MAL 2010
Group Assignment

An Efficient Python Approach for Simulation of Poisson Distribution

Group Members-
Hritin Raj (B22ES015)
Pranav Mukesh Bhansali (B22ES003)
Prateek Kumar (B22ES024)
Samriddhi Jain (B22ES008)
Soumen Kumar (B22ES006)

Abstract

Objective: The paper aimed to develop a foundational understanding of probability by delving into its core mathematical features and exploring its practical aspects in specific scenarios.

Distribution of Probability: In the paper, probability distributions based on the inherent nature of variables are defined. We showcase probability through binomial or Poisson distributions, particularly in discrete variables.

Types of Distributions: The paper provides a concise yet comprehensive overview of probability distributions supported by illustrative examples.

Poisson Distribution: The paper exploits the Poisson distribution, a discrete model mostly used to distribute count data within specific time intervals. The paper includes practical instances, such as predicting the frequency of traffic accidents or the number of received phone calls.

Obtaining or Estimating Poisson Distribution: Every example in the paper engages in a detailed discussion on methodologies and approaches for effectively getting or estimating the Poisson distribution, enhancing the practical applicability of the concept.

Applications Using Python Libraries: The paper provides an insightful picture of real-world applications that harness the capabilities of Python libraries.

Conclusion: In short, the paper explores the concepts of probability and Poisson distribution, emphasizing the practical relevance and applications in various domains.

Introduction

Statistical Properties of Poisson Distribution:

- The properties are complementary and full, possessing enough figures.
- Exponential structure allows for the construction of "exact" conditional tests.

Exploration of Properties:

- The paper focuses on exploring the effective utilization of Poisson distribution properties.
- Part I involves deriving Poisson conditional types based on variance and cumulants of samples.
- Left-truncated samples are considered, expanding the 'variance' test.

Significance Checks:

- Section II discusses significance checks for cross-product fractions of Poisson averages.
- Biomedical science applications are elucidated with realistic examples.

Applications in Biological Models:

- Poisson distribution is a crucial component of models for testing biological phenomena.
- It's employed to analyze the occurrence of events over time, including those deemed to occur spontaneously.

Additional Statistical Properties:

- Poisson distribution is additive, comprising independent Poisson variates.
- When collected from a Poisson distribution, the average is a suitable estimate.
- The completeness property is highlighted, wherein all other cumulants are equal to the average.

Conditional Relevance Metrics:

- Sophisticated Poisson properties are used to derive contingent methods for evaluating count outcomes.
- Completeness property is utilized to extract skewness and kurtosis-related moments from variance test statistics.

Applications Beyond Basic System:

- The Poisson average is considered an exponential function of associated inputs in the basic system.
- Relevant statistics are derived, and "exact" conditional relevance metrics are calculated in certain cases.

Freelancing Income Exploration:

- There has been a shift in focus to freelancing income, estimated at around $80,000 annually for the past decade.
- Aspirations to achieve a six-figure income are prompting an exploration of the likelihood of this accomplishment.

Unpredictable Occurrence Scenarios:

- There are also situations where organizations determine the uncertainty of subset occurrence larger or smaller than a known rate in the future.
- Examples include preparing for special occasions like Black Fridays or Cyber Mondays based on average sales predictions.

Using Poisson Distribution:

- The paper discusses the theory behind the distribution of Poisson, which is used to design situations as above, and how to recognize and use its formula.

Python Code Simulation:

- The paper gives insight into using Python code to simulate the Poisson distribution.
- It also presents the importance of simulation in practical applications.

Paper Structure:

- The paper is divided into six components, with the second element focusing on discrete probability distributions.
- Section 3 explains the probability mass function.
- Section 4 delves into Poisson distribution and its explanation with Python.
- Section 5 involves simulating the Poisson distribution.
- Section 6 concludes the article.

Discrete Probability Distribution

Definition: Probability distributions are mathematical functions that describe the likelihood of different outcomes in a sample space. They provide a systematic way to model and analyze uncertainty and randomness in various phenomena.

Types of Probability Distributions:

There are two main types of probability distributions: discrete and continuous.

Discrete Probability Distributions:

- Deal with countable outcomes, typically associated with random experiments.
- Examples include the binomial, Poisson, Bernoulli, and multinomial distributions.
- Used when outcomes are distinct and separate, often involving integers.
- The probability mass function (PMF) is used to describe the probabilities of specific outcomes.

Continuous Probability Distributions:

- Deal with uncountable outcomes and are typically associated with measurements.
- Examples include the normal (Gaussian), uniform, exponential, and beta distributions.
- Used when outcomes can take any value within a range.
- The probability density function (PDF) is used to describe the probabilities over a continuous range.

Discrete Probability Distributions: A Deeper Dive

1. Binomial Distribution:

- Models the number of successes in a fixed number of independent Bernoulli trials.
- Parameters: number of trials (n) and probability of success (p).
- Example: Tossing a coin multiple times and counting the number of heads.

2. Poisson Distribution:

- Models the number of events occurring in a fixed interval of time or space.
- Parameter: average rate of occurrence (λ).
- Example: Counting the number of emails received in an hour.

3. Bernoulli Distribution:

- Represents a binary outcome (success/failure) in a single experiment.
- Parameter: probability of success (p).
- Example: Flipping a coin and observing whether it lands heads or tails.

4. Multinomial Distribution:

- Generalization of the binomial distribution for more than two categories.
- Parameters: number of trials (n) and probabilities of each category.
- Example: Rolling a die with multiple outcomes.

Applications of Discrete Probability Distributions:

Inventory Management:

Binomial and Poisson distributions help analyze inventory levels and make informed restocking decisions.

Biomedical Sciences:

Poisson distribution is used to model the number of occurrences of a specific event in a given time period.

Quality Control:

The binomial distribution is applied in quality control scenarios, such as assessing the number of defective items in a sample.

Probability Mass Function

Definition: The Probability Mass Function (PMF) is a fundamental concept in probability theory and statistics. It describes the probability distribution of a discrete random variable. In simpler terms, the PMF assigns probabilities to each possible outcome of a discrete random variable.

Key Characteristics

1. Discreteness:

- The random variable must have distinct, countable outcomes.
- Typically used for scenarios where outcomes can be enumerated, such as rolling a die or counting the number of heads in coin flips.

2. Probability Assignment:

- The PMF provides the probability of each possible outcome.
- Mathematically denoted as P(X = x), where (X) is the random variable and (x) is a specific outcome.

3. Sum Rule:

- The sum of probabilities across all possible outcomes is equal to 1.
- (sum P(X = x) = 1)

Example:

For a six-sided die, the PMF might be:(P(X = 1) = frac{1}{6}),(P(X = 2) = frac{1}{6}), and so on.

Poisson Probability Mass Function (PMF):

Overview:

The Poisson distribution is a discrete probability distribution that describes the number of events occurring within a fixed interval of time or space. The PMF for the Poisson distribution is defined as:

[P(X = k) = frac{e^{lambda} * lambda^k}/{k!}]

Where:

- (k) is the number of events.
- (lambda) is the average rate of events per interval.
- (e) is the mathematical constant approximately equal to 2.71828.

Key Components:

1. e Factor:

- The term (e^{lambda}) is a normalization factor ensuring that the probabilities sum to 1

2. Average Rate (lambda):

- Represents the average number of events occurring in the specified interval.
- Determines the shape and location of the Poisson distribution.

3. Probability of (k) Events:

- (P(X = k)) calculates the probability of observing exactly (k) events in the interval.

4. Factorial Term (k!):

- Accounts for the number of ways (k) events can occur in a specific order.

Example:

If (lambda = 3) (average rate of 3 events per interval), the Poisson PMF for observing exactly 2 events (k = 2)) is:

P(X = 2) = frac[{e^(-3)*(3^2)},(2!)]

Applications:

- **Traffic Flow**: Modeling the number of accidents in a given time period.
- **Biomedical Sciences**: Analyzing the occurrence of rare diseases in a specific population.
- **Call Centers**: Estimating the number of incoming calls during a particular hour.

Case Study: New-Born Babies in a Hospital

The case study introduces the concept of a stochastic process, specifically using the example of observing newborn babies in a hospital. A stochastic process is a random process that represents the evolution of a system over time, where the outcome is not entirely predictable.

In this scenario, the focus is on Poisson distribution, which is a discrete probability distribution often used to model the number of events occurring in a fixed interval of time or space. The Poisson distribution is suitable for this context because it satisfies certain conditions, as mentioned in the explanation:

- Known Rate of Events: The average rate of newborn babies is given as 6 per hour.
- Events Occur Independently: The birth of one baby does not affect the timing of the next birth.
- Constant Rate Over Time: The average number of babies per hour remains the same over time.
- No Simultaneous Events: Two babies are not born at exactly the same instant.

The case then discusses the implications of the Poisson distribution in practical scenarios, especially in business applications. Given the average rate, companies can use the Poisson distribution to predict the number of transactions, customers, or events within a specific time frame.

The Poisson Probability Mass Function (PMF) is introduced, represented by the formula:

$$P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

where
x = 0, 1, 2, 3, …
λ = mean number of occurrences in the interval
e = Euler's constant $\approx$ 2.71828

The case then calculates the probability of seeing 10 newborn babies in an hour, given the average rate (λ=6). The calculation results in a **~ 4%** chance of observing 10 babies.

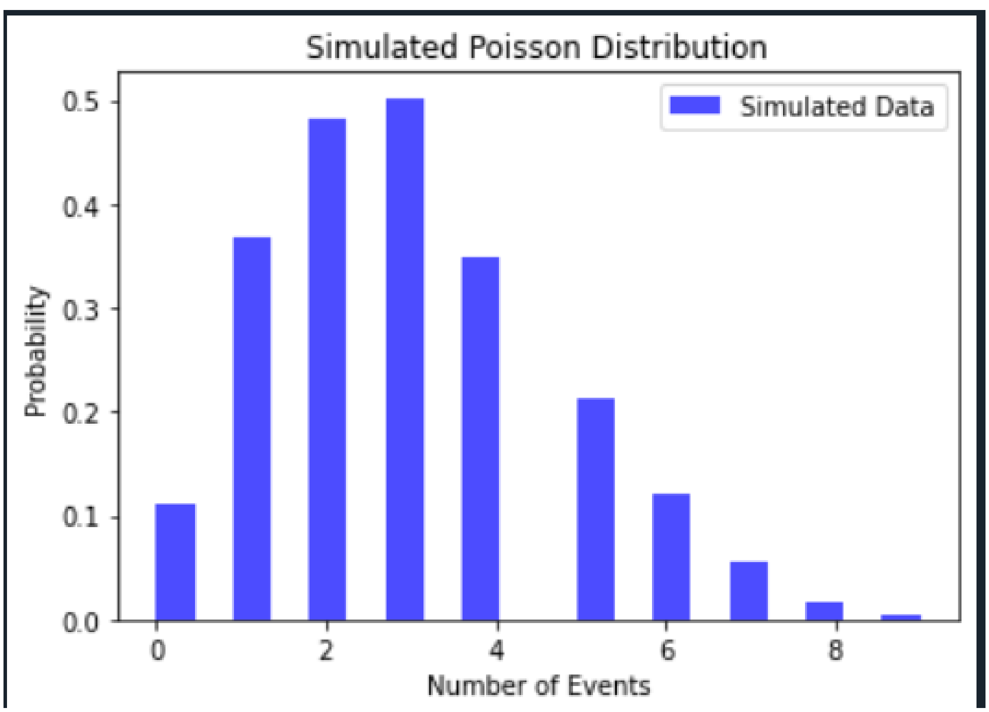
The case study also acknowledges some practical considerations. Even though there is a known average rate, the timing of events can be random. In the context of newborn babies, this means that there might be instances of back-to-back births or longer waiting times between births. Additionally, the average rate (λ) may not always be constant in practice, but for the purposes of this case study, it's assumed to be constant enough to use the Poisson distribution as an approximation.

In summary, the case study illustrates the application of the Poisson distribution in modeling random events, using the example of observing new-born babies in a hospital, and discusses its relevance in making predictions and decisions in various business scenarios

Poisson Distribution with Python and its Simulation

A Python code for simulating Poisson distribution-

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

# Setting the seed for reproducibility
np.random.seed(42)

lambda_value = 3  # Average rate of occurrence

poisson_data = np.random.poisson(lambda_value, size=1000)

# Step 3: Visualize the simulated data
plt.hist(poisson_data, bins=20, density=True, alpha=0.7, color='blue', label='Simulated Data')
plt.title('Simulated Poisson Distribution')
plt.xlabel('Number of Events')
plt.ylabel('Probability')
plt.legend()
plt.show()
```



Explanation:

Importing Libraries:

- import numpy as np: NumPy is a powerful library for numerical operations in Python.
- import matplotlib.pyplot as plt: Matplotlib is used for data visualization.
- Setting the Seed:np.random.seed(42): Sets the seed for NumPy's random number generator for reproducibility.

Defining Parameters:

- lambda_value = 3: The average rate of occurrence for the Poisson distribution.
- Simulating the Poisson Distribution:poisson_data = np.random.poisson(lambda_value, size=1000): Generates 1000 random numbers following a Poisson distribution with the specified average rate.

Visualizing the Data:

- plt.hist(...): Creates a histogram of the simulated Poisson data.
- bins=20: Specifies the number of bins in the histogram.
- density=True: Normalizes the histogram to represent probabilities.
- alpha=0.7: Adjusts the transparency of the bars.
- color='blue': Sets the color of the bars to blue.
- label='Simulated Data': Adds a label for the legend.

Plot Customization:

- plt.title(...), plt.xlabel(...), plt.ylabel(...): Adds a title and labels to the plot.
- plt.legend(): Displays the legend.

Displaying the Plot:

- plt.show(): Displays the generated histogram.

Interpretation of the Graph: The histogram represents the simulated Poisson distribution with an average rate (lambda) of The x-axis ('Number of Events') represents the possible outcomes (counts of events), and the y-axis ('Probability') represents the probability density for each bin.The bars illustrate the frequency or probability of each count occurring.The shape of the histogram approximates the characteristic shape of a Poisson distribution, with higher probabilities around the mean value (lambda) and a gradually decreasing tail.

Application of Poisson Distribution

Modeling Traffic Accidents with the Poisson Distribution:

The Poisson distribution is commonly used to model the occurrence of rare events over time. In the context of traffic accidents, it assumes that accidents happen independently at a constant average rate, and the Poisson distribution can describe the probability of a certain number of accidents within a specific time period.

Probability Mass Function (PMF) of Poisson Distribution:

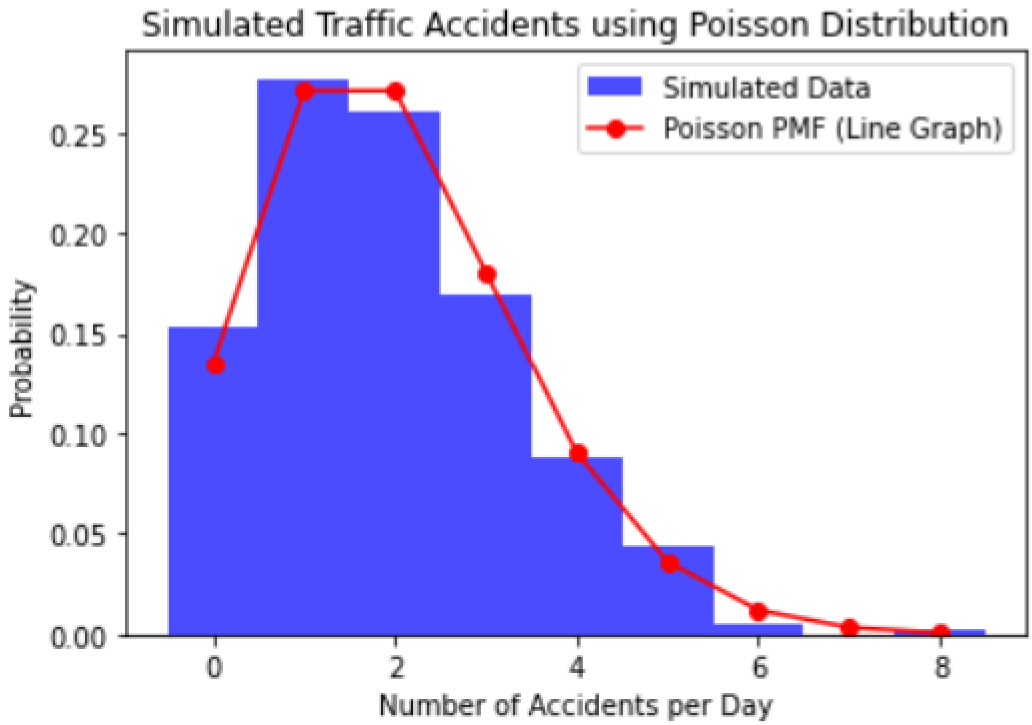
The PMF of the Poisson distribution is given by:

$$P(\mathrm{X}=x)=\frac{\lambda^{x}e^{-\lambda}}{x!}$$

where
x = 0, 1, 2, 3, …
λ = mean number of occurrences in the interval
e = Euler's constant $\approx$ 2.71828

Here's a Python program using NumPy and Matplotlib to simulate and visualize traffic accidents

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import poisson

np.random.seed(42)

average_accidents_per_day = 2
days_simulated = 365
time_period = 1  # In days

# Simulate the number of accidents per day using the Poisson distribution
accidents_per_day = np.random.poisson(average_accidents_per_day, days_simulated)

# Plot the simulated data
plt.hist(accidents_per_day, bins=np.arange(0, max(accidents_per_day) + 2) - 0.5, density=True, alpha=0.7, color='blue', label='Simulated Data')

# Plot the Poisson PMF as a line graph for comparison
x = np.arange(0, max(accidents_per_day) + 1)
poisson_pmf = poisson.pmf(x, average_accidents_per_day)
plt.plot(x, poisson_pmf, 'ro-', label='Poisson PMF (Line Graph)')

plt.title('Simulated Traffic Accidents using Poisson Distribution')
plt.xlabel('Number of Accidents per Day')
plt.ylabel('Probability')
plt.legend()
plt.show()
```



Explanation:

Setting the Seed:

- np.random.seed(42): Sets the seed for reproducibility, ensuring the same random numbers are generated each time the code is run.

Defining Parameters:

- average_accidents_per_day: Average rate of accidents per day.
- days_simulated: Number of days simulated.
- time_period: Time period for simulation (in days).

Simulating Accidents:

- accidents_per_day = np.random.poisson(average_accidents_per_day, days_simulated): Generates random numbers representing the number of accidents per day using the Poisson distribution.

Plotting the Simulated Data:

- plt.hist(...): Creates a histogram to visualize the simulated data.
- bins=np.arange(0, max(accidents_per_day) + 2) - 0.5: Defines bins for the histogram.
- density=True: Normalizes the histogram to represent probabilities.
- alpha=0.7: Adjusts the transparency of the bars.
- color='blue': Sets the color of the bars.
- label='Simulated Data': Adds a label for the legend.

Plotting the Poisson PMF as a Line Graph:

- plt.plot(x, poisson_pmf, 'ro-', label='Poisson PMF (Line Graph)'): Plots the Poisson probability mass function (PMF) as a line graph.
- x: The x-values represent the number of accidents.
- poisson_pmf: The corresponding Poisson PMF values.
- 'ro-': Styling for the line graph (red dots connected by lines).
- label='Poisson PMF (Line Graph)': Adds a label for the legend.

Labeling and Formatting:

- plt.title(...), plt.xlabel(...), plt.ylabel(...): Adds title and labels to the plot.
- plt.legend(): Displays the legend for better interpretation.

Interpretation of the Graph:

The blue histogram represents the simulated distribution of traffic accidents per day.

The red line graph depicts the Poisson probability mass function (PMF), showcasing the probability of observing a specific number of accidents per day according to the Poisson distribution.

The overlapping of the line graph with the histogram allows for a visual comparison between the simulated data and the theoretical Poisson distribution.

Conclusion

This study focused on extracting the probability mass feature from empirical observations, emphasizing instances characterized by heightened probability. Notably, in 75 percent of cases, the Poisson distribution, including its probability mass function, demonstrated the best fit. The evaluation of alternative distributions underscored that while they exhibited some impact on the hosting ability of the distribution function, this influence remained limited yet comparable.

The versatility of probability distributions was highlighted, showcasing that various distributions could yield meaningful results when employed with Python libraries. Notably, foundational libraries such as numpy and matplotlib were found to be sufficiently familiar, allowing for the generalization of these findings. Currently, pending the discovery of counterexamples, the broad applicability of these basic libraries underscores their reliability.

The discussion extended to the corporate realm, exploring the fundamental utility of the Poisson distribution and its consequential implications. Specific sections on the Poisson distribution were thoroughly examined, delving into its relevance to the distribution of Binomial, providing a nuanced understanding of its multifaceted applications.

Furthermore, an additional application (its use in traffic accident prediction) was explored, delving into its simulation using Python. This hands-on approach not only reinforced theoretical concepts but also provided practical insights into the real-world implementation of the Poisson distribution, further enriching the comprehensive analysis presented in the paper

Reference

An Efficient Python Approach for Simulation of Poisson Distribution

[1]D.K. Sharma, [2]Bhopendra Singh, [3]Raja M, [4]R. Regin, [5]S. Suman Rajest,
[1]Department of Mathematics, Jaypee University of Engineering and Technology, M.P., India.
[2]Amity University, Dubai.
[3]Department of Multimedia, VIT School of Design, Vellore Institute of Technology, Vellore, India.
[4]Department of information Technology, Adhiyamaan college of Engineering, Tamil Nadu, India.
[5]Vels Institute of Science, Technology & Advanced Studies, Tamil Nadu, India.
dilipsharmajiet@gmail.com, bsingh@amityuniversity.ae, raja.m@vit.ac.in,
regin12006@yahoo.co.in, sumanrajest414@gmail.com