mean and variance with the true mean $\sqrt{\pi/2}\log 2 \approx 0.869$, and true variance $\pi^2/12 \approx 0.068$.
If your acceptance–rejection algorithm failed at generating anything sensible discuss where the problem might be in.

Assignment 2 (10p)

Consider Easom's function

$$f(x,y) = -\cos(x)\cdot\cos(y)\cdot\exp\left(-(x-\pi)^2 - (y-\pi)^2\right).$$

This function has multiple local minima, and a global minimum at (π, π).
YOU MAY ONLY USE THE INFORMATION ON THE LOCATION OF THE GLOBAL MINIMUM WHEN EXPLICITLY PERMITTED!

Question 2.1 (4p)

Implement a simulated annealing algorithm for finding the global minimum of Easom's function. Take into consideration Questions 2.2 and 2.3 when implementing. Provide example calls to your code.
VERSION OF QUESTION WITH REDUCED POINTS: *IF YOU FAIL TO IMPLEMENT HERE YOU MAY USE* `optim()` with `method="SANN"` for the next questions, however this might result in REDUCED points. If you use `optim()` you will most probably need to explore the control options `tmax` and `maxit`. Read `?optim` concerning this. Here, you can earn a maximum of **1 point** for this sub–question, for justifying useful values of `tmax`, `maxit`, and possibly other control values.

Question 2.2 (4p)

Explore how well the simulated annealing algorithm is able to find the global minimum depending on the starting position of the search. Consider when the starting position is close to **(BUT NOT AT)** the global optimum, and some distance away from it. If your implementation completely fails at finding the global optimum, discuss why this might be the case.

Question 2.3 (2p)

Provide a plot that shows the considered points by the your algorithm in its search for the global optimum. Clearly indicate the point (π, π) on your graph.