Lab 4 - Computational Statistics (732A89)

Helena Llorens Lluís (hllor282), Yi Yang (yiyan338)

QUESTION 1: Computations with Metropolis–Hastings

First, the target function is defined as

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

The function has the following density.

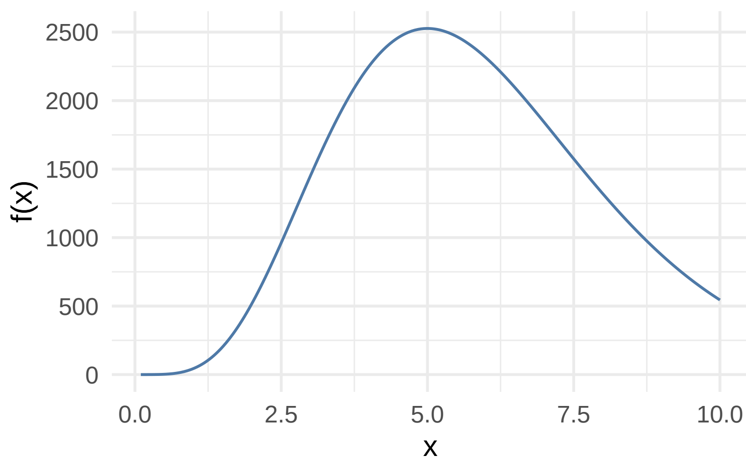

Figure 1: Plot of f(x)

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.

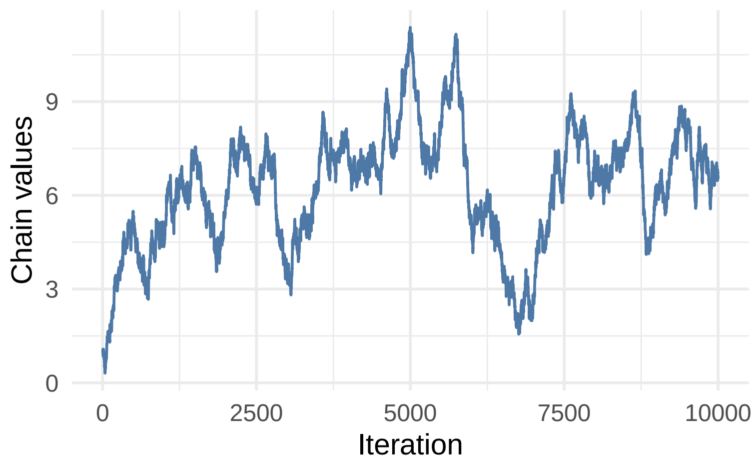

Figure 2: Chain with proposed Normal distribution

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 first 2500 iterations show a clear upward drift, meaning they should be discarded as burn-in.

This sample has the following distribution.

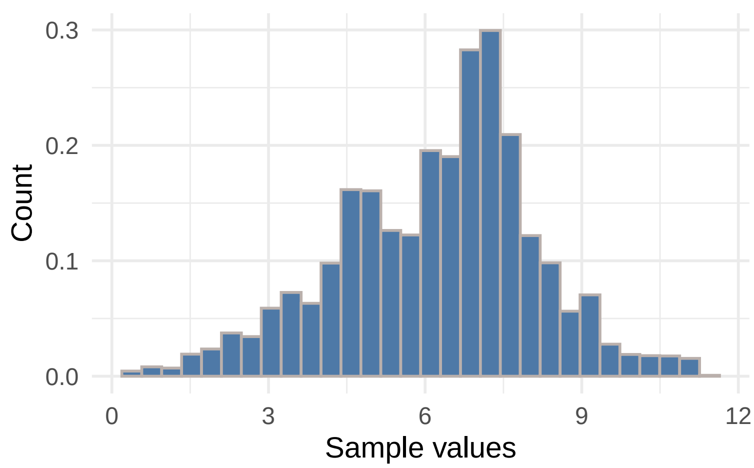

Figure 3: Histogram of sample with Normal proposed distribution

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

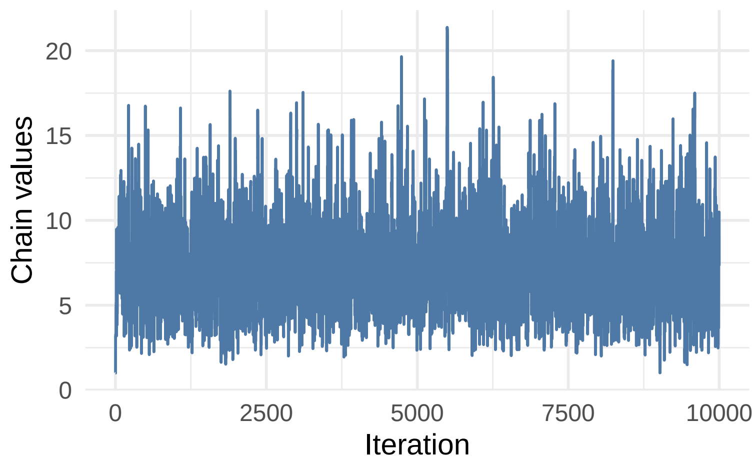

Figure 4: Chain with proposed Chi distribution

The chain exhibits larger jumps, suggesting a more exploratory behavior. The acceptance rate is 0.6001, 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 suggests a short burn-in, about the first 500 iterations.

This sample has the following distribution.

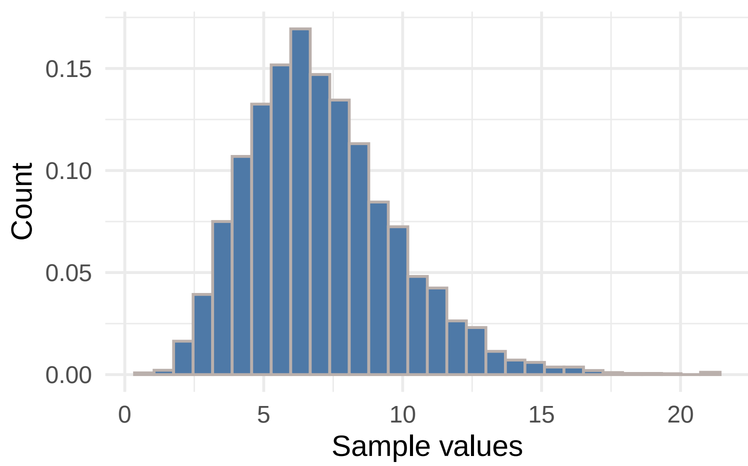

Figure 5: Histogram of sample with Chi proposed distribution

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

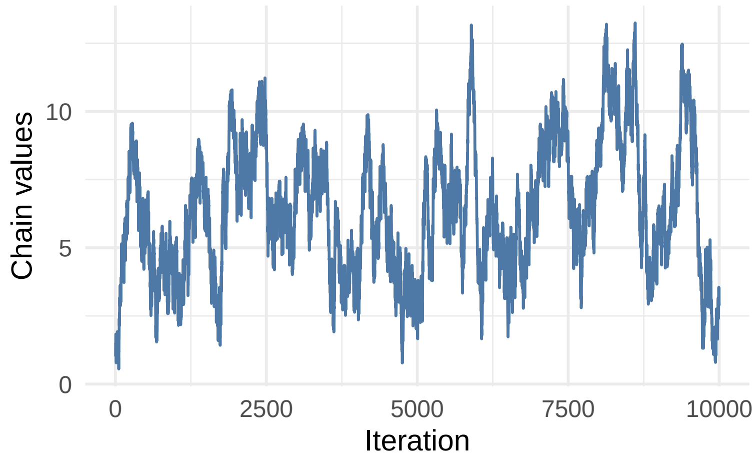

Figure 6: Chain with proposed Uniform distribution

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 starts from a low value and fluctuates a lot in the first 1000–2000 iterations, so a burn-in of around 2000 iterations would be reasonable.

This sample has the following distribution.

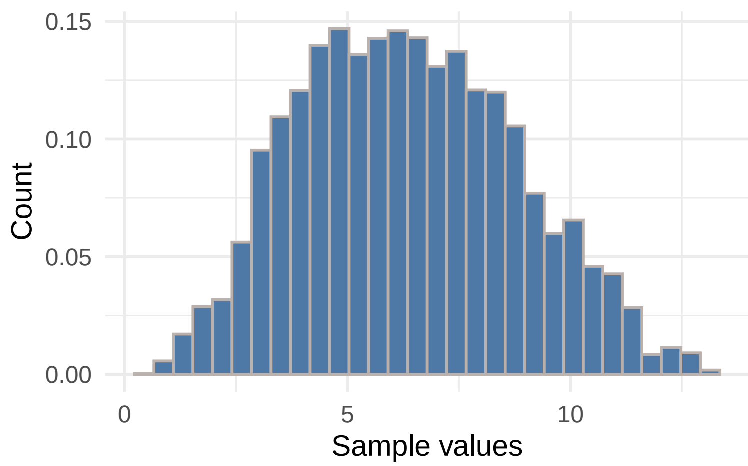

Figure 7: Histogram of sample with Uniform proposed distribution

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	7.088860
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, using the Normal and Uniform proposal distributions provides a reliable estimation of $E(X)$. However, the Chi-square proposal distribution results in a slightly less accurate estimate, likely due to its lower acceptance rate and larger step variability.

Appendix

```
library(ggplot2)

# define target function
f <- function(x){

  if(x > 0){
    120*(x^5)*exp(-x)

  } else {
    stop("x must be positive")

  }
}

MetropolisHastings <- function(x_init, n_iter, proposal_dist = "Other"){
  set.seed(1234)
  x <- numeric(length = n_iter)
  x[1] <- x_init
  acc <- 0

  for(t in 2:(n_iter)){

    # sample candidate for x* from a proposal distribution
    if(proposal_dist == "Normal"){
      x_candidate <- rnorm(n = 1, mean = x[t - 1], sd = 0.1)

    } else if(proposal_dist == "Chi"){
```

```r
      x_candidate <- rchisq(n = 1, floor(x[t - 1] + 1))

    } else {
      x_candidate <- runif(n = 1, min = x[t - 1] - 0.5, max = x[t - 1] + 0.5)
    }

    # if Chisq
    if(proposal_dist == "Chi"){
      num <- f(x_candidate) * dchisq(x[t - 1], df = floor(x_candidate))
      denom <- f(x[t - 1]) * dchisq(x_candidate, df = floor(x[t - 1]))
      r <- num / denom
    } else{
      # calculate acceptance rate
      r <- f(x_candidate) / f(x[t - 1]) # since proposal is symmetric, R = f(x*)/f(x_t)
    }

    # sample 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))
}

x1 <- MetropolisHastings(x_init = 1, n_iter = 10000, proposal_dist = "Normal")$chain
x2 <- MetropolisHastings(x_init = 1, n_iter = 10000, proposal_dist = "Chi")$chain
x3 <- MetropolisHastings(x_init = 1, n_iter = 10000)$chain

acc1 <- MetropolisHastings(x_init = 1, n_iter = 10000, proposal_dist = "Normal")$acceptance_rate
acc2 <- MetropolisHastings(x_init = 1, n_iter = 10000, proposal_dist = "Chi")$acceptance_rate
acc3 <- MetropolisHastings(x_init = 1, n_iter = 10000)$acceptance_rate

data <- data.frame(it = 1:10000, x1 = x1, x2 = x2, x3 = x3)

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") +
  theme_minimal() +
  labs(x = "Sample values", y = "Count")

ggplot(data, aes(x = it, y = x2)) +
  geom_line(col = "#4E79A7") +
```

```r
  theme_minimal() +
  labs(x = "Iteration", y = "Chain values")

ggplot(data, aes(x = x2)) +
  geom_histogram(aes(y = ..density..), color = "#BAB0AC", fill = "#4E79A7") +
  theme_minimal() +
  labs(x = "Sample values", y = "Count") +
  geom_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") +
  theme_minimal() +
  labs(x = "Sample values", y = "Count") +
  geom_density()

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