

Lab 4 - Computational Statistics (732A89)

Helena Llorens Lluís (hllor282), Yi Yang (yiyang338)

QUESTION 1: Computations with Metropolis–Hastings

First, the target function is defined as

$$f(x) = 120x^5e^{-x}, \quad x > 0$$

We are going to generate a Markov chain with the Metropolis-Hastings (MH) algorithm, which generates a sequence of dependent observations which follow the target distribution approximately. The next observation of the chain is generated based on a proposal distribution which depends on the current observation.

For this assignment, we are asked to try three different proposal distributions.

1. Normal: $g_1(X) \sim N(\mu = X_t, \sigma = 0.1)$
2. Chi squared: $g_2(X) \sim \chi^2(\lfloor X_t + 1 \rfloor)$
3. Uniform: $g_3(X) \sim U(X_t - 0.5, X_t + 0.5)$

The Uniform is proposed because it is a symmetric distribution, and it makes the computation straightforward. Compared to the normal proposal, which can generate very small steps, the uniform proposal ensures a consistent range of exploration. Unlike the Chi-square proposal, which can introduce very large jumps, the uniform step size prevents excessive variation.

For the first proposed distribution, the chain looks as follows.

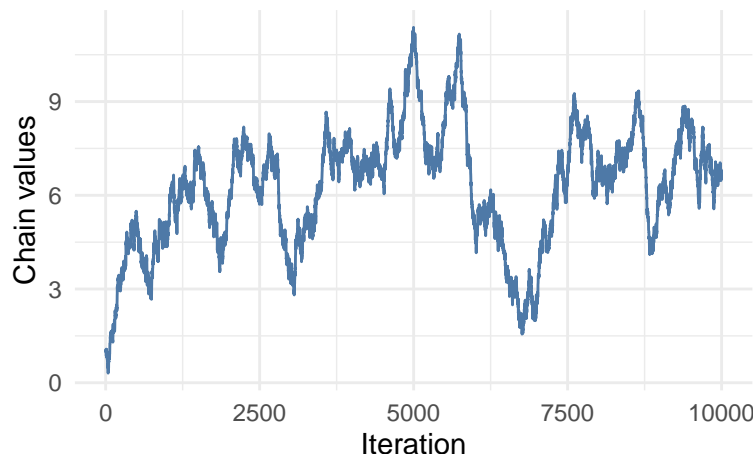


Figure 1: Chain with proposed Normal distribution

The chain appears to mix well and explore the distribution efficiently. The acceptance rate is 0.985, indicating that most proposals are accepted. However, the small step size may slow convergence. The plot shows a

clear upward trend in the first 2000-2500 iterations, meaning the chain is still moving toward the stationary distribution during that phase. For this chain, a burn-in of 2500 iterations is recommended to remove the initial transient phase.

In the plot below, we compare the sample distribution (histogram) with the density of the target distribution (orange line).

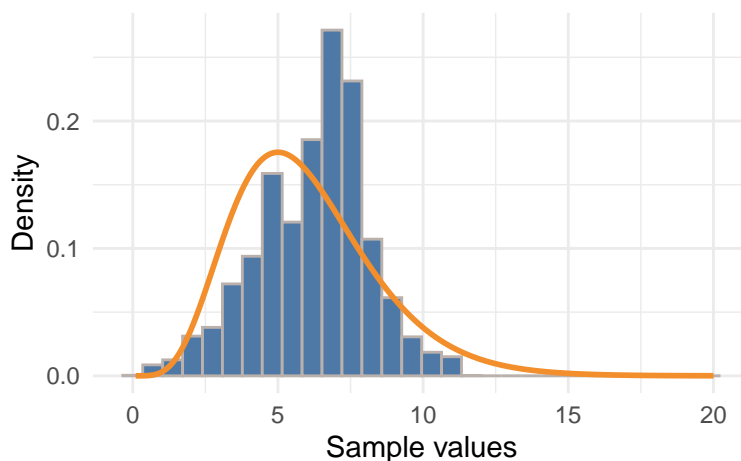


Figure 2: Histogram of sample with Normal proposed distribution

For the second proposed distribution, the chain looks as follows.

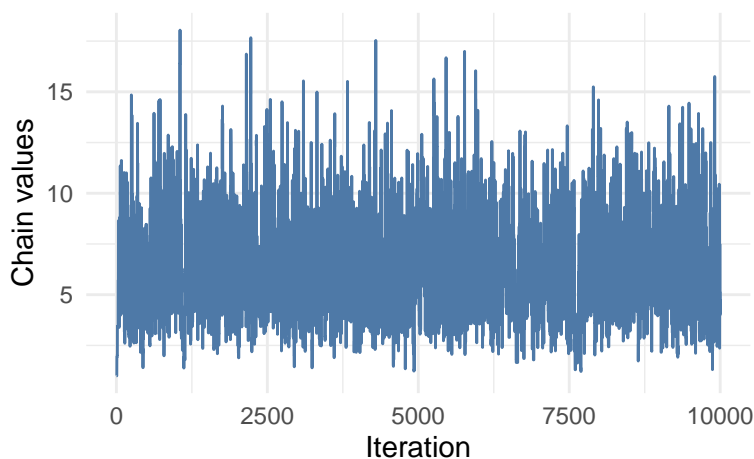


Figure 3: Chain with proposed Chi distribution

The chain exhibits larger jumps, suggesting a more exploratory behavior. The acceptance rate is 0.5288, lower than the normal case, meaning more proposals are rejected. This could lead to a more robust exploration of the target distribution. The chain seems to fluctuate around a stationary distribution almost immediately, which suggest a sort burn-in, about the first 500 iterations.

In the plot below, we compare the sample distribution with the density of the target distribution.

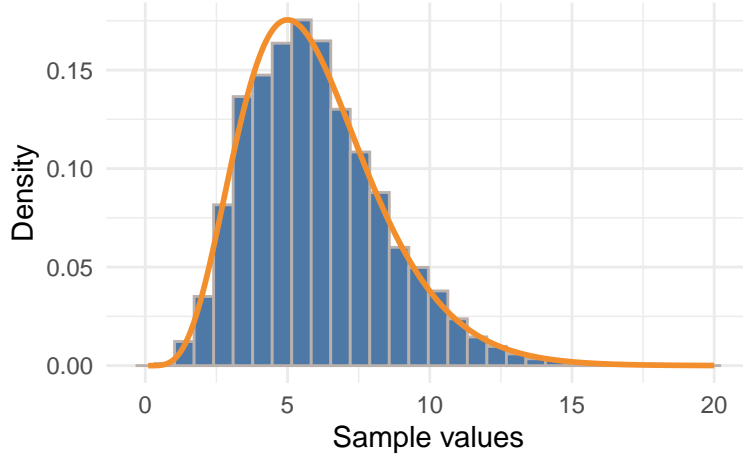


Figure 4: Histogram of sample with Chi proposed distribution

For the third proposed distribution, the chain looks as follows.

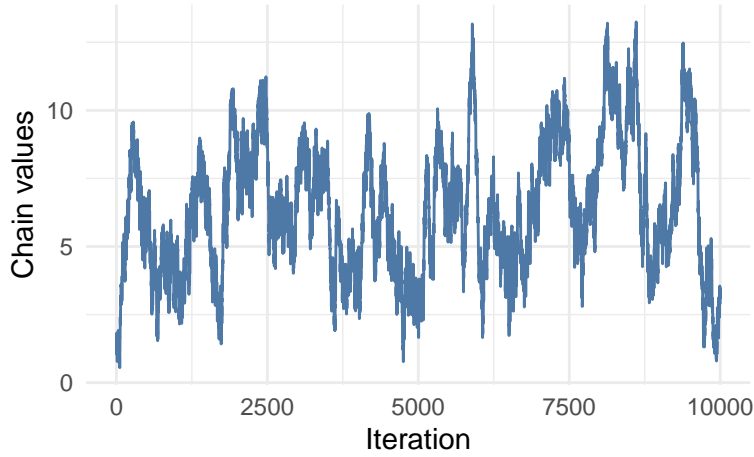


Figure 5: Chain with proposed Uniform distribution

The chain moves moderately between states with an acceptance rate of 0.956. The Uniform proposal offers a balanced exploration but may still be less efficient than the normal proposal. The chain appears to start at lower values and gradually reach a region of more stable fluctuation. The first few iterations (likely the first 1,000 to 2,000 iterations) show a clear upward trend before the chain appears to mix well.

In the plot below, we compare the sample distribution with the density of the target distribution.

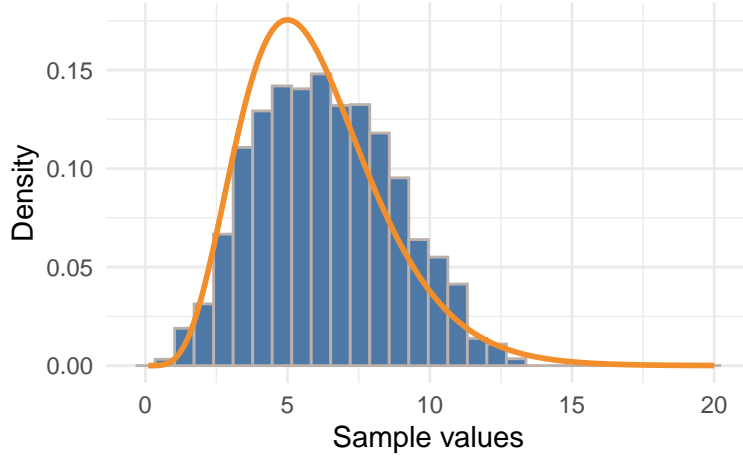


Figure 6: Histogram of sample with Uniform proposed distribution

We can estimate $E(X)$ from the samples as

$$E(X) \approx \frac{1}{N} \sum_{t=1}^N X_t$$

The results are shown in the following table.

Table 1: Sample mean values

Proposal.distribution	Mean.values
Normal	6.238768
Chi	6.031685
Uniform	6.357880

Our target distribution follows a $Gamma(\alpha = 6, \beta = 1)$, since the $Gamma$ distribution has following probability density function:

$$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$$

Then, we can calculate rge theoretical expected value as

$$E(X) = \alpha\beta = 6 \cdot 1 = 6$$

As observed, the MH algorithm with the three proposal distributions provides a reliable estimation of $E(X)$.

Appendix

```
library(ggplot2)
library(knitr)
```

```

# define target function
f <- function(x){
  ifelse(x > 0, 120*(x^5)*exp(-x), 0)
}

# Normalize f(x) so it integrates to 1
normalizing_constant <- integrate(f, lower = 0, upper = Inf)$value
f_normalized <- function(x) f(x) / normalizing_constant

# Generate points for the normalized density function
x_f <- seq(0.1, 20, length.out = 1000)
y_f <- sapply(x_f, f_normalized)
data_f <- data.frame(x = x_f, y = y_f)

# MH algorithm function
MetropolisHastings <- function(x_init, n_iter, prop_dist, prop_dist_rand, target){
  set.seed(1234)

  # initial values
  x <- numeric(length = n_iter)
  x[1] <- x_init
  acc <- 0

  for(t in 2:(n_iter)){

    # get candidate for x from the proposal distribution
    x_candidate <- prop_dist_rand(x[t - 1])

    # calculate acceptance rate
    r <- (target(x_candidate) * prop_dist(x[t-1], x_candidate)) / (target(x[t - 1]) * prop_dist(x_candidate, x[t-1]))

    if (runif(1) < min(r, 1)) {
      # Accept new candidate
      x[t] <- x_candidate
      acc <- acc + 1
    } else {
      # Stay at previous value
      x[t] <- x[t - 1]
    }
  }

  return(list(chain = x, acceptance_rate = sum(acc)/n_iter))
}

prop_normal <- function(x1, x2){dnorm(x1, mean = x2, sd = 0.1)}
prop_normal_rand <- function(x){rnorm(1, mean = x, sd = 0.1)}
res1 <- MetropolisHastings(1, 10000, prop_normal, prop_normal_rand, f)

prop_chi <- function(x1, x2){dchisq(x1, df = floor(x2))}
prop_chi_rand <- function(x){rchisq(1, df = floor(x))}
res2 <- MetropolisHastings(1, 10000, prop_chi, prop_chi_rand, f)

```

```

prop_unif <- function(x1, x2){dunif(x1, min = x2 - 0.5, max = x2 + 0.5)}
prop_unif_rand <- function(x){runif(1, x - 0.5, x + 0.5)}
res3 <- MetropolisHastings(1, 10000, prop_unif, prop_unif_rand, f)

data <- data.frame(it = 1:10000, x1 = res1$chain, x2 = res2$chain, x3 = res3$chain)

ggplot(data, aes(x = it, y = x1)) +
  geom_line(col = "#4E79A7") +
  theme_minimal() +
  labs(x = "Iteration", y = "Chain values")

ggplot(data, aes(x = x1)) +
  geom_histogram(aes(y = ..density..), color = "#BAB0AC", fill = "#4E79A7", bins = 30) +
  geom_line(data = data_f, aes(x = x, y = y), color = "#F28E2B", linewidth = 1) +
  theme_minimal() +
  labs(x = "Sample values", y = "Density")

ggplot(data, aes(x = it, y = x2)) +
  geom_line(col = "#4E79A7", size = 0.5) +
  theme_minimal() +
  labs(x = "Iteration", y = "Chain values")

ggplot(data, aes(x = x2)) +
  geom_histogram(aes(y = ..density..), color = "#BAB0AC", fill = "#4E79A7", bins = 30) +
  geom_line(data = data_f, aes(x = x, y = y), color = "#F28E2B", linewidth = 1) +
  theme_minimal() +
  labs(x = "Sample values", y = "Density")

ggplot(data, aes(x = it, y = x3)) +
  geom_line(col = "#4E79A7") +
  theme_minimal() +
  labs(x = "Iteration", y = "Chain values")

ggplot(data, aes(x = x3)) +
  geom_histogram(aes(y = ..density..), color = "#BAB0AC", fill = "#4E79A7", bins = 30) +
  geom_line(data = data_f, aes(x = x, y = y), color = "#F28E2B", linewidth = 1) +
  theme_minimal() +
  labs(x = "Sample values", y = "Density")

mean_values <- c(mean(data$x1), mean(data$x2), mean(data$x3))
distribution <- c("Normal", "Chi", "Uniform")
mean_data <- data.frame("Proposal distribution" = distribution, "Mean values" = mean_values)
kable(mean_data, caption = "Sample mean values")

```