Exercise 1

TMA4300 Computer Intensive Statistical Models

Mads Adrian Simonsen, William Scott Grundeland Olsen

07 februar, 2021

```
#remove this-------------------------------------------------------------------------------
generate_from_exp <- function(n, rate = 1) {
  Y <- runif(n)
  X <- -(1 / rate) * log(Y)
  return(X)
}
std_normal <- function(n) {
  X1 <- pi * runif(n)   # n samples from Uniform(0, pi)
  X2 <- generate_from_exp(n, 1/2)   # n samples from Exponential(1/2)
  Z <- X2^(1/2) * cos(X1)   # Z ~ Normal(0, 1)
  return(Z)
}
#-----------------------------------------------------------------------------------------
```

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
```

```r
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"
)
```

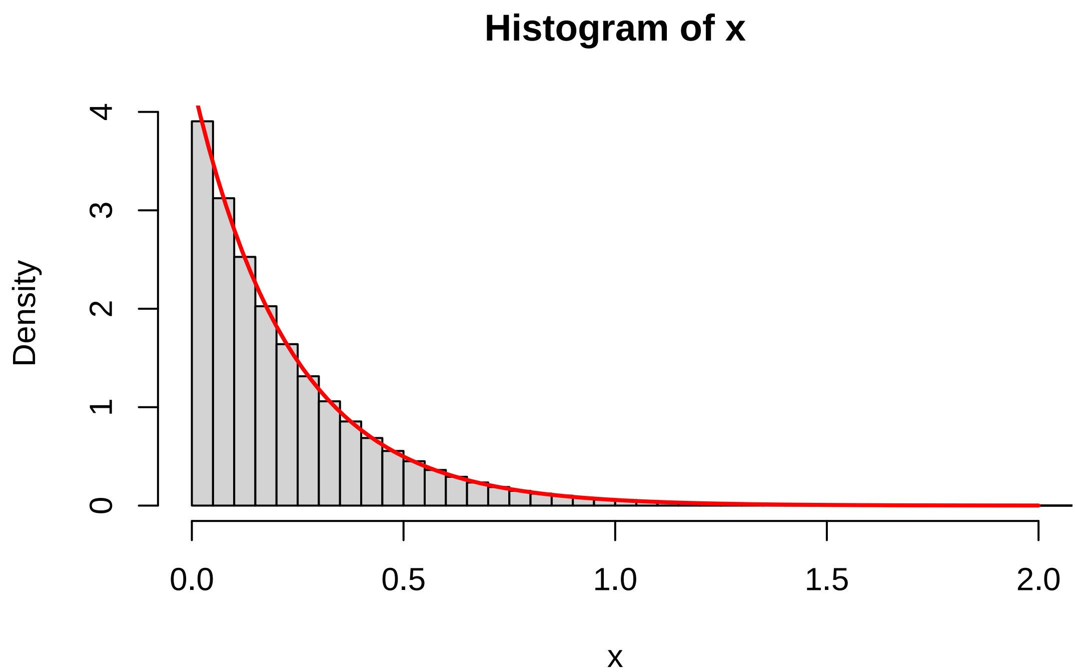

Figure 1: Normalized histogram of one million samples drawn from the exponential distribution, together with the theoretical pdf, with $\lambda = 4.32$.

2.

(a)

(b)

3.

(a)

(b)

(c)

4.

5

Problem B: The gamma distribution

1.

(a)

(b)

2.

(a)

(b)

3.

(a)

(b)

4.

5.

(a)

(b)

Problem C: Monte Carlo integration and variance reduction

1.

Let $X \sim \mathrm{N}(0,1)$, and $\theta = \Pr(X > 4) \approx 3.1671242 \times 10^{-5}$. Let also $h(x) = I(x > 4)$, where $I(\cdot)$ is the indicator function. Then

$$\mathrm{E}[h(X)] = \int_{-\infty}^{\infty} h(x) f_X(x)\, dx = \int_{-\infty}^{\infty} I(x > 4) f_X(x)\, dx = \Pr(X > 4) = \theta.$$

Let $X_1, \dots X_n \sim \mathrm{N}(0,1)$ be a sample. Then the simple Monte Carlo estimator of θ is

$$\hat{\theta}_{\mathrm{MC}} = \frac{1}{n}\sum_{i=1}^{n} h(X_i),$$

with expectation

$$\mathrm{E}\left[\hat{\theta}_{\mathrm{MC}}\right] = \frac{1}{n}\sum_{i=1}^{n} \mathrm{E}\left[h(X_i)\right] = \frac{1}{n}\sum_{i=1}^{n} \theta = \theta,$$

and sampling variance

$$\widehat{\text{Var}}\left[\hat{\theta}_{\text{MC}}\right] = \frac{1}{n^2}\sum_{i=1}^{n}\widehat{\text{Var}}\left[h(X_i)\right] = \frac{1}{n}\widehat{\text{Var}}[h(X)] = \frac{1}{n(n-1)}\sum_{i=1}^{n}\left(h(X_i) - \hat{\theta}_{\text{MC}}\right)^2.$$

Then the statistic

$$T = \frac{\hat{\theta}_{\text{MC}} - \theta}{\sqrt{\widehat{\text{Var}}\left[\hat{\theta}_{\text{MC}}\right]}} \sim \text{t}_{n-1},$$

and $t_{\alpha/2,\,n-1} = F_T^{-1}(1-\alpha/2)$, where $F_T^{-1}(\cdot)$ is the quantile function of the t_{n-1} distribution.

```r
set.seed(321)
theta <- pnorm(4, lower.tail = FALSE)
n <- 100000
x <- std_normal(n)
h <- function(x) {
  1 * (x > 4)
}

hh <- h(x)                  # I(X > 4) vector of ones and zeros
theta_MC <- mean(hh)        # Monte Carlo estimate of Pr(X > 4)
sample_var_MC <- var(hh)  # Sampling variance

t <- qt(0.05/2, df = n - 1, lower.tail = FALSE)  # quantile with 5% significance level
ci_MC <- theta_MC + t * sqrt(sample_var_MC / n) * c(-1, 1)  # Confidence Interval

# Result
list(
  theta_MC      = theta_MC,
  sample_var_MC = sample_var_MC,
  confint       = ci_MC,
  error         = abs(theta_MC - theta)
)
```

```
## $theta_MC
## [1] 6e-05
##
## $sample_var_MC
## [1] 5.9997e-05
##
## $confint
## [1] 0.0000119915 0.0001080085
##
## $error
## [1] 2.832876e-05
```

2.

We will sample from the proposal distribution

$$g_X(x) = \begin{cases} cxe^{-\frac{1}{2}x^2}, & x > 4 \\ 0, & \text{otherwise.} \end{cases}$$

but first we must find the normalizing constant c.

$$
\begin{aligned}
c &= \left(\int_4^\infty x e^{-\frac{1}{2}x^2} \, dx \right)^{-1} = \left(\int_{\frac{1}{2}4^2}^\infty e^{-u} \, du \right)^{-1} = \left(e^{-\frac{1}{2}4^2} - 0 \right)^{-1} = e^{\frac{1}{2}4^2}, \\
\Rightarrow g_X(x) &= \begin{cases} x e^{-\frac{1}{2}(x^2 - 4^2)}, & x > 4, \\ 0, & \text{otherwise.} \end{cases}
\end{aligned}
$$

We can easily sample from the proposal distribution using inversion sampling. The cdf for $x > 4$ is found by integrating.

$$
G_X(x) = \int_4^x y e^{-\frac{1}{2}(y^2 - 4^2)} \, dy = \int_0^{\frac{1}{2}(x^2-4^2)} e^{-u} \, du = 1 - e^{-\frac{1}{2}(x^2 - 4^2)}, \quad x > 4,
$$

and $G_X(x) = 0$ for $x \leq 4$. Let $U = G_X(X) \sim \text{Uniform}(0, 1)$. Then we solve for X.

$$
\begin{aligned}
U &= 1 - e^{-\frac{1}{2}(X^2 - 4^2)} \\
-\frac{1}{2}(X^2 - 4^2) &= \log(1 - U) \\
X &= \sqrt{4^2 - 2\log(1 - U)}, \quad U \sim \text{Uniform}(0, 1).
\end{aligned}
$$

Let $X_1, \ldots, X_n$ be a sample drawn from the proposal distribution $g_X(x)$. Then the importance sampling estimator of θ is given by

$$
\hat{\theta}_{\text{IS}} = \frac{1}{n} \sum_{i=1}^n h(X_i) w(X_i),
$$

where $w(x) = f_X(x)/g_X(x)$, with expectation

$$
\begin{aligned}
\mathrm{E}\left[\hat{\theta}_{\text{IS}}\right] &= \frac{1}{n} \sum_{i=1}^n \int_0^\infty h(x_i) w(x_i) g_X(x_i) \, dx_i \\
&= \frac{1}{n} \sum_{i=1}^n \int_0^\infty h(x_i) f_X(x_i) \, dx_i \\
&= \frac{1}{n} \sum_{i=1}^n \mathrm{E}\left[h(X_i) \mid X_i \sim \mathrm{N}(0, 1)\right] \\
&= \frac{1}{n} \sum_{i=1}^n \theta \\
&= \theta,
\end{aligned}
$$

and sampling variance

$$
\widehat{\text{Var}}\left[\hat{\theta}_{\text{IS}}\right] = \frac{1}{n^2} \sum_{i=1}^n \widehat{\text{Var}}\left[h(X_i) w(X_i)\right] = \frac{1}{n} \widehat{\text{Var}}[h(X) w(X)] = \frac{1}{n(n-1)} \sum_{i=1}^n \left(h(X_i) w(X_i) - \hat{\theta}_{\text{IS}} \right)^2.
$$

```r
set.seed(321)

sample_from_proposal <- function(n) {
  u <- runif(n)
  sqrt(4^2 - 2 * log(1 - u))
}
```

```r
n <- 100000
x <- sample_from_proposal(n)

w <- function(x) {
  f <- dnorm(x)                                # target density
  g <- ifelse(                                 # proposal density
         test = x > 4,
         yes  = x * exp(-0.5 * (x^2 - 16)),
         no   = 0
       )
  return(f / g)
}

hw <- h(x) * w(x)

theta_IS <- mean(hw)      # Importance sampling estimate of Pr(X > 4)
sample_var_IS <- var(hw)  # Sampling variance

t <- qt(0.05/2, df = n - 1, lower.tail = FALSE)  # quantile with 5% significance level
ci_IS <- theta_IS + t * sqrt(sample_var_IS / n) * c(-1, 1)  # Confidence Interval

# Result
list(
  theta_IS      = theta_IS,
  sample_var_IS = sample_var_IS,
  confint       = ci_IS,
  error         = abs(theta_IS - theta)
)
```

```
## $theta_IS
## [1] 3.167611e-05
##
## $sample_var_IS
## [1] 2.410122e-12
##
## $confint
## [1] 3.166649e-05 3.168573e-05
##
## $error
## [1] 4.866683e-09
```

The number of samples m needed for the simple Monte Carlo estimator to achieve the same precision as the importance sampling approach, we would need

$$m = n\frac{\widehat{\mathrm{Var}}[h(X)]}{\widehat{\mathrm{Var}}[h(X)w(X)]} = 10^5\frac{5.9997 \times 10^{-5}}{2.4101218 \times 10^{-12}} = 2.4893763 \times 10^{12},$$

samples. That is, we need about 10 million times more samples.

3.

(a)

We modify `sample_from_proposal()` to return a pair of samples, where one takes $U \sim \mathrm{Uniform}(0, 1)$ as argument and the other $1 - U$ as argument.

```r
sample_from_proposal_mod <- function(n) {
  u <- runif(n)
  list(
    x_1 = sqrt(4^2 - 2 * log(1 - u)),
    x_2 = sqrt(4^2 - 2 * log(u))
  )
}
```

(b)

```r
set.seed(53)
n <- 50000
sample_pair <- sample_from_proposal_mod(n)

hw1 <- h(sample_pair$x_1) * w(sample_pair$x_1)
hw2 <- h(sample_pair$x_2) * w(sample_pair$x_2)
hw_AS <- 0.5 * (hw1 + hw2)  # Antithetic sample

theta_AS <- mean(hw_AS)      # Antithetic sampling estimate of Pr(X > 4)
sample_var_AS <- var(hw_AS)  # Sampling variance

t <- qt(0.05/2, df = n - 1, lower.tail = FALSE)  # quantile with 5% significance level
ci_AS <- theta_AS + t * sqrt(sample_var_AS / n) * c(-1, 1)  # Confidence Interval

# Result
list(
  theta_AS      = theta_AS,
  sample_var_AS = sample_var_AS,
  confint       = ci_AS,
  error         = abs(theta_AS - theta)
)
```

```
## $theta_AS
## [1] 3.167307e-05
##
## $sample_var_AS
## [1] 2.849882e-13
##
## $confint
## [1] 3.166839e-05 3.167774e-05
##
## $error
## [1] 1.823745e-09
```

Problem D: Rejection sampling and importance sampling

1.

2.

3.

4.