TMA4300 Computer Intensive Statistical Methods
Exercise 1, Spring 2016

Note: Solutions of Problems C and D must be handed in no later than **29 February 2016, 24:00**. All answers including derivations, computer code and graphics (preferably in one pdf document) should be submitted to

Xin Luo (xin.luo@math.ntnu.no).

Getting started: *The aim of this exercise is to make R functions that generate random numbers from a number of different probability distributions using the methods discussed in the lectures. Therefore, the R function* `runif` *can be used to generate random numbers that are uniformly distributed between* 0 *and* 1 *(!), but no other built-in random number functions in R (like* `rexp`*,* `rgamma`*,* `rbeta` *and* `rnorm`*) should be used.*

Important: *For* ***each function or code chunk*** *you write in this exercise you are supposed to check that it is working properly. You may compare properties of the random numbers generated with known properties of the theoretical distribution. For example, you may compute the empirical mean (*`mean(x)`*) and variance (*`var(x)`*) of the vector of generated samples and compare with the known theoretical moments, and make histograms of the generated numbers and compare with the known theoretical density function.*

Note: *Your code will run much faster if you, whenever possible, do operations on vectors instead of using for loops. For example, "*`x = log(runif(n))`*" runs much faster than "*`u = runif(n); for (i in 1:length(u)) x[i]=log(u[i])`*".*

Note: *To avoid numerical problems causing underflows or overflows it might be sensible to do certain computations on log-scale and then re-transform the final result.*

Problem A: Stochastic simulation by the probability integral transform

1. Write an R function that generates samples from an exponential distribution with (rate) parameter λ. Let the function take two arguments as input, the (rate) parameter of the exponential distribution, λ, and the number of samples to generate, n, and let it return a vector with the generated random numbers.

2. Consider the probability density function

$$f(x) = \frac{c\mathrm{e}^{\alpha x}}{(1+\mathrm{e}^{\alpha x})^2}, \quad -\infty < x < \infty, \alpha > 0$$

   where c is the normalising constant.

   (a) Find the value of c by integrating f from minus infinity to infinity.

   (b) Find a formula for the cumulative distribution function, F, and the inverse of F.

   (c) Write an R function that generates samples from f. As in 1, let the function take two input arguments, α and n, and let it return a vector with the generated random numbers. Remember also to check, as discussed above, that your function is working properly.

3. Consider the probability density function

$$g(x) = \begin{cases} cx^{\alpha-1}, & 0 < x < 1, \\ c\mathrm{e}^{-x}, & 1 \leq x, \\ 0, & \text{otherwise}, \end{cases}$$

where $\alpha \in (0, 1)$ and c is the normalising constant.

(a) Find the cumulative distribution function and the inverse of the cumulative distribution function.

(b) Write an R function that generates samples from g. Again check your implementation as discussed above.

Problem B: Rejection sampling and importance sampling

Consider the data of Rao (1973, 368f)[1] on a certain recombination rate in genetics. Here, 197 counts are classified into four categories and assumed to be multinomial distributed. Table 1 shows the data $\boldsymbol{y} = (y_1, y_2, y_3, y_4)$:

Cell count	Probability
$y_1 = 125$	$\frac{1}{2} + \frac{\theta}{4}$
$y_2 = 18$	$\frac{1-\theta}{4}$
$y_3 = 20$	$\frac{1-\theta}{4}$
$y_4 = 34$	$\frac{\theta}{4}$

Table 1: Genetic linkage data.

The multinomial mass function is given by $f(\boldsymbol{y}|\theta) \propto (2+\theta)^{y_1}(1-\theta)^{y_2+y_3}\theta^{y_4}$. Using a uniform prior on $(0, 1)$, which is equivalent to a Beta$(1, 1)$ prior, for θ the observed posterior density is:

$$f(\theta|\boldsymbol{y}) \propto (2+\theta)^{y_1}(1-\theta)^{y_2+y_3}\theta^{y_4}.$$

We are interested in the posterior mean $E(\theta|\boldsymbol{y})$.

1. Construct a rejection sampling algorithm to simulate from $f(\theta|\boldsymbol{y})$ using a $\mathcal{U}(0, 1)$ density as the proposal density.

2. Estimate the posterior mean of θ by Monte-Carlo integration using $M = 10000$ samples from $f(\theta|\boldsymbol{y})$. Draw a histogram of the samples and compare it with the theoretical density distribution. Mark also the estimated posterior mean.

3. How many random numbers does your sampling algorithm need to generate on average in order to obtain one sample of $f(\theta|\boldsymbol{y})$. Derive your answer practically using your sampler and compare it with the theoretical result computed numerically.

4. In part (2) we obtained samples of the posterior distribution assuming a uniform prior, i.e. Beta$(1, 1)$ prior, for θ. Suppose we assume a Beta$(1, 5)$ prior instead of the previous Beta$(1, 1)$. Use the importance sampling weights to estimate the posterior mean under the new prior based on the samples from part (2). Comment your result.

[1] Rao, C. R. (1973). *Linear Statistical Inference and its Applications*, 2nd edn, Wiley, New York.

Problem C: Stochastic simulation by bivariate techniques and rejection sampling

1. Write an R function that uses the Box-Muller algorithm to generate a vector of n independent samples from the standard normal distribution. Check that your function is working properly by comparing to known quantities from the theoretical distribution.

2. Consider a gamma distribution with parameters $\alpha \in (0, 1)$ and $\beta = 1$, i.e.

$$f(x) = \begin{cases} \frac{1}{\Gamma(\alpha)} x^{\alpha-1} \mathrm{e}^{-x}, & 0 < x, \\ 0, & \text{otherwise.} \end{cases}$$

   Rejection sampling can be used to generate samples from this distribution by proposing samples from

$$g(x) = \begin{cases} cx^{\alpha-1}, & 0 < x < 1, \\ c\mathrm{e}^{-x}, & 1 \leq x, \\ 0, & \text{otherwise,} \end{cases}$$

   where c is the normalising constant.

   Note: *The distribution g is the one you considered in Problem A.3.*

   (a) Find an expression for the acceptance probability in the rejection sampling algorithm.

   (b) Write an R function that generates a vector of n independent samples from f.

3. Consider a gamma distribution with parameters $\alpha > 1$ and $\beta = 1$, i.e.

$$f(x) = \begin{cases} \frac{1}{\Gamma(\alpha)} x^{\alpha-1} \mathrm{e}^{-x}, & 0 < x, \\ 0, & \text{otherwise,} \end{cases}$$

   We will use the ratio of uniforms method to simulate from this distribution. Define, as in the lectures,

$$C_f = \left\{ (x_1, x_2) : 0 \leq x_1 \leq \sqrt{f^\star\left(\frac{x_2}{x_1}\right)} \right\} \quad \text{where} \quad f^\star(x) = \begin{cases} x^{\alpha-1} \mathrm{e}^{-x}, & 0 < x, \\ 0, & \text{otherwise,} \end{cases}$$

   and

$$a = \sqrt{\sup_x f^\star(x)}\ ,\ b_+ = \sqrt{\sup_{x \geq 0} (x^2 f^\star(x))} \text{ and } b_- = -\sqrt{\sup_{x \leq 0} (x^2 f^\star(x))},$$

   so that $C_f \subset [0, a] \times [b_-, b_+]$.

   (a) Find the values of a, b_- and b_+.

   (b) Write an R function that generates a vector of n independent samples from f. Use the function to check how long the algorithm needs to generate $n = 1000$ realisations depending on the value of $\alpha \in (1, 2000]$. Generate a plot with values of α on the x-axis and the time used on the y-axis. Interpret the result.
   Caution: You need to implement the algorithm on log-scale otherwise you will get `NA`s already for α around 30.

4. Write an R function that generates a vector of n independent samples from a gamma distribution with parameters α and β. Note that the function should work for any values $\alpha > 0$ and $\beta > 0$. *Hint: For the gamma distribution β is an (inverse) scale parameter.*

Problem D: Multivariate distributions

1. Write an R function that generates one realisation from a d-variate normal distribution with given mean vector μ and covariance matrix Σ.
   Check that your function is working properly by comparing the true mean and covariance matrix to the estimated mean and empirical covariance matrix.

2. Let $x = (x_1, \ldots, x_K)$ be a vector of stochastic variables where $x_k \in [0, 1]$ for $k = 1, \ldots, K$ and $\sum_{k=1}^{K} x_k = 1$. The vector x is said to have a Dirichlet distribution with parameter vector $\alpha = (\alpha_1, \ldots, \alpha_K)$ if the density for $(x_1, \ldots, x_{K-1})$ is given by

$$f(x_1, \ldots, x_{K-1}) = \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\Gamma(\alpha_1) \cdot \ldots \Gamma(\alpha_K)} \cdot x_1^{\alpha_1 - 1} \cdot \ldots \cdot x_{K-1}^{\alpha_{K-1} - 1} \cdot \left(1 - \sum_{k=1}^{K-1} x_k\right)^{\alpha_K - 1},$$

   for $x_1, \ldots, x_{K-1} > 0$ and $\sum_{k=1}^{K-1} x_k < 1$.

   (a) Assume $z_k \sim \text{gamma}(\alpha_k, 1)$ for $k = 1, \ldots, K$ independently, and define $x_k = z_k/(z_1 + \ldots, z_K)$ for $k = 1, \ldots, K$. Show that then $x = (x_1, \ldots, x_K)$ has a Dirichlet distribution with parameter vector $\alpha = (\alpha_1, \ldots, \alpha_K)$. *Hint: Use the change-of-variables formula and transform the original variables $(z_1, \ldots, z_K)$ to $(x_1, \ldots, x_{K-1}, v)$, with $v = z_1 + \ldots + z_K$).*

   (b) Write an R function that generates one realisation from a Dirichlet distribution with parameter vector $\alpha = (\alpha_1, \ldots, \alpha_K)$.

Oral presentations

Date	Exercise	Team
15.02.2016	1: Problem A1 and A2	Kristian Aga, Moritz Klaus Bogs
	1: Problem A3	Tomasz Kusmierczyk, Eliezer de Souza da Silva
	1: Problem B	John Darkwah, Audun Sektnan
22.02.2016	1: Problem C1 and C2	Fredrik Lundquist, Ole Bernhard Forberg
	1: Problem C3 and C4	Yingzi Jin, Himanshu Srivastav
	1: Problem D1 and D2	Eivind Berg Fosse, Espen Høegh Sørum
04.04.2016	2	Eyvind Thommesen
		Negar Olfati, Abdolreza Sabzi Shahrebabaki
		Arnaud Vacogne, Truls Brubak
11.04.2016	2	Sandra Sæther, Dechasa Obsi Gudeta
		Martina Hall, Jorid Ødegrd
		Øyvind Øksnes Dalheim, Anna Swider
25.04.2016	3	Gunhild Elisabet Berget, Ingeborg Gullikstad Hem
		Andreas Malmgård, Haris Fawad
		Filip Emil Schjerven, Rafael Schwarzenegger
		Inger Ellingsen