CompStatLab3

Xuan Wang & Priyarani Patil

2023-11-16

Computational Statistics 732A90 | Computer Lab 3

Question 1: Sampling algorithms for a triangle distribution

a

See Appendix.

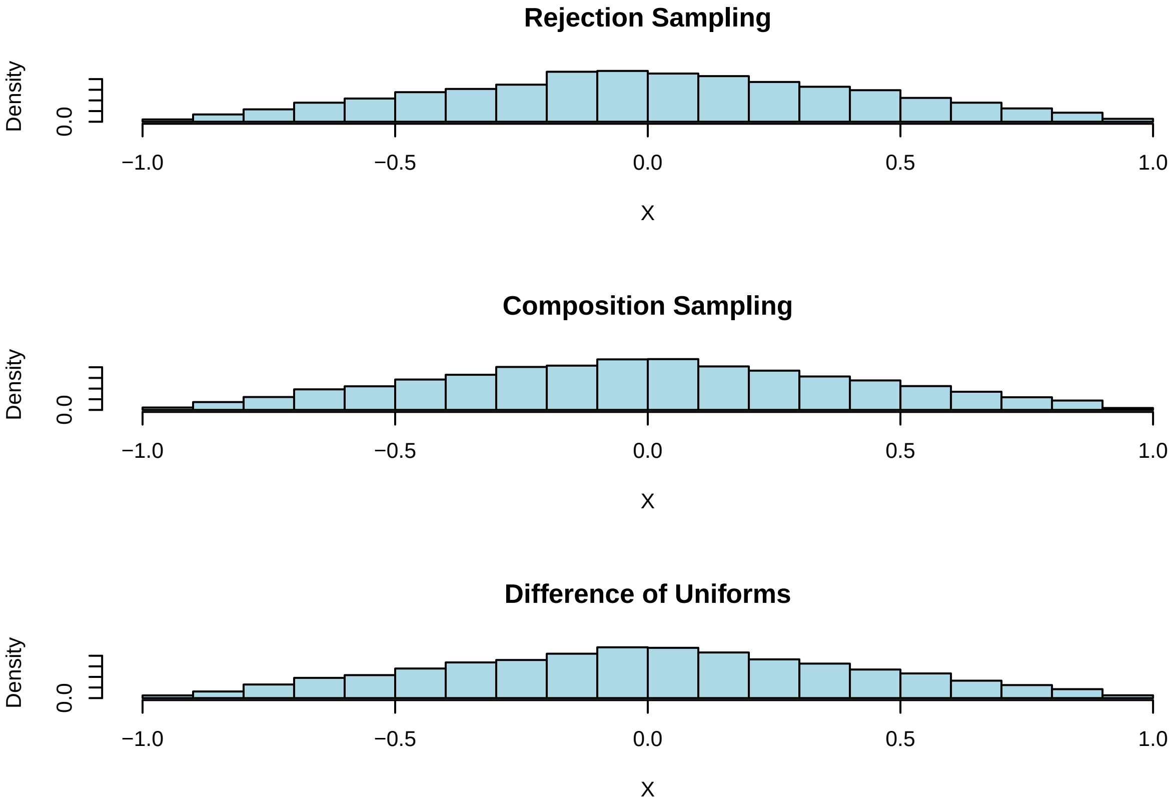
b

See Appendix.

c

See Appendix.

d



```
## Variance of X (Rejection Sampling): 0.16629
```

```
## Variance of X (Composition Sampling): 0.167099
```

```
## Variance of X (Differences of Uniforms): 0.1675804
```

According to above histograms and variance results, it's preferred to choose uniform difference sampling, as its histogram more accurately reflect the properties of the target density distribution. And its variance matches the theoretical variance of the target distribution. Therefore, uniform difference sampling is better.

Question 2: Laplace distribution

a

The double exponential (Laplace) distribution is given by formula:

$$DE(\mu, \lambda) = \frac{\lambda}{2} e\left(-\lambda|x-\mu\right)$$

where, x is a real number,μ is a location parameter, and λ is the scale parameter.

The cumulative density function (CDF) of a continuous random variable X is defined as:

$$F(x) = \int_{-\infty}^{x} f(x)dx, -\infty < x < \infty$$

The cumulative distribution for the double exponential distribution: For $x > \mu$:

$$F(x) = \int_{-\infty}^{x} \lambda/2e^{-\lambda(x-\mu)}dx, x > \mu$$

For $x <= \mu$:

$$F(x) = 1 - \int_{x}^{\infty} \lambda/2e^{-\lambda(x-\mu)}dx, x <= \mu$$

Integrating with respect to x, it's given:

For $x > \mu$:

$$F(x) = 1/2e^{\lambda(x-\mu)}$$

For $x <= \mu$:

$$F(x) = 1 - 1/2e^{-\lambda(x-\mu)}$$

The inverse CDF: Below illustrates how to solve the equation F(x)=U for x:

For $U > 1/2$,

$$U = 1/2e^{\lambda(x-\mu)}$$

solving for x,

$$x = \mu + log(2U)/\lambda$$

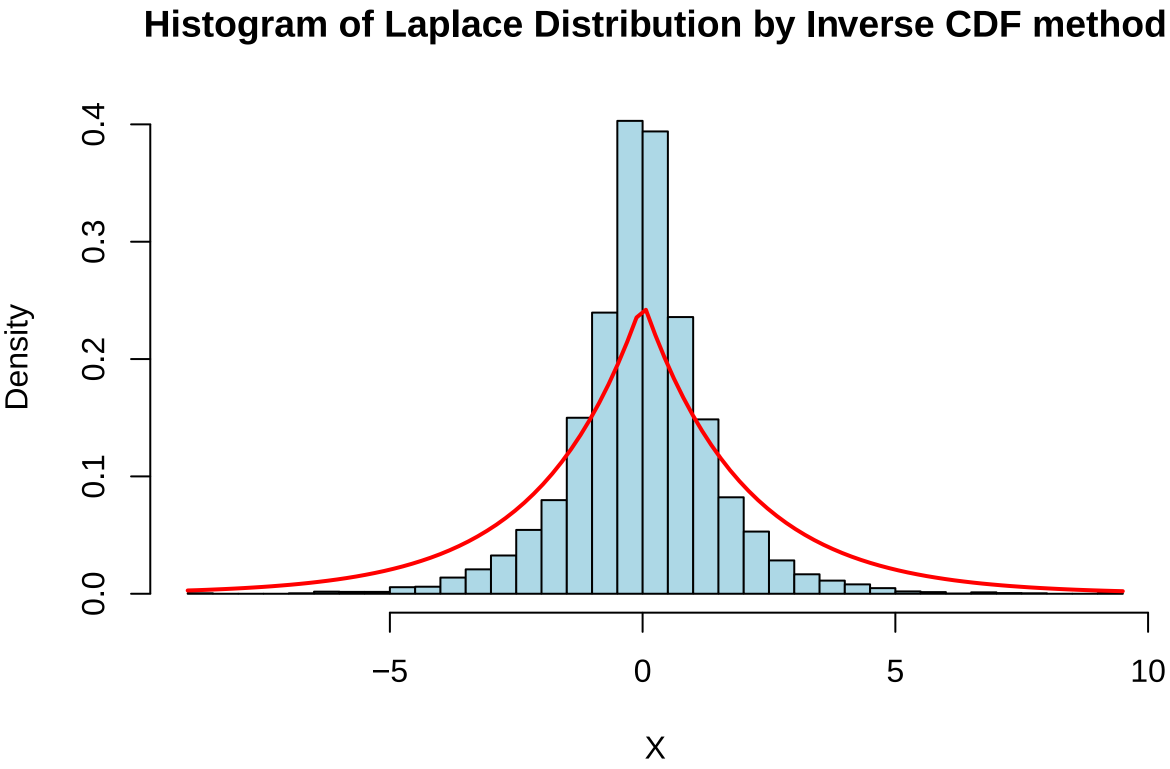
For $U < 1/2$,

$$U = 1 - 1/2e^{\lambda(x-\mu)}$$

solving for x,

$$x = \mu - log(2 - 2U)/\lambda$$

Note: where U is a probability generated from a uniform distribution Unif(0,1).



The histogram supports the results because it closely matches the Laplace distribution plot. This means that the generated random numbers are accurate.

b

Rejection sampling is a method for generating random numbers from a target distribution by using a proposal distribution. In this case, we want to generate numbers from the standard normal distribution N(0, 1) using the Laplace distribution as the proposal distribution.

The constant a is chosen such that $a * f(x)$ is always greater than or equal to $g(x)$ for all x, where $f(x)$ is the PDF of the proposal distribution (Laplace) and $g(x)$ is the PDF of the target distribution (Normal). For the standard Laplace and normal distributions, a can be chosen as $sqrt(2 * exp(1)/pi)$.

For rejection sampling method, $g(x)/a \geq f(x)$ for all x and the constant $a < 1$, so that $g(x)/a$ is always greater than or equal to $f(x)$, where $g(x)$ is the PDF of the proposal distribution(Laplace) and $f(x)$ is the PDF of the target distribution(Normal).

Therefore the proposed distribution $g(x) \sim \mathcal{DE}(0, 1)$ is:

$$g(x) = \frac{1}{2}e^{-|x|}$$

and the target distribution $f(x) \sim \mathcal{N}(0, 1)$ is:

$$f(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$$

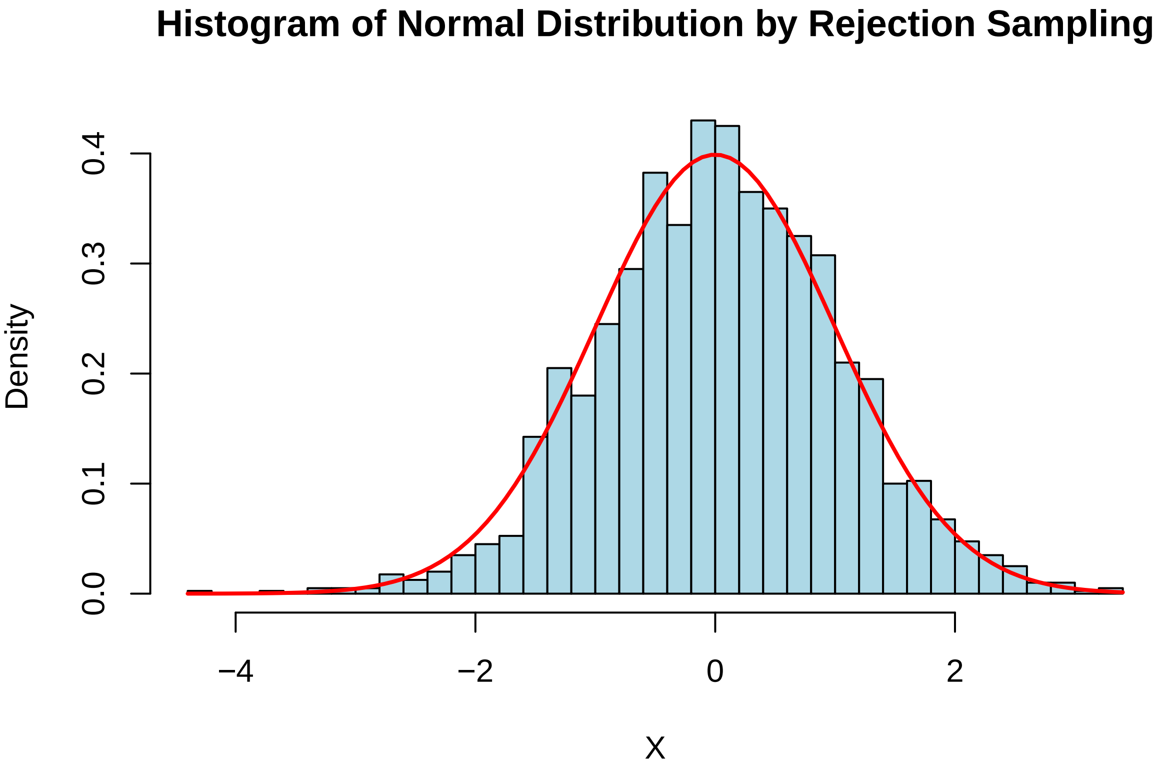
Evaluating the constant a,

$$a \geq \frac{f(x)}{g(x)}$$

$$a \geq \sqrt{\frac{2}{\pi}} e^{-\frac{x^2}{2}+|x|}$$

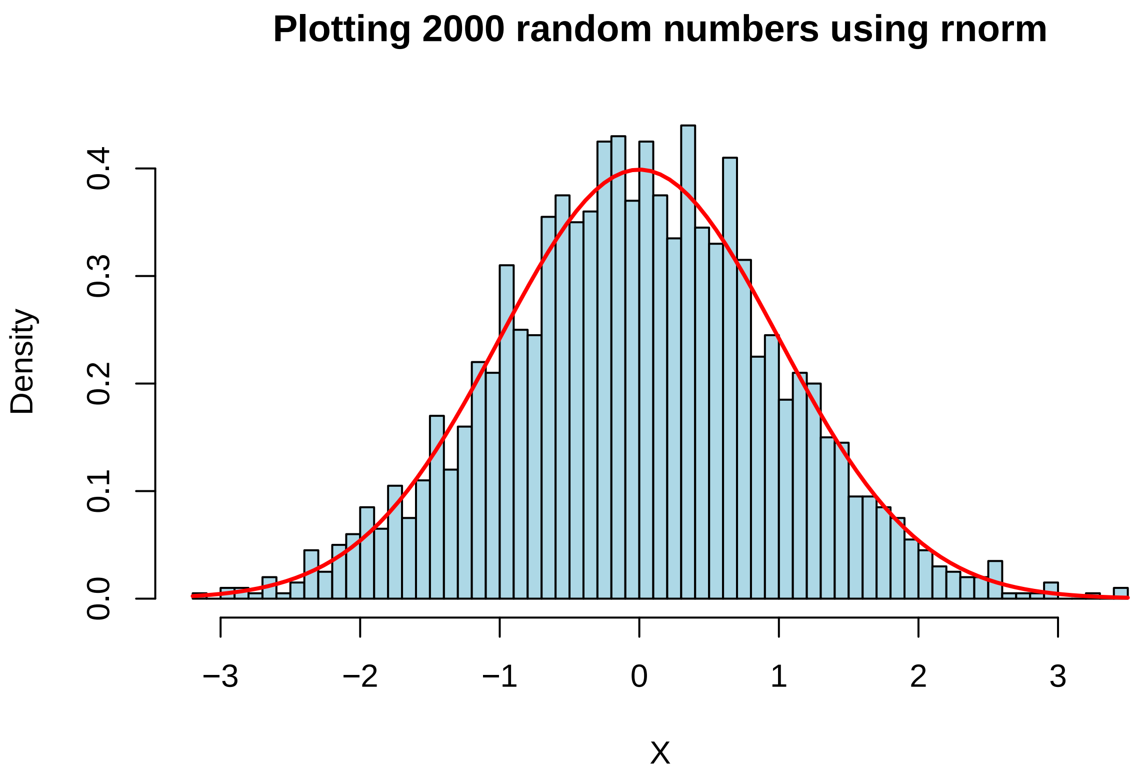
we obtain maximum at x=1 and by setting the first derivative to 0 a will be:

$$a = \sqrt{\frac{2}{\pi} e}$$



```
## Average rejection rate (R): 0.3065
```

```
## Expected rejection rate (ER): 0.2398265
```



The expected rejection rate and average rejection rate are very close, showing only slight differences. When we look at the histogram plots of two methods—one using the acceptance/rejection technique and the other using the rnorm() function—they look quite similar and balanced.

Appendix:

Question 1

a

```r
# rejection sampling
rejection_sampling <- function(n) {
  x <- runif(n, min = -1, max = 1)
  y <- runif(n, min = 0, max = 1)
  envelop_fx <- ifelse(x <= 0, x + 1, 1 - x)
  accepted <- y <= envelop_fx
  return(x[accepted])
}
```

b

```r
# composition sampling
composition_sampling <- function(n) {
  Y <- ifelse(runif(n) < 0.5, sqrt(runif(n)) - 1, 1 - sqrt(1 - runif(n)))
  X <- ifelse(runif(n) < 0.5, Y, -Y)
  return(X)
}
```

c

```r
# difference of uniforms
difference_of_uniforms <- function(n) {
  U1 <- runif(n)
  U2 <- runif(n)
  return(U1 - U2)
}
```

d

```r
# checking the random generators
set.seed(123)
n <- 10000
par(mfrow = c(3, 1))

# Rejection Sampling
x_rejection_samples <- rejection_sampling(n)
hist(x_rejection_samples, main = "Rejection Sampling", xlab = "X", col = "lightblue", freq = FALSE)

# Composition Sampling
x_composition_samples <- composition_sampling(n)
hist(x_composition_samples, main = "Composition Sampling", xlab = "X", col = "lightblue", freq = FALSE)

# Difference of Uniforms
x_difference_uniforms_samples <- difference_of_uniforms(n)
hist(x_difference_uniforms_samples, main = "Difference of Uniforms", xlab = "X", col = "lightblue", fre

# Calculate variance for each method
var_rejection <- var(x_rejection_samples)
var_composition <- var(x_composition_samples)
var_difference <- var(x_difference_uniforms_samples)

cat("Variance of X (Rejection Sampling):", var_rejection, "\n")
cat("Variance of X (Composition Sampling):", var_composition, "\n")
cat("Variance of X (Differences of Uniforms):", var_difference, "\n")
```

Question 2

a

```r
# inverse CDF
# default that mu = 0; lambda =1
inverse_cdf_laplace <- function(n) {
  U <- runif(n)
  X <- ifelse(U < 0.5, log(2*U), -log(2*(1-U)))
  return(X)
}

# Generate 10000 random numbers and plot histogram
set.seed(123)
x <- inverse_cdf_laplace(10000)
hist(x, breaks=50, prob=TRUE, main = "Histogram of Laplace Distribution by Inverse CDF method", xlab =
# Add theoretical density
curve(dexp(abs(x), rate=1/2)/2, add=TRUE, col="red", lwd=2)
```

b

```
# rejection sampling for Normal distribution
rejection_sampling_normal <- function(n) {
  a <- sqrt(2 *exp(1)/pi) # Envelop constant for Laplace distribution
  X <- rep(NA, n)
  i <- 1
  count_rejected <- 0

  while(i <= n) {
    Z <- inverse_cdf_laplace(1)
    U <- runif(1)
    f_y <- 2/sqrt(2*pi) * exp(-Z^2/2)
    g_y <- exp(-abs(Z))
    if (U <= (f_y /(a * g_y))) {
      X[i] <- Z
      i <- i + 1
    } else {
      count_rejected <- count_rejected + 1
    }
  }

  rejection_rate <- count_rejected / n
  return(list(samples = X, rejection_rate = rejection_rate))
}

# Generate 2000 random numbers and plot histogram
set.seed(123)
results <- rejection_sampling_normal(2000)
x <- results$samples
hist(x, breaks=50, prob=TRUE, main = "Histogram of Normal Distribution by Rejection Sampling", xlab = "X

# Add theoretical density
curve(dnorm(x), add=TRUE, col="red", lwd=2)

# Compute rejection rate
rejection_rate <- results$rejection_rate

# Compare rejection rates
cat("Average rejection rate (R):", rejection_rate, "\n")
cat("Expected rejection rate (ER):", 1 - (1 / sqrt(2 *exp(1)/pi)) , "\n")

# Generate 2000 random numbers using rnorm and plot histogram
y <- rnorm(2000)
hist(y, breaks=50, prob=TRUE, main = "Plotting 2000 random numbers using rnorm", xlab = "X", col = "lig

# Add theoretical density
curve(dnorm(x), add=TRUE, col="red", lwd=2)
```