Compound Poisson Process: Theory, Distribution, Sensitivity and R Shiny Implementation

Abstract

This document presents the theoretical derivation, distributional properties, and sensitivity analysis of a Compound Poisson Process with exponential interarrival times and exponential jump sizes. The complete R Shiny code for interactive simulation is also included.

1 Introduction

A Compound Poisson Process is a fundamental stochastic model widely used in insurance, finance, queueing systems, and reliability engineering. It describes the cumulative impact of random events occurring over time, where each event contributes a random size.

We assume:

- Interarrival times follow an exponential distribution with rate λ.
- Jump sizes X_i are i.i.d. exponential with parameter μ.

2 Theory of the Compound Poisson Process

Let $N(t)$ be a Poisson process with rate λ. Define the compound process:

$$N(t)$$

$$S(t) = \sum_{i=1} X_i$$

where X_i are independent exponential random variables with rate μ, independent of $N(t)$.

Distribution of $S(t)$

Since

$$N(t) \sim \text{Poisson}(\lambda t),$$

we have:

$$P(N(t) = n) = e^{-\lambda t} \frac{(\lambda t)^n}{n!}.$$

Conditional on $N(t) = n$,

$$S(t) \mid N(t) = n \sim \text{Gamma}(n, \mu)$$

because the sum of n i.i.d. exponential(μ) random variables is Gamma distributed.

Thus, $S(t)$ is a Poisson-Gamma mixture:

$$P(S(t) \leq s) = \sum_{n=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} F_{\Gamma(n,\mu)}(s).$$

Mean and Variance

$$\mathbb{E}[S(t)] = \lambda t \cdot \frac{1}{\mu},$$

$$\text{Var}[S(t)] = 2\frac{\lambda t}{\mu^2}.$$

3 Parameter Sensitivity

Effect of λ

- Higher λ: more frequent jumps.

- $S(t)$ increases faster.
- Higher variance.

Effect of μ

- Higher μ: smaller mean jump size.
- Smaller μ: larger jump sizes and heavier tail.

These behaviors are visualized interactively through the R Shiny app.

4 Full R Shiny Code

R Script

```
install.packages("shiny")
################################################################
# COMPLETE R SHINY APP FOR COMPOUND POISSON-EXPONENTIAL
################################################################

library(shiny)

# Function to simulate S(t)
simulate_S_t <- function(lambda, mu, t, n_sim = 5000) {
  S_t <- numeric(n_sim)

  for (i in 1:n_sim) {
    N <- rpois(1, lambda * t)
    if (N > 0) {
      S_t[i] <- sum(rexp(N, rate = mu))
```

```
    } else {
      S_t[i] <- 0
    }
  }
  return(S_t)
}

################################################################
# UI
################################################################

ui <- fluidPage(
  titlePanel("Compound Poisson–Exponential Process: S(t)"),

  sidebarLayout(
    sidebarPanel(
      h4("Parameters"),

      sliderInput("lambda",
                  "Interarrival rate λ:",
                  min = 0.1, max = 2, value = 0.5, step = 0.1),

      sliderInput("mu",
                  "Rate of Xᵢ (μ):",
                  min = 0.1, max = 2, value = 1, step = 0.1),

      sliderInput("time",
                  "Time t:",
                  min = 1, max = 10000, value = 10, step = 1),
```

```
    numericInput("n_sim",
                 "Number of simulations:",
                 value = 5000, min = 1000, max = 50000),

    actionButton("simulate", "Run Simulation", class = "btn-primary")
   ),

   mainPanel(
    h3("Histogram of S(t)"),
    plotOutput("histPlot"),
    hr(),
    h3("Summary Statistics"),
    verbatimTextOutput("summaryStats"),
    hr(),
    h3("Compare S(t) at Fixed Times"),
    plotOutput("multiHist")
   )
  )
)

############################################################
# SERVER
############################################################

server <- function(input, output) {

  # Run simulation when button is clicked
  data_sim <- eventReactive(input$simulate, {
   simulate_S_t(input$lambda, input$mu, input$time, input$n_sim)
  })
```

```
# Histogram for chosen t
output$histPlot <- renderPlot({
  req(data_sim())
  hist(data_sim(),
       breaks = 50,
       main = paste("Histogram of S(t) at t =", input$time),
       xlab = "S(t)",
       col = "skyblue",
       border = "white",
       freq = TRUE)
})

# Summary
output$summaryStats <- renderPrint({
  s <- data_sim()
  cat("Mean of S(t):", mean(s), "\n")
  cat("Variance of S(t):", var(s), "\n")
  cat("\nSummary:\n")
  print(summary(s))
})

# Comparison histograms (t = 10, 100, 1000, 10000)
output$multiHist <- renderPlot({
  par(mfrow = c(2,2))
  times <- c(10,100,1000,10000)

  for (t in times) {
    S_t <- simulate_S_t(input$lambda, input$mu, t, 3000)
    hist(S_t, breaks = 40,
```

```
        main = paste("S(t) at t =", t),
        xlab = "S(t)", col = "red", border = "white")
    }
  })
}

#################################################################
# Run Shiny App
#################################################################

shinyApp(ui, server)
```

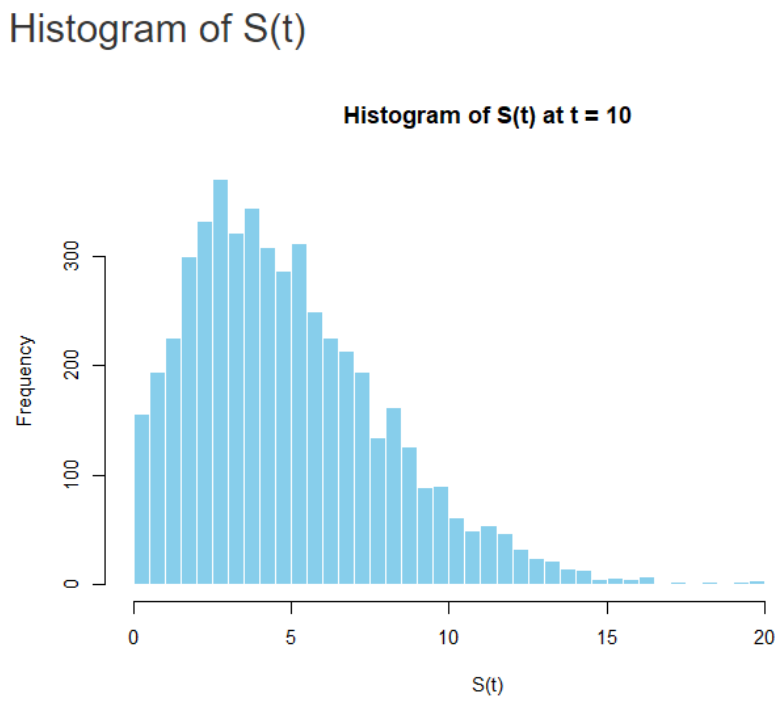

Figure 1: Histogram of Exponential Jump Sizes X_i

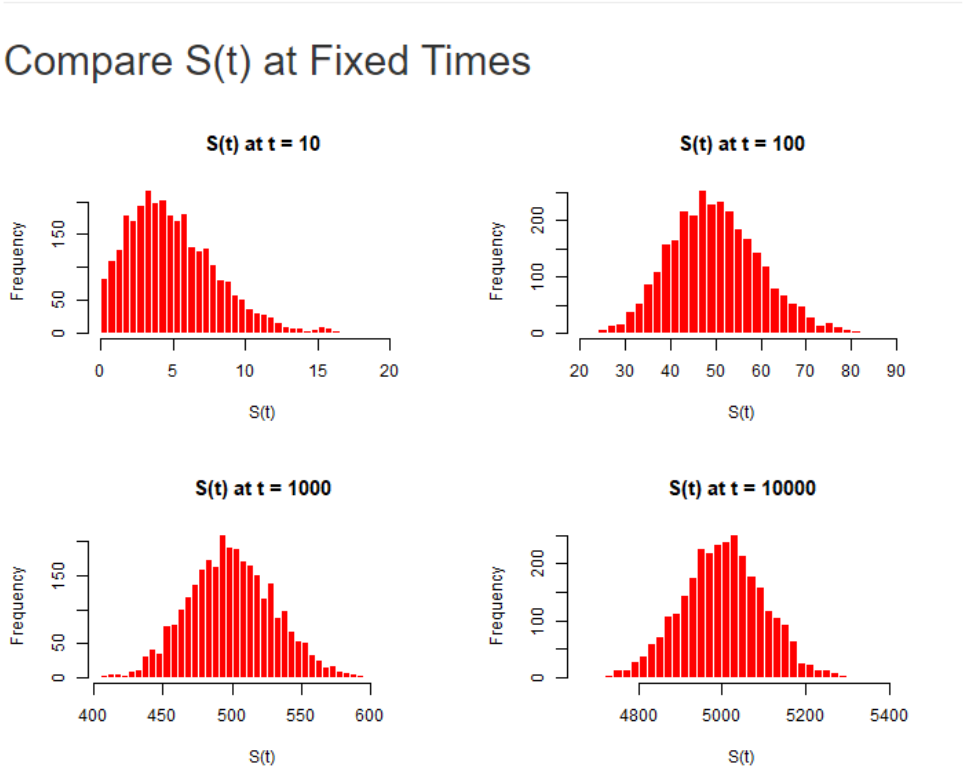

Figure 2: Step Path of Compound Poisson Process $S(t)$

5 Conclusion

This document described the compound Poisson process with exponential jump sizes, derived its distribution, analyzed parameter sensitivity, and provided a complete R Shiny implementation to simulate and visualize the process.