TMA4300 Computer Intensive Statistical Methods
Exercise 1, Spring 2023

Note: Solutions must be handed in no later than **Sunday February 12th, 11:59 pm**. All answers including derivations, computer code and graphics (all in one pdf document!) should be submitted in Blackboard as specified in the course home page.

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. You should store all the functions you make in this exercise, as you may need to use them in Exercises 2 and 3.*

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 and bivariate techniques

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

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

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

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

   (b) Write an R function that generates samples from g. Check your implementation as discussed in the introduction.

3. 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 a normalising constant.

   (a) Find the value of c.

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

   (c) Write an R function that generates samples from f. 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.

4. 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.

5. Write an R function that generates realisations 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.

Problem B: The gamma distribution

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

$$f(x) = \begin{cases} \frac{1}{\Gamma(\alpha)} x^{\alpha-1}\mathrm{e}^{-x}, & 0 < x, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

   Rejection sampling can be used to generate samples from this distribution by proposing samples from the distribution you considered in Problem A.2.

   (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.

2. Consider a gamma distribution with parameters $\alpha > 1$ and $\beta = 1$. Recall that the density function is given in (2).We will use the ratio of uniforms method to simulate from this distribution. Define, as in the lectures,

$$C_f = \left\{ (x_1, x_2) : 0 \le x_1 \le \sqrt{f^\star\left(\frac{x_2}{x_1}\right)} \right\} \quad \text{where} \quad f^\star(x) = \begin{cases} x^{\alpha-1}\mathrm{e}^{-x}, & 0 < x, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

   and

$$a = \sqrt{\sup_x f^\star(x)}\ ,\ b_+ = \sqrt{\sup_{x\ge 0}(x^2 f^\star(x))} \text{ and } b_- = -\sqrt{\sup_{x\le 0}(x^2 f^\star(x))}, \tag{4}$$

   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 many tries 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 number of tries 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.

3. 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 $\beta > 0$, including $\alpha = 1$. *Hint: For the gamma distribution β is an (inverse) scale parameter.*

4. Let $x \sim \text{Gamma}(\alpha, 1)$ and $y \sim \text{Gamma}(\beta, 1)$ independently, and let $z = x/(x + y)$.

   (a) Show that $z \sim \text{beta}(\alpha, \beta)$, i.e. that the density of z is

   $$f(z) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} z^{\alpha-1}(1 - z)^{\beta-1} \ , \ \ z \in [0, 1].$$

   (b) Write an R function that generates a vector of n independent samples from a beta distribution with parameters α and β.

Problem C: Monte Carlo integration and variance reduction

The aim of this exercise is to learn about variance reduction techniques in Monte Carlo integration by using it in one (simple) example. Assume we want to use Monte Carlo integration to find $\theta = \text{Prob}(X > 4)$ when $X \sim \text{N}(0, 1)$.

1. Estimate θ by Monte Carlo integration using $n = 100000$ samples from the standard normal distribution. Compute also a 95% confidence interval for θ based on these n samples.

2. Write an R function that uses importance sampling to estimate θ via sampling from

   $$g(x) = \begin{cases} cx \exp\left\{-\frac{1}{2}x^2\right\} & , \ x > 4, \\ 0 & , \ \text{otherwise}, \end{cases}$$

   where c is a normalising constant. Note that you can use inversion sample to simulate from $g(x)$.

   Compute also a 95% confidence interval for θ based on $n = 100000$ samples from the importance sampling algorithm. Compare the precisions of the two estimators. How many samples, n, would you have needed in C.1 to get the same precision as you obtained with $n = 100000$ here.

3. Combine the importance sampling in the previous item with the use of antithetic variates. More specifically,

   (a) make a modified version of the R function written in Problem C.2 so that it produces n **pairs** of antithetic variates from the specified distribution by using u and $1-u$, respectively, as input to $F^{-1}(\cdot)$.

   (b) Use $n = 50000$ pairs of samples from $g(x)$ to estimate θ. Why should you use $n = 50000$ here to get a fair comparison with what you have done above? Compute also a 95% confidence interval for θ based on these $n = 50000$ samples.

Problem D: Rejection sampling and importance sampling

Consider the data of Rao (1973)[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:

$$f(\theta|\boldsymbol{y}) \propto (2+\theta)^{y_1}(1-\theta)^{y_2+y_3}\theta^{y_4} \quad \text{for} \quad \theta \in (0,1).$$

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.

   Check the value using numerical integration. *Hint: You can use the function* `integrate()` *in R.*

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 a 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.

Oral presentations

Date	Problem	Team
15.02	Ex 1: Problem A 1-3	Group TBA
15.02	Ex1: Problem A 4-5	Group TBA
15.02	Ex1: Problem B 1-2	Group TBA
15.02	Ex1: Problem B 3-5	Group TBA
15.02	Ex1: Problem C	Group TBA
15.02	Ex1: Problem D	Group TBA