The Chinese University of Hong Kong, Shenzhen
DDA4002 2024-2025 Spring

Solutions to Assignment 2

Yinqi (Marcus) Zhong
121090847

March 5, 2025

1 Theoretic Questions

1.1 Embedded Chain

#1 **(Embedded Chain)** Consider a CTMC with 3 states and transition rate matrix

$$Q = \begin{pmatrix} -10 & 5 & 5 \\ 2 & -5 & 3 \\ 4 & 1 & -5 \end{pmatrix}.$$

Specify the transition probability matrix of its embedded chain and the exponential clocks corresponding to each state.

Figure 1: Q1 Description

In a CTMC, the embedded chain is formed by considering the rates of transition between states without worrying about the time spent in each state.

The diagonal elements of the transition matrix Q (Q_{ii}) are the rate at which the process leaves state i. And the off-diagonal elements Q_{ij} represent the transition rates from state i to j.

The transition probabilities matrices is calculated by normalizing the transition rates:

$$P_{ij} = -\frac{Q_{ij}}{Q_{ii}} \text{ , for i} \neq \text{j}$$

$$P_{ii} = 0$$

So, the transition probability matrix P of the embedded chain would be:

$$P = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.4 & 0 & 0.6 \\ 0.8 & 0.2 & 0 \end{bmatrix}$$

The exponential clocks are $exp(10)$, $exp(5)$, and $exp(5)$ for state 0,1,2.

2 Numeric Experiments

2.1 Copulas

a

The Guassian Copula is induced by two **standard normal random variables**, to generate $i.i.d$ samples of (U_1, U_2) Guassian Copula, the general algorithm would be:

1. Use Acceptance-Rejection Method to generate uniform random variables
2. Generate the bivariate normal (W_1, W_2) using **Cholesky method**
3. Compute the CDF $U_i = \Phi(W_i)$.
4. Return $X_i = F_i^{-1}(U_i), i = 1, ..., n$

b

The output of simulating using the given $\rho \in \{\pm 0.8, \pm 0.6, \pm 0.4, \pm 0.2\}$ is listed as:

The estimated coveriance under correlation coefficient -0.8 is -0.064645.

The estimated coveriance under correlation coefficient -0.6 is -0.04867.

The estimated coveriance under correlation coefficient -0.4 is -0.0324.

The estimated coveriance under correlation coefficient -0.2 is -0.015664.

The estimated coveriance under correlation coefficient 0.2 is 0.015444.

The estimated coveriance under correlation coefficient 0.4 is 0.032344.

The estimated coveriance under correlation coefficient 0.6 is 0.047936.

The estimated coveriance under correlation coefficient 0.8 is 0.065289.

The Scatter Plot for 100 samples from Copula with $\rho \in \{\pm 0.6, 0\}$ is:

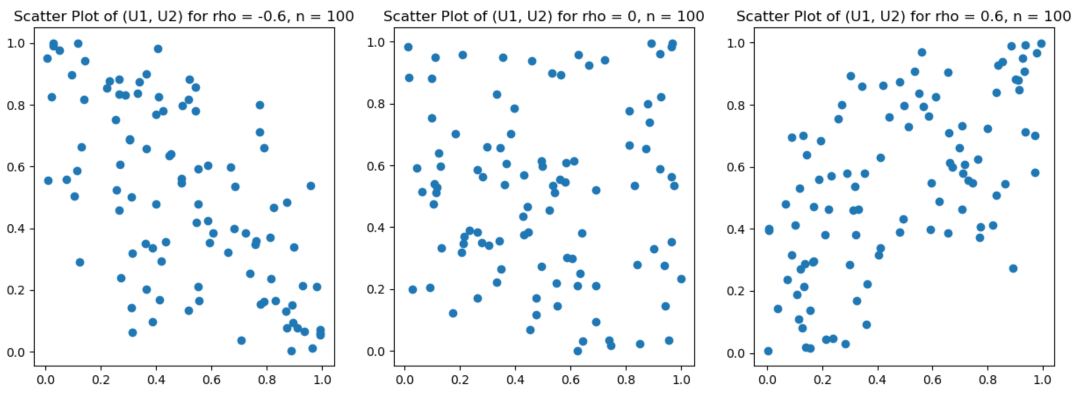

Figure 2: Scatter Plot for 100 samples

c

To generate dependent exponential random variables (X_1, X_2), we may directly use the Copula Normal Function to generate U, and then get the exponential X by applying its inverse CDF.

The general algorithm would be:

1. Use the Gaussian Copula to generate bivariate uniform distributed variables (U_1, U_2)
2. Compute the CDF $X_i = \Phi(U_i)$.
3. Return $X_i = F_i^{-1}(U_i) = -\ln(1 - U_i), i = 1, ..., n$

d

The output of simulating using the given $\rho \in \{\pm 0.8, \pm 0.6, \pm 0.4, \pm 0.2\}$ is listed as:

The estimated coveriance under correlation coefficient -0.8 is -0.536889.

The estimated coveriance under correlation coefficient -0.6 is -0.438738.

The estimated coveriance under correlation coefficient -0.4 is -0.304481.

The estimated coveriance under correlation coefficient -0.2 is -0.162765.

The estimated coveriance under correlation coefficient 0.2 is 0.175175.

The estimated coveriance under correlation coefficient 0.4 is 0.336091.

The estimated coveriance under correlation coefficient 0.6 is 0.558405.

The estimated coveriance under correlation coefficient 0.8 is 0.733588.

And the corresponding Scatter plot using 100 samples would be:

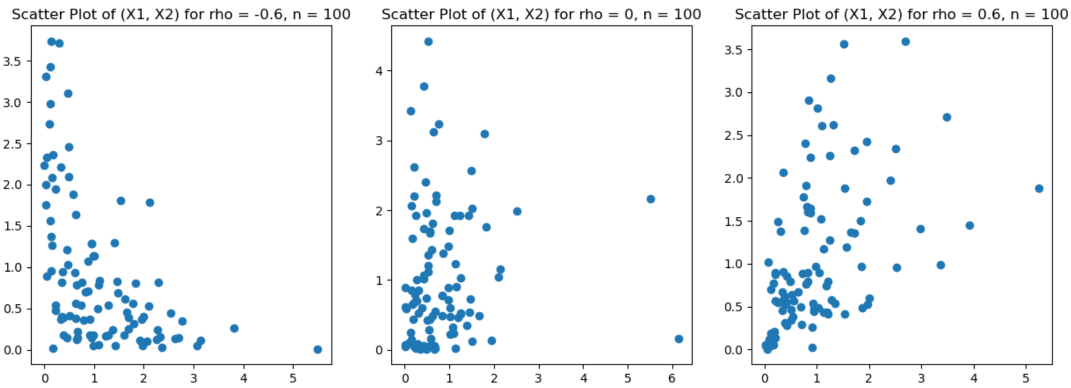

Figure 3: Scatter Plot for exponential 100 samples

2.2 Poisson Processes

i

Method1: Generate **homogeneous Poisson process** via inter-arrival times. General Algorithm:

1. Generate the first inter-arrival time from $exp(\frac{1}{rate})$, in this problem, rate = 5
2. Continue generating the subsequent inter-arrival time, until its cumulated time exceeds the upper bound T
3. The total arrivals should follow a Poisson Distribution with mean : **rate * T**. In this problem, it would be $Poisson(10)$

The visualization of the poisson process is as below. As we can see, the event arrives at random time.

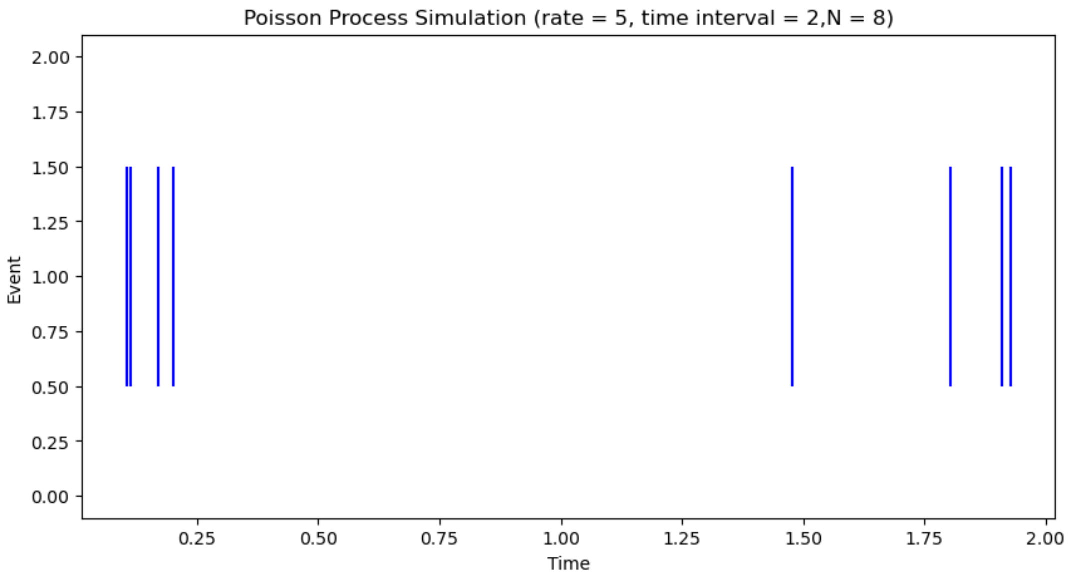

Figure 4: Visualization of Method 1

As can be easily seen in the following two figures, the simulated total number of arrivals follows a Poisson distribution, given the time interval = 2, rate = 5, the distribution should be $Poisson(5 * 2) = Poisson(10)$.

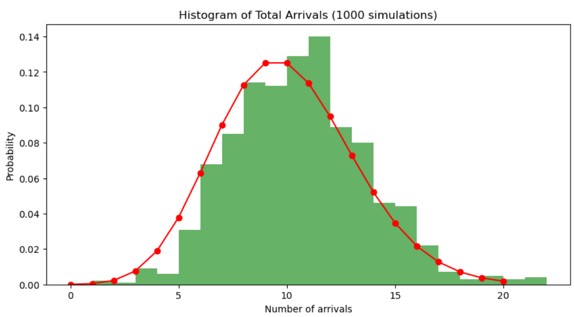

Figure 5: Simulating with 1000 samples

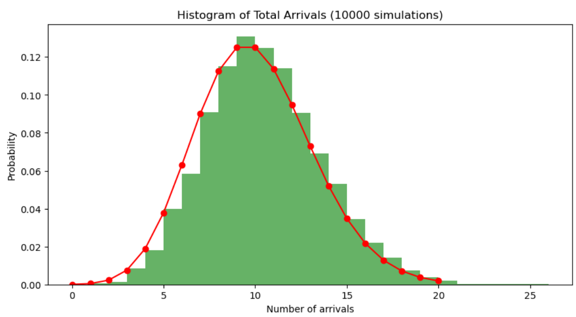

Figure 6: Simulating with 10000 samples

ii

The general algorithm for generating poisson process by first generating the total number of arrivals and then distributing each arrival uniformly on the time interval would be:

1. First, we sample total arrival times following Poisson(rate * time interval)
2. Then, Generate N uniform(0,time interval) randomly uniformly distributed
3. sort these times to get ordered arrival times, and to see whether it follows an exponential distribution with rate equals to rate given.

The visualization of the poisson process is as below. As we can see, the event arrives at random time.

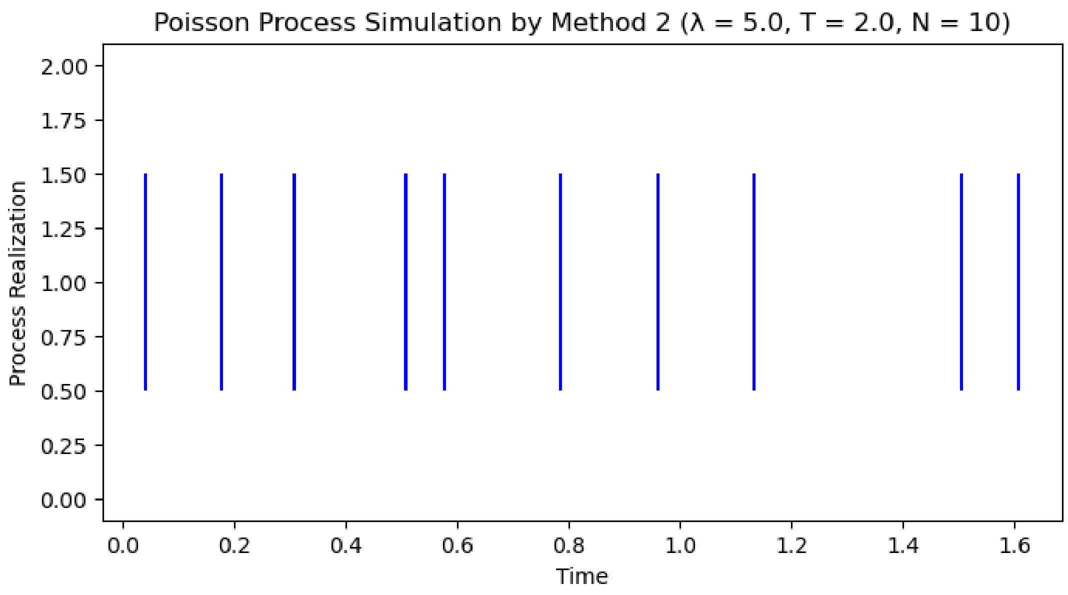

Figure 7: Visualization of Method 2

Verifying using exponential(-λ), which is closely matched.

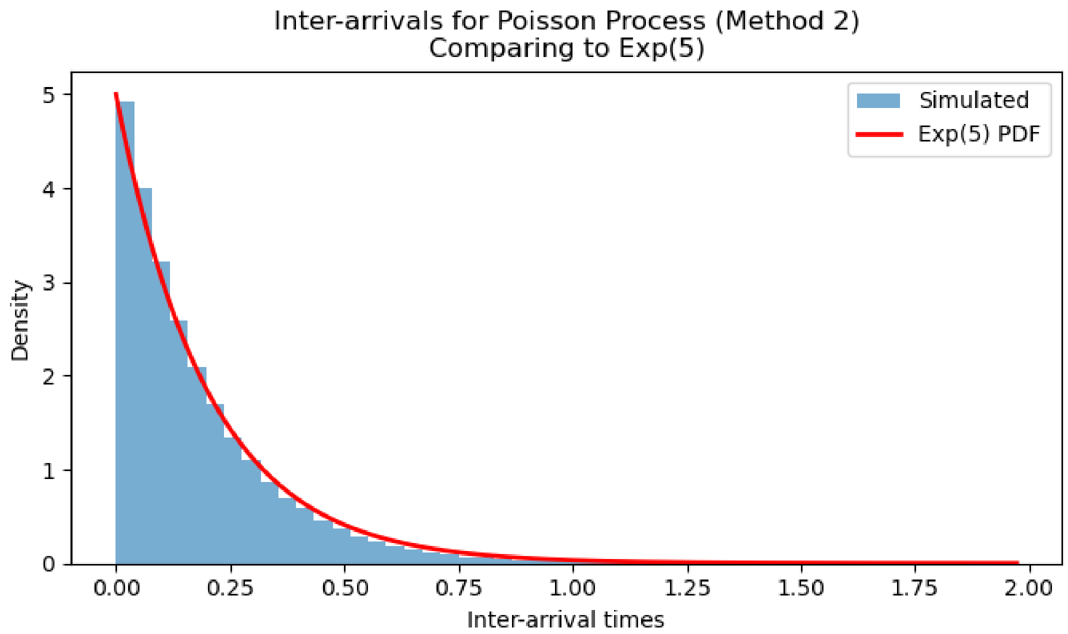

Figure 8: Inter-arrivals for Poisson Process (Method 2) Comparing to Exp(-λ)

iii

Method2 needs to be modified to generate a non-homogeneous process (NHPP) with a time varying rate $\lambda(t)$.

The total number of arrivals still follows a Poisson distribution with mean $\Lambda(t) = \int_0^t \lambda(s)\,ds$ in [0,T].

The arrivals are not placed and ordered in to a uniform distribution, instead, it follow a distribution with CDF:

$$F(t) = \frac{\Lambda(t)}{\Lambda(T)}, \text{ where } \Lambda(t) = \int_0^t \lambda(s)\, ds$$

So, by Inverse Transform Method, the arrival times can be sampled from $T_i = \Lambda^{-1}(U \cdot \Lambda(T))$

The complete pesudo code for this problem would be:

STEP1: Compute $\Lambda(T) = \int_0^T \lambda(s)\, ds$ in [0,T]

STEP2: Generate N following $Poisson(\Lambda(T))$.

STEP3: For i in 1,...,N:

1. Generate U_i ~$Uniform(0, 1)$.
2. Compute arrival time $T_i = \Lambda^{-1}(U \cdot \Lambda(T))$

STEP4: Sort the arrival times in ascending order.

return: The sorted set of arrival times $T_1, T_2, ..., T_N$