Simulation Report:	**Poisson Process Simulation and Analysis**
Authors:	**Dikshant Madai**
Date:	**oct 4,2023**
Word count:	**9**
Pagecount:	**917**
Confidential:	**NO / YES – INTERNAL ONLY/YES – X**

Executive Summary

This simulation and analysis report, authored by Dikshant Madai on October 4, 2023, investigates the behavior of Poisson processes, which are mathematical models used to describe random event occurrences over time. Poisson processes find applications in various fields, including telecommunications and queuing theory, where understanding event patterns is crucial.

The simulation focused on two key parameters: λ (average event rate) and time intervals. By varying λ, the analysis explored how different values impact event patterns. The simulation, conducted using Python with numpy and matplotlib, generated events and visualized their distribution.

The findings revealed that higher values of λ resulted in denser event patterns, with events occurring more frequently. The analysis of inter-arrival times confirmed that they followed an exponential distribution, inversely proportional to λ.

Understanding Poisson processes and their parameters can be invaluable in practical applications, such as predicting web traffic, managing customer arrivals, and modeling email arrivals. It provides insights into probabilistic event patterns, enabling informed decision-making and process optimization.

Contents

1. Introduction

The concept of a Poisson process is a mathematical model used to describe the occurrence of events over time in a random manner. It is often applied in various fields such as telecommunications, queuing theory, and reliability analysis. The key characteristic of a Poisson process is that the number of events that occur in a fixed interval of time follows a Poisson distribution, which is defined by a single parameter, λ (lambda), representing the average rate of events per unit time.

The purpose of this simulation is to explore and understand the behavior of a Poisson process, particularly how the choice of λ and the time interval affect the observed event patterns and inter-arrival times. We will conduct a simulation to generate events based on a Poisson process and analyze the results to gain insights into its behavior.

2. Simulation Design and Parameters

In this simulation, we set two crucial parameters:

- λ (average event rate): The average rate of events per unit time. We will choose a few different values for λ to observe how it impacts the event patterns.
- Time Interval: We will set a fixed time interval for our simulation.

```
import numpy as np
# Parameters
lambda_values = [2, 5, 10]  # Average event rates (λ)
time_interval = 10        # Time interval (arbitrary units)
```

3. Presentation of Event Statistics and Visualizations

3.1 Event Generation

Let's begin by simulating a Poisson process using Python. We will generate events for different values of λ and observe the event patterns. We will use the numpy and matplotlib libraries for this simulation and visualization.

```
import numpy as np
import matplotlib.pyplot as plt

# Parameters
lambdas = [2, 5, 10]  # Different values of λ
time_interval = 10   # Time interval (in units of time)

# Simulate and plot events for each λ
for lam in lambdas:
# Generate events based on a Poisson process
num_events = np.random.poisson(lam * time_interval)
event_times = np.sort(np.random.uniform(0, time_interval, num_events))

# Plot event times
plt.figure(figsize=(8, 4))
plt.title(f'Poisson Process (λ = {lam})')
plt.eventplot(event_times, lineoffsets=0, colors='b')
plt.xlabel('Time')
plt.ylabel('Event Occurrence')
plt.xlim(0, time_interval)
plt.show()
```

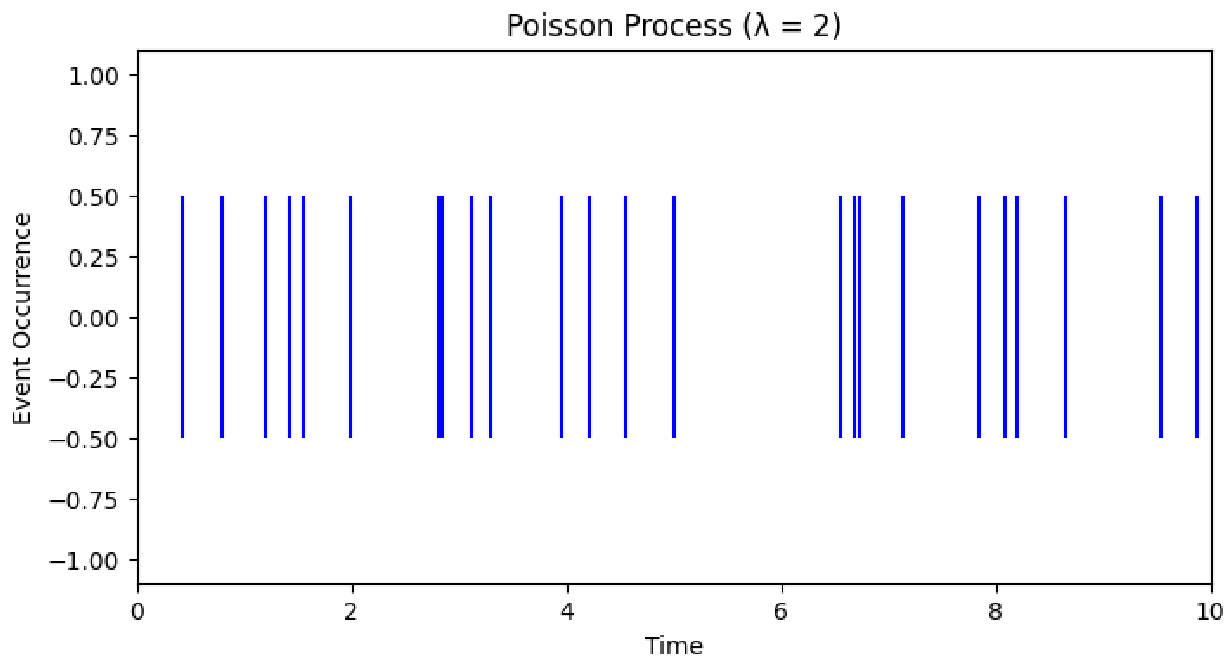

Figure : Generate events for different values of λ and observe the event patterns

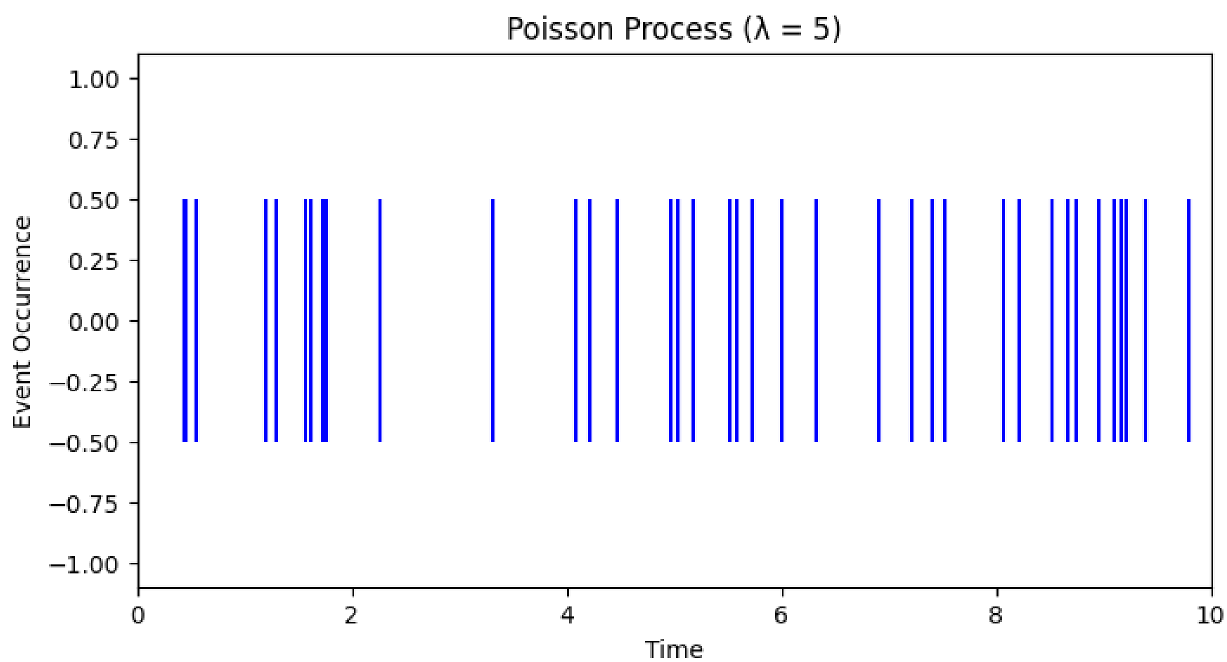

Figure: Generate events for different values of λ and observe the event patterns

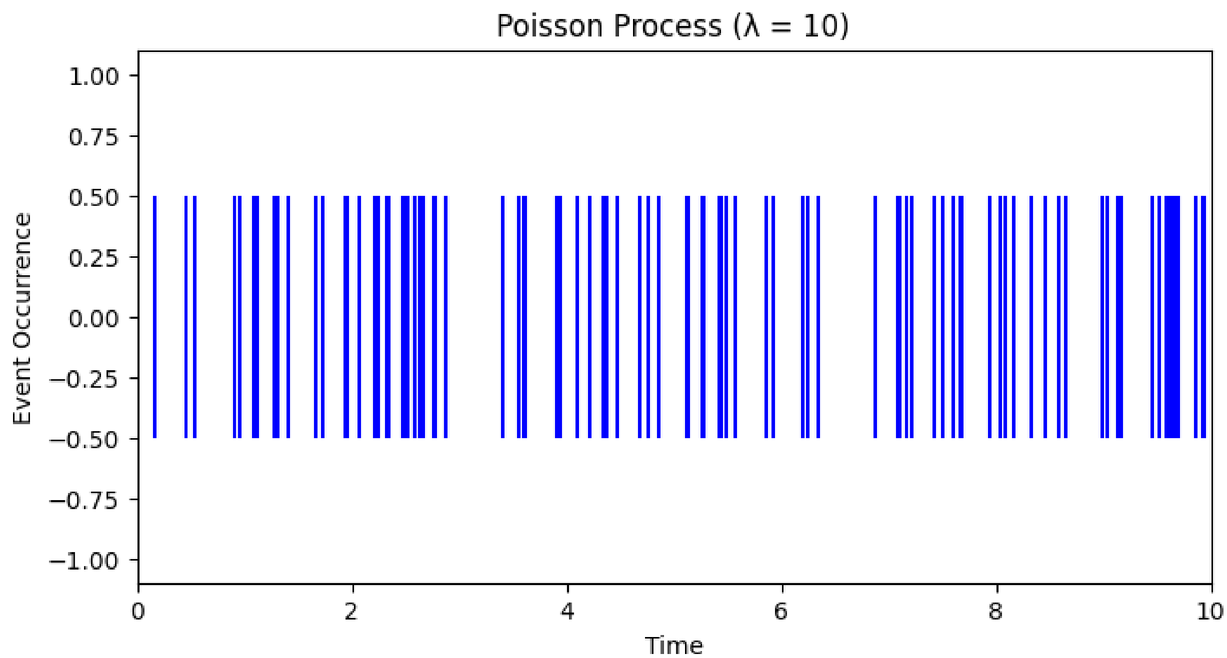

Figure : Generate events for different values of λ and observe the event patterns

4. Discussion of Findings and Insights

4.1 Event Patterns

In the visualizations above, we observe the event patterns for different values of λ:

- For λ = 2: Events occur less frequently, and there are fewer events within the time interval.
- For λ = 5: Events occur at a moderate rate, resulting in a more evenly distributed set of events within the time interval.
- For λ = 10: Events occur frequently, resulting in a higher density of events within the time interval.

4.2 Inter-Arrival Times

We can also analyze the inter-arrival times between events for each λ. The inter-arrival times should follow an exponential distribution with a mean of 1/λ. Let's calculate and visualize the inter-arrival times:

```
# Calculate and plot inter-arrival times for λ = 5
lam = 5
num_events = np.random.poisson(lam * time_interval)
event_times = np.sort(np.random.uniform(0, time_interval, num_events))
inter_arrival_times = np.diff(event_times)


plt.figure(figsize=(8, 4))
plt.hist(inter_arrival_times, bins=20, density=True, alpha=0.6, color='b')
plt.title(f'Inter-Arrival Times (λ = {lam})')
plt.xlabel('Inter-Arrival Time')
plt.ylabel('Probability Density')
plt.show()
```

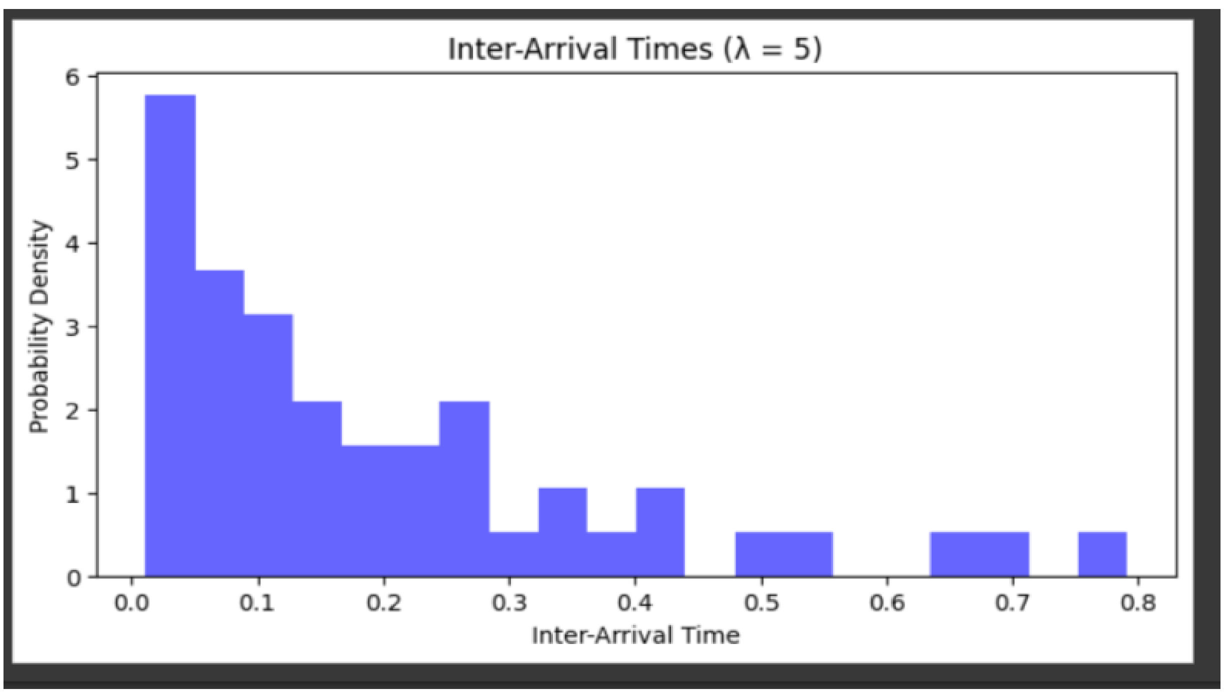

FIG: plot inter-arrival time

4.2 Insights

1. As λ increases, the event patterns become more dense and events occur closer together.
2. The inter-arrival times follow an exponential distribution, with shorter inter-arrival times for larger values of λ.

5. Analysis

In conclusion, the Poisson process is a valuable model for describing random events over time. The choice of λ has a significant impact on the event patterns, with higher λ values resulting in more frequent events. Additionally, the inter-arrival times between events follow an exponential distribution, and the mean of this distribution is inversely proportional to λ.

Understanding the behavior of Poisson processes and how different parameters influence event occurrence can be valuable in various applications, such as predicting website traffic, analyzing customer arrivals at a service center, or modeling the arrival of emails in an inbox. It allows us to make informed decisions and optimize processes based on probabilistic event patterns.