

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.

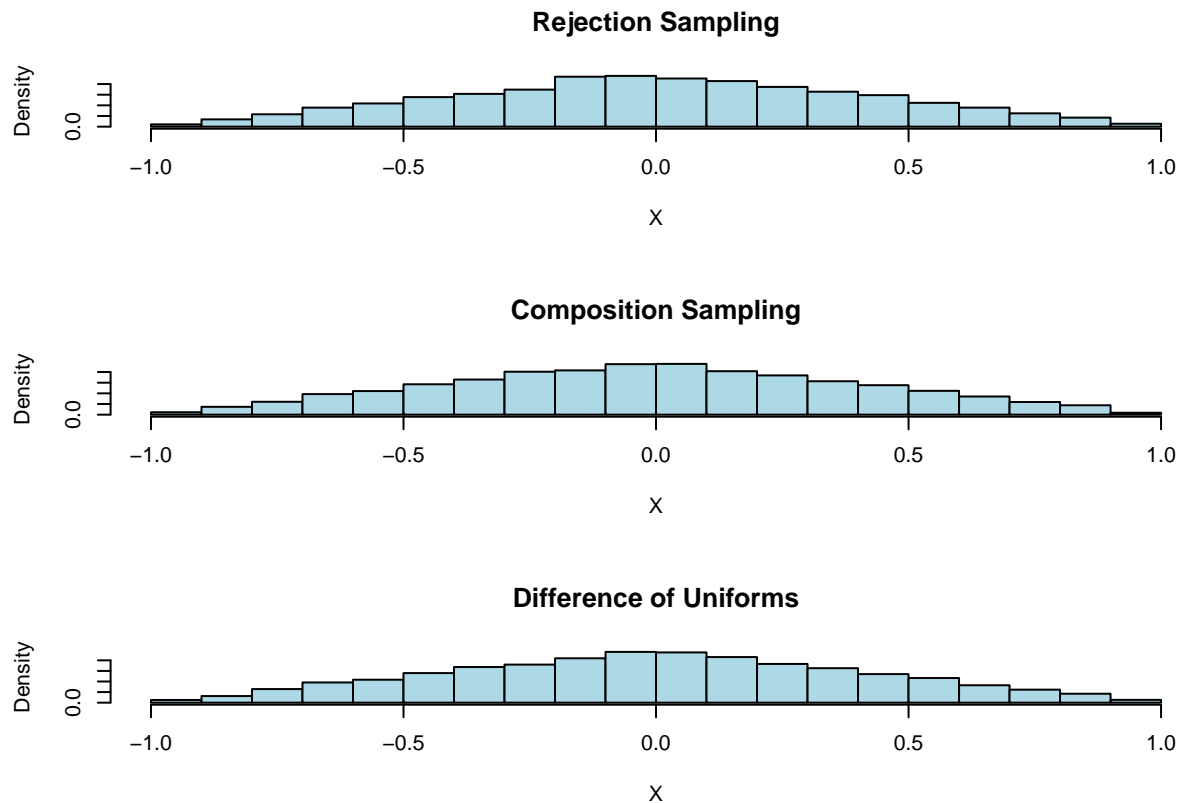
b

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 cumulative distribution function(CDF) of Laplace(double exponential) distribution is given by:

$$F(x, \mu, \lambda) = 0.5 + 0.5 * \text{sgn}(x - \mu) * (1 - e^{-\lambda * |x - \mu|})$$

where:

μ is the location parameter

λ is the scale parameter

sgn is the sign function

Instead of using sign function, we can split the distribution into two parts based on whether x is less than or greater than μ . This gives:

$$\text{For } x < \mu : F(x, \mu, \lambda) = 0.5 * e^{\lambda * (x - \mu)}$$

$$\text{For } x \geq \mu : F(x, \mu, \lambda) = 1 - 0.5 * e^{-\lambda * (x - \mu)}$$

So the inverse of the CDF is given by:

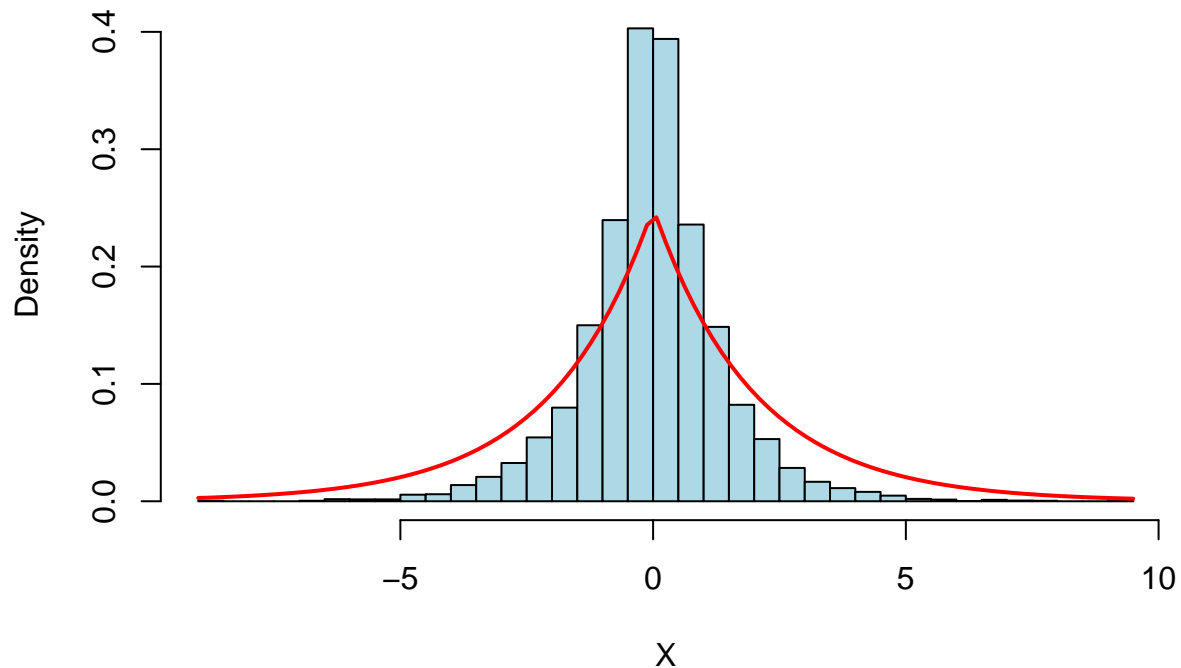
$$\text{For } p < 0.5 : F^{-1}(x, \mu, \lambda) = \mu + \log(2p)/\lambda$$

$$\text{For } p \geq 0.5 : F^{-1}(x, \mu, \lambda) = \mu - \log(2(1 - p))/\lambda$$

where p is a probability generated from a uniform distribution $\text{Unif}(0,1)$.

Finally, please check the Appendix for the R code and below histogram.

Histogram of Laplace Distribution by Inverse CDF method

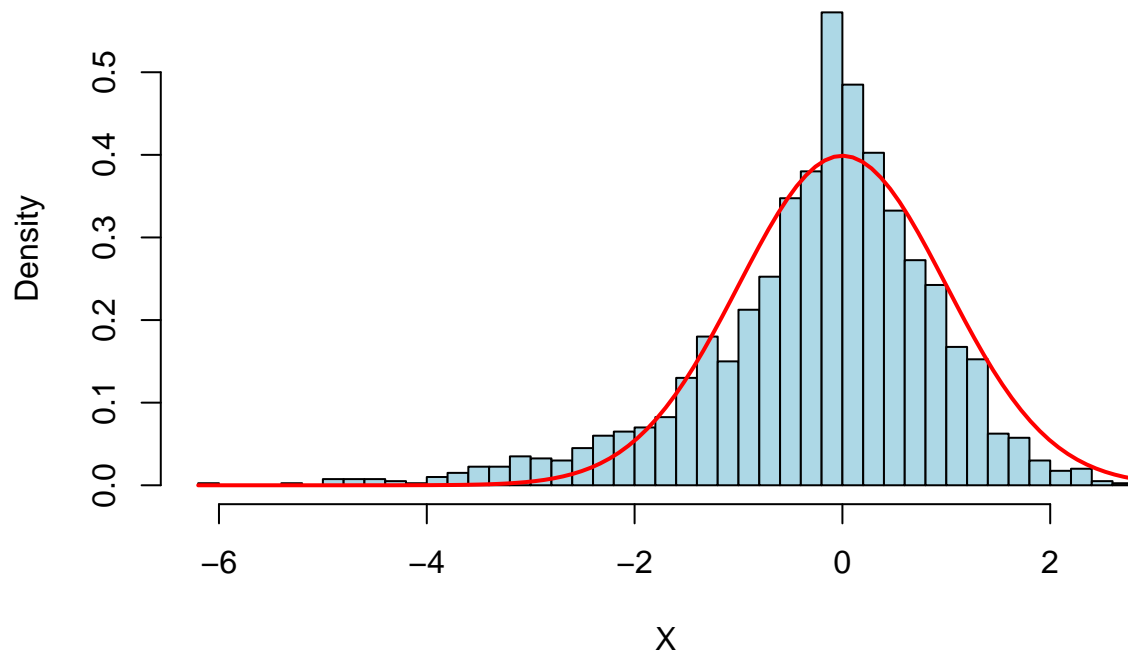


b

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 $\sqrt{2/e}$.

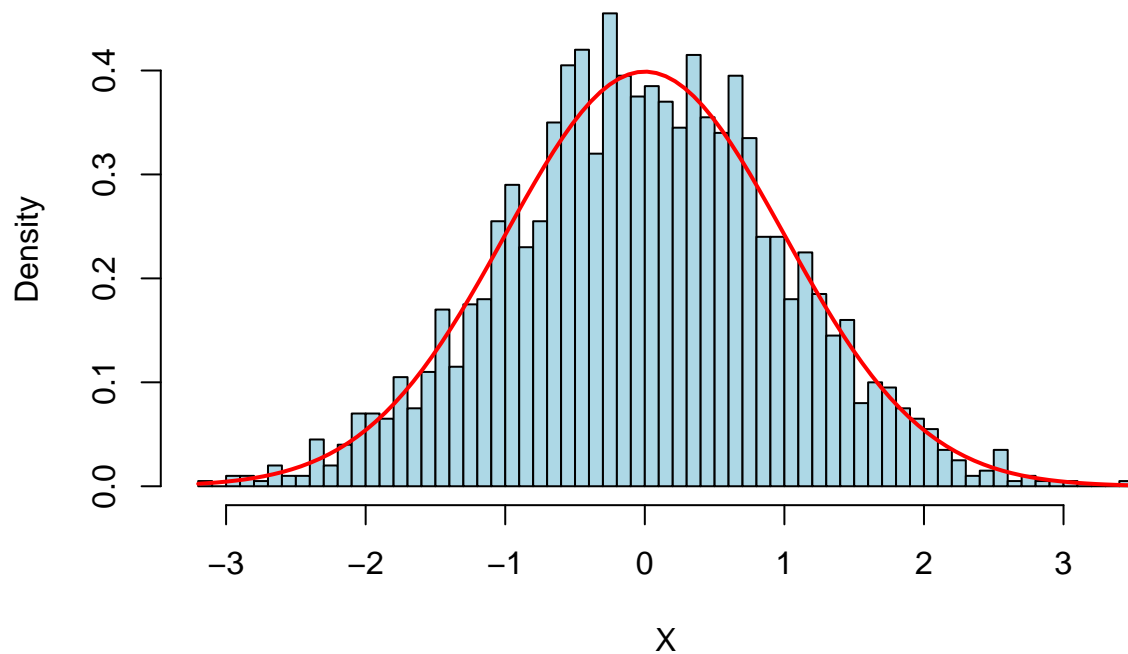
Histogram of Normal Distribution by Rejection Sampling



Average rejection rate (R): 0.1015

Expected rejection rate (ER): -0.165822

rnorm



The average rejection rate R can be computed by counting the number of rejected samples divided by the total number of samples. The expected rejection rate ER is $1 - 1/a$.

It can be seen that two histograms shows some discrepancy, it could be due to a few reasons:

- 1) both methods generate random numbers, so there will always be some variability in the results. This is especially possible when the number of samples is relatively small.
- 2) The Laplace distribution used as the proposal distribution in the rejection sampling method has heavier tails than the normal distribution. This means that it is more likely to generate extreme values, which could explain why there are more values in the range -6 to -3. It also explains the expected rejection rate is negative. This is because the Laplace distribution is not a perfect envelope for the normal distribution. The tails

Appendix:

Question 1

a

```
# 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

```
# 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

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

d

```
# 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", freq = FALSE)

# 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

```

# 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 = "X")

# 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)) # 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)
    if (U <= dnorm(Z) / (a * dexp(Z, rate=1/2))) {
      X[i] <- Z
    } else {
      count_rejected <- count_rejected + 1
    }
    i <- i + 1
  }
}

```

```

    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))) , "\n")

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

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

```