Examination Computational Statistics

Linköpings Universitet, IDA, Statistik

Course code and name:	732A90 Computational Statistics
Date:	2018/08/27, 8–13
Assisting teacher:	Krzysztof Bartoszek
Allowed aids:	Printed books, 100 page computer document, and material in the zip file **732A90_Examination_ExtraMaterial.zip**
Grades:	A= $[17-20]$ points B= $[14.5-17)$ points C= $[10.5-14.5)$ points D= $[8.5-10.5)$ points E= $[7-8.5)$ points F= $[0-7)$ 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]_[own file description].[format]** 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.

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)

Consider the congruential generator of the form

$$x_{n+1} = (ax_n) \mod m. \tag{1}$$

Your task will be to study its quality for various values of a and m.

Question 1.1 (3p)

Implement a function that generates number in the interval $[0, 1]$ based on Eq. (1). The function should have as parameters a, m, x_0 and $nmax$ (the number of values to generate. Your function should check if $nmax$ is a sensible value. If it is not, warn the user and decide how the function should proceed.

Question 1.2 (4p)

Generate 10000 pseudo–random numbers for each of the pairs of parameters $(a = 69069, m = 2^{32})$, $(a = 630360016, m = 2^{31} - 1)$, $(a = 742938285, m = 2^{31} - 1)$ and $(a = 1226874153, m = 2^{31} - 1)$. For each of the four parameter combinations plot a histogram and compare to the standard uniform distribution (use `runif()` to obtain the values for comparison). Use `R`'s `ks.test()` with `y="punif"` to formally test if your generated samples are Unif$[0, 1]$. Report on the tests' results.

Question 1.3 (3p)

Implement a pseudo–random number generators from the normal distribution. Your generator has to be based on your previously implemented congruential generator. Generate 10000 normal values for each of the previous a and m combinations. Plot histograms of both your numbers and true normal numbers (use `rnorm()` to obtain the values for comparison). Use `R`'s `ks.test()` with `y="pnorm"` to formally test if your generated samples are standard normal. Report on the tests' results. Can you draw any conclusions which a and m values are better or worse?

Assignment 2 (10p)

Finding the minimum or maximum of a function is usually presented as a goal in itself. Here we will 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.

Question 2.1 (3p)

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.

Question 2.2 (5p)

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. The target function is known at the ends of the interval and also at the mid–point of the interval. How you provide the values of the target function is up to you. You may hard–code the target function inside your approximation procedure (remember that you will have to provide then separate code for each different function you approximate), you may pass a matrix with the values of the function at the "known" points or you may do this in a dynamical fashion by passing the target function as a parameter. Independently on each subinterval you should approximate the target function using the parabolic interpolater implemented in Question 2.1, i.e. use the parabolic interpolater to find a_0, a_1, a_2 for each subinterval.
HINT: one way is that your function returns a matrix with five columns. Each row corresponds to a subinterval. The first two columns are the intervals start and end and the last three are the values of a_0, a_1, a_2.

Question 2.3 (2p)

Apply your function from Question 2.2 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.