Examination Computational Statistics

Linköpings Universitet, IDA, Statistik

Course code and name:	732A90 Computational Statistics
Date:	2018/03/19, 8–19
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)

A random variable has a Gamma distribution with parameters α (shape) and β (rate), denoted Gamma(α, β), if its density equals

$$f(x) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)},$$

where $\Gamma(\alpha)$ is the Euler Gamma function defined as

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} \mathrm{d}x$$

and available in `R` as `gamma(`α`)`. The aim of the assignment is to sample from the Gamma(α, β) distribution by means of an acceptance–rejection algorithm using an exponential with rate λ distribution, denoted $\exp(\lambda)$, as the proposal distribution. Recall the exponential with rate $\lambda > 0$ distribution has density equalling

$$g(x) = \lambda e^{-\lambda x}.$$

Question 1.1 (2p)

Consider the Gamma$(2, \beta)$ density, with $\beta \geq 2$. If $\exp(\lambda)$'s density is the proposal density, then find some admissible values of λ and M. Provide your derivations. If you find it too difficult to show for all $\beta \geq 2$, then you may choose a particular value of β (1 point will be deducted).
TIP: Remember that for $\gamma \geq 1$ and $x \geq 0$ it holds $x \leq e^{\gamma x}$.
(**NOT PART OF THE EXAM**, plot the functions x and $e^{\gamma x}$ for some value of γ if you need.)

Question 1.2 (5p)

Implement an acceptance–rejection algorithm for sampling from the Gamma$(2, \beta)$ distribution using $\exp(\lambda)$ as the proposal distribution. You should implement your sampler as a function that takes β as a parameter and then calculates an appropriate M and λ (based on your calculations in the previous question). If you did not succeed with the derivations in Question 1.1 implement your acceptance–rejection algorithm with the following parameters $\beta = 2, M = 4, \lambda = 1$ (1 point will be deducted). Your sampler should be still written as a function.

Generate a sample of size 10000 using your implemented sampler for a number of values of β of your own choice (or only $\beta = 2$ if you failed to do Question 1.1). Compare (graphically and using relevant statistics) your simulated samples with the true Gamma distributions. Provide appropriate plots and comments.
NOTE: You are allowed to use `R`'s built–in `rexp()` function.

Question 1.3 (3p)

Using the your implemented sampler from Question 1.2 estimate the following integral

$$\int_0^\infty x^4 e^{-2x} \mathrm{d}x.$$

As a check compare the value of the integral if you would use `R`'s gamma distribution sampler. If you failed to do Question 1.2 you can of course only estimate the integral using `rgamma()`, do this as you *might* be eligible for partial points.

Assignment 2 (10p)

Your task is to minimize the function

$$f(x) = -x \cdot \sin(10\pi x) + 1, \quad \text{for } x \in [-1, 2]$$

using a genetic algorithm. As a first step you should always plot the function to minimize.

Question 2.1 (2p)

An initial step when implementing a genetic algorithm is the decision on how individuals in the population will be represented. Often one wants to represent them as binary strings as it makes the implementation of crossover, mutation e.t.c. operators easy.

Each individual is to be represented as a binary vector of length $m = 15$. You may use the following `R` code to transform a binary vector to the integer it represents

```
u<-Reduce(function(s,r) {s*2+r}, v)
```

for example

```
> v<-c(1,0,1)
> Reduce(function(s,r) {s*2+r}, v)
[1] 5
> v<-c(1,0,1,0)
> Reduce(function(s,r) {s*2+r}, v)
[1] 10
```

Notice that `v[1]` is the highest bit. Then, to transform such a number into the interval $[a, b]$ one uses the transformation (in our case $a = -1$, $b = 2$, $m = 15$)

$$z = a + u\frac{b-a}{2^m - 1},$$

i.e. $z \in [a, b]$. Implement a function that takes as its input a chromosome (i.e. a binary vector of length $m = 15$) and returns the value of $f(\cdot)$ associated with it.

If you are unable to implement this coding of the state space you may use another one of your own choice in order to do the next questions. You *might* be eligible for partial points.

GENERAL COMMENT The parameter m corresponds to the number of bits used to store the argument of $f(\cdot)$ and hence is used to control the numerical accuracy of the optimizer.

Question 2.2 (3p)

The next step is to implement a selection procedure in order to choose the pool of individuals that will be allowed to contribute to the next generation. You are to implement a function that does selection according to a procedure called *tournament selection*.

The way it works is that you draw two groups of size two or three (you choose this) individuals from the population. Then in each group you choose the best individual from each group and these two individuals are taken for reproduction in the next step. To keep a constant population size the two losing individuals are removed from the population (see also the printout from Wikipedia 732A90_CS_Examination_TournamentSelectionWikipedia.pdf)

Sometimes one varies this tournament selection and the best individual is chosen with a certain probability lesser than 1. Can you provide a motivation for this?

If you are unable to implement this type of selection procedure you may use another one of your own choice in order to do the next questions. You *might* be eligible for partial points

Question 2.3 (5p)

Implement all the other necessary components of the genetic algorithm and using it find the minimum of the function $f(x)$. Take a population size of 55. It is recommended that you first try your code on a smaller population (say 6) and only after it runs without error, run it on the population of size 55. Follow your population for 100 generations. Provide example calls to your code.

Visualize the population's behaviour with the number of generations for mutation probability 0.0077 and 0.5 and provide comments. At which generation was the best value found? Was the minimum found? Can you explain the observed behaviour, especially when taking into account the mutation probability?