Exercise 1

TMA4300 Computer Intensive Statistical Models

Mads Adrian Simonsen, William Scott Grundeland Olsen

28 januar, 2021

Problem A: Stochastic simulation by the probability integral transform and bivariate techniques

1.

Let $X \sim \text{Exp}(\lambda)$, with the cdf

$$F_X(x) = 1 - e^{-\lambda x}, \quad x \geq 0.$$

Then the random variable $Y := F_X(X)$ has a $\text{Uniform}(0, 1)$ distribution. The probability integral transform becomes

$$Y = 1 - e^{-\lambda X} \Leftrightarrow X = -\frac{1}{\lambda}\log(1 - Y). \tag{1}$$

Thus, we sample Y from `runif()` and transform it using (1), to sample from the exponential distribution. Figure 1 shows one million samples drawn from the `generate_from_exp()` function defined in the code chunk below.

```
set.seed(123)

generate_from_exp <- function(n, rate = 1) {
  Y <- runif(n)
  X <- -(1 / rate) * log(1 - Y)
  X
}

# sample
n <- 1000000  # One million samples
lambda <- 4.32
x <- generate_from_exp(n, rate = lambda)

# plot
hist(x,
  breaks      = 80,
  probability = TRUE,
  xlim        = c(0, 2)
)
curve(dexp(x, rate = lambda),
  add = TRUE,
  lwd = 2,
  col = "red"
)
```

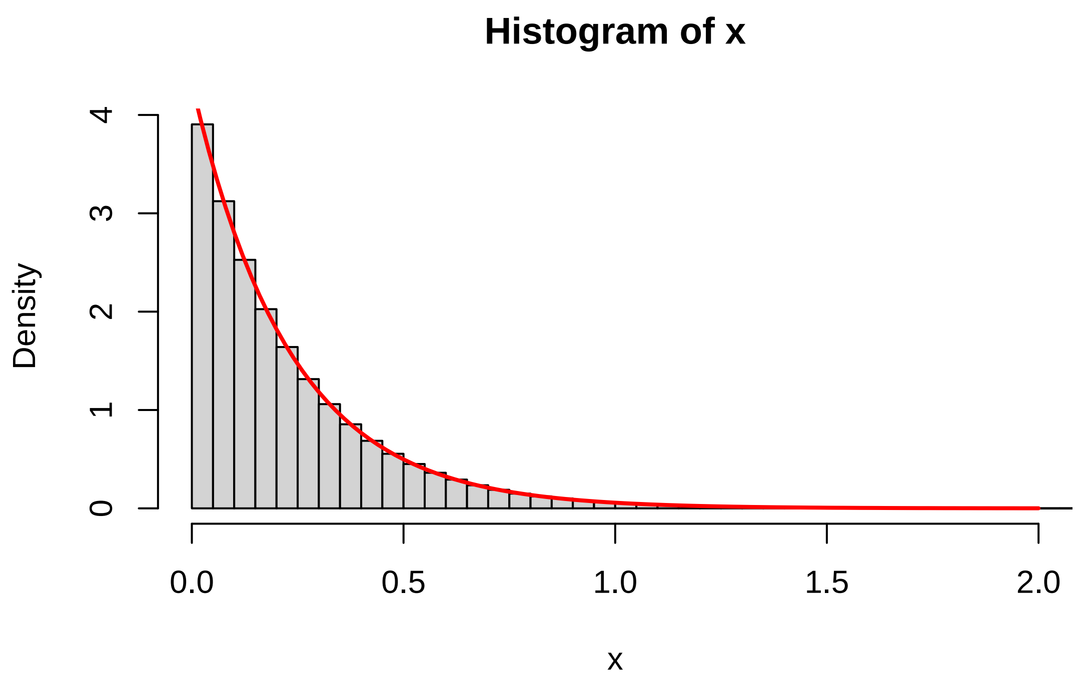

Figure 1: Normalized histogram of one million samples drawn from the exponential distribution, together with the theoretical pdf, with $\lambda = 4.32$.

2.

(a)

(b)

3.

(a)

(b)

(c)

4.

5

Problem B: The gamma distribution

1.

(a)

Let $f(x)$ be the target distribution we wish to sample from, and let $g(x)$ be the proposal distribution. For the rejection sampling algorithm, we require that

$$f(x) \leq c \cdot g(x), \quad \forall x \in \mathbb{R}, \tag{2}$$

for some constant $c > 0$. Let X and U be independent samples where $X \sim g(x)$ and $U \sim \text{Uniform}(0, 1)$. Then the acceptance probability is

$$\begin{aligned}
\Pr\left(U \leq \frac{f(X)}{c \cdot g(X)}\right) &= \int_{-\infty}^{\infty} \int_{0}^{f(x)/(c\ g(x))} f_{X,U}(x, u)\, du\, dx \\
&= \int_{-\infty}^{\infty} \int_{0}^{f(x)/(c\ g(x))} g(x) \cdot 1\, du\, dx \\
&= \int_{-\infty}^{\infty} \frac{f(x)}{c\ g(x)} g(x)\, dx \\
&= \frac{1}{c} \int_{-\infty}^{\infty} f(x)\, dx \\
&= \frac{1}{c}
\end{aligned}$$

We wish to sample from $\text{Gamma}(\alpha, \beta = 1)$, using the proposal distribution $g(x)$ given in `eqref{??????}`. We want to choose c such that the acceptance probability is maximized while (2) is satisfied. We must check three cases. The trivial case when $x \leq 0$, we have $f(x) = g(x) = 0$ so (2) is satisfied for all c. When $0 < x < 1$ we have

$$\begin{aligned}
f(x) &\leq c\, g(x) \\
\frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x} &\leq c\, \frac{1}{\alpha^{-1} + e^{-1}} x^{\alpha-1} \\
c &\geq \left(\frac{1}{\alpha} + \frac{1}{e}\right) \frac{1}{\Gamma(\alpha)} e^{-x} \\
c &\geq \left(\frac{1}{\alpha} + \frac{1}{e}\right) \frac{1}{\Gamma(\alpha)}.
\end{aligned}$$

The last case, when $x \geq 1$, we have

$$\begin{aligned}
f(x) &\leq c\, g(x) \\
\frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x} &\leq c\, \frac{1}{\alpha^{-1} + e^{-1}} e^{-x} \\
c &\geq \left(\frac{1}{\alpha} + \frac{1}{e}\right) \frac{1}{\Gamma(\alpha)} x^{\alpha-1} \\
c &\geq \left(\frac{1}{\alpha} + \frac{1}{e}\right) \frac{1}{\Gamma(\alpha)}.
\end{aligned}$$

That is, we choose $c := (\alpha^{-1} + e^{-1})/\Gamma(\alpha)$, such that the acceptance probability becomes

$$\Pr\left(U \leq \frac{f(X)}{c \cdot g(X)}\right) = \frac{1}{c} = \frac{\Gamma(\alpha)}{\alpha^{-1} + e^{-1}}, \quad \alpha \in (0, 1).$$

(b)

```
set.seed(137)

g_proposal <- function(x, alpha = 0.5) {
  k <- 1 / (1 / alpha + 1 / exp(1))
  y <- ifelse(
        test = x <= 0,
```

```r
         yes  = 0,
         no   = ifelse(
                  test = x < 1,
                  yes  = k * x^(alpha - 1),
                  no   = k * exp(-x)
                )
       )
  return(y)
}


generate_from_gamma <- function(n, shape = 0.5) {
  c <- gamma(shape) / (1 / shape + 1 / exp(1))
  x <- vector(mode = "numeric", length = n)
  for (i in 1:n) {
    repeat{
      x[i] <- generate_from_g(1, shape)  # draw from g !!!!CURRENTLY MISSING THIS FUNCTION
      u <- runif(1)  # draw from U(0, 1)
      f <- dgamma(x[i], shape = shape)
      g <- g_proposal(x[i], alpha = shape)
      alpha <- (1 / c) * (f / g)
      if (u <= alpha) {
        break
      }
    }
  }
  return(x)
}

# n <- 1000000
# alpha <- 0.9
# x <- generate_from_gamma(n, shape = alpha)
# hist(x,
#   breaks      = 80,
#   probability = TRUE,
#   xlim        = c(0, 6)
# )
# curve(dgamma(x, shape = alpha),
#   add = TRUE,
#   lwd = 2,
#   col = "red"
# )
```

2.

(a)

(b)

3.

(a)

(b)

4.

5.

(a)

(b)

Problem C: Monte Carlo integration and variance reduction

1.

2.

3.

(a)

(b)

Problem D: Rejection sampling and importance sampling

1.

2.

3.

4.