Random Variables, Exponential Distribution, and Code Analysis

1 Introduction to Stochastic Models

A stochastic model is a mathematical tool used to describe systems or processes that include an element of randomness or uncertainty. Unlike deterministic models, where the same initial conditions always yield the same result, stochastic models produce different outcomes in each execution based on probabilities.

Characteristic	Stochastic Model	Deterministic Model
Outcome	Different results each run (probabilistic)	Always the same result for same inputs
Factors	Includes at least one random variable	All factors are known and fixed
Example	Predicting stock prices	Calculating planetary orbits

Table 1: Comparison between Stochastic and Deterministic Models

2 Random Variables

In stochastic models, we cannot predict exactly what will happen, but we know the probabilities. A **Random Variable (RV)** is a function that assigns a numerical value to each random outcome.

- **Example:** "Time until the next customer arrival" is a positive real value.
- **Discrete RVs:** Take specific values (e.g., number of arrivals per minute).
- **Continuous RVs:** Take any value within an interval (e.g., service time).

2.1 Probability Distributions

For every RV, we need a probability distribution to tell us how likely each value is.

- **Probability Density Function (PDF),** $f(x)$**:** For continuous variables, it describes the density of probability around a point x. To find the actual probability that X falls within an interval $[a, b]$, we integrate the PDF:

$$P(a \leq X \leq b) = \int_a^b f(x)dx$$

- **Cumulative Distribution Function (CDF),** $F(x)$**:** Gives the probability that the variable is less than or equal to x ($P(X \leq x)$).

3 The Exponential Distribution

The exponential distribution is a continuous probability distribution that models the time elapsed until a specific event occurs. It is fundamental to Queueing Theory (M/M/1 queues).

3.1 Connection to Poisson Process

It is closely related to the Poisson process. While the Poisson distribution counts "how many events" occur in a fixed time interval, the exponential distribution measures "how much time" until the next event.

- **Example:** Bus arrivals at a stop happen randomly but with a fixed average rate. The waiting time for the next bus follows an exponential distribution.

3.2 Mathematical Definition

The Probability Density Function (PDF) is defined as:

$$f(x; \lambda) = \lambda e^{-\lambda x} \quad \text{for } x \geq 0$$

Where:

- x: The time of interest (e.g., waiting time).
- λ (lambda): The **rate parameter**. It represents the average number of events per unit of time (e.g., $\lambda = 3$ customers/hour).

3.3 Mean and Variance

For a random variable $X \sim \text{Exp}(\lambda)$:

- **Mean (Expected Time):** $E[X] = \frac{1}{\lambda}$
- **Variance:** $Var(X) = \frac{1}{\lambda^2}$

Interpretation: A high arrival rate (λ) leads to a small mean inter-arrival time ($1/\lambda$) and low variability.

3.4 Memoryless Property

The exponential distribution is the only continuous distribution with the "memoryless" property. The past does not affect the future.

$$P(T > s + t \mid T > s) = P(T > t)$$

Meaning: If you have already waited s minutes, the probability of waiting another t minutes is the same as if you had just started waiting.

3.5 Calculation Example

If $\lambda = 3$ customers/hour, what is the probability that the waiting time is between 0 and 10 minutes?

- Convert 10 minutes to hours: $10/60 = 1/6$ hours.
- Calculate integral:

$$P(0 \leq X \leq 1/6) = \int_0^{1/6} 3e^{-3x} dx = 1 - e^{-3(1/6)} = 1 - e^{-0.5} \approx 0.393$$

There is a 39.3% chance the customer arrives within the first 10 minutes.

4 Code Analysis and Simulation

This section analyzes the theoretical vs. empirical comparison typically performed in Python (e.g., using Google Colab).

4.1 Theoretical vs. Empirical Quantities

A simulation code compares theoretical values derived from formulas with empirical values calculated from random samples.

Quantity	Interpretation	Formula
λ (Lambda)	Rate parameter (events per time unit).	Input
E_{theory}	Theoretical mean inter-arrival time.	$1/\lambda$
$E_{empirical}$	Calculated mean from data samples.	`samples.mean()`
Var_{theory}	Theoretical variance.	$1/\lambda^2$
$Var_{empirical}$	Variance from samples.	`samples.var()`

Table 2: Comparing Theoretical and Empirical Metrics

If $\lambda = 3$, then $E[X] = 1/3 \approx 0.3333$. This means we expect an event every 0.33 units of time on average.

4.2 Tail Probabilities ($P(X > t)$)

This measures the probability of waiting longer than time t.

- **Theory:** $P(X > t) = e^{-\lambda t}$
- **Empirical:** Calculated as the ratio: `(samples > t).mean()`

If the theoretical and empirical values are close, the data fits the distribution well. For example, with $\lambda = 3$ and $t = 1$, $P(X > 1) = e^{-3} \approx 0.05$. Only 5% of customers wait longer than 1 unit of time.

4.3 Graphical Verification

Visual tools are used to verify the fit of the data:

1. **PDF vs. Histogram:** The theoretical curve ($f(x) = \lambda e^{-\lambda x}$) should align with the histogram of the sampled data.
2. **CDF vs. ECDF:** The theoretical Cumulative Distribution Function ($F(x) = 1 - e^{-\lambda x}$) should match the Empirical CDF derived from the data.
3. **Q-Q Plot:** Plots theoretical quantiles against empirical quantiles. If the points lie on the $y = x$ line, the data follows the distribution. Deviations indicate "heavy tails" or an incorrect λ.

4.4 Parameter Estimation (MLE)

The Maximum Likelihood Estimation (MLE) for λ is calculated from the data:

$$\hat{\lambda} = \frac{1}{\bar{x}}$$

where $\bar{x}$ is the sample mean. If the simulation is correct, $\hat{\lambda}$ should be very close to the true λ.

4.5 Kolmogorov-Smirnov (KS) Test

The KS statistic measures the maximum difference between the Empirical CDF (data) and the Theoretical CDF (model).

- Lower values indicate a better fit.
- For large samples ($> 10,000$), a value below 0.02 typically indicates a very good fit.

5 Coding Notes

- **Python (NumPy):** The function `rng.exponential(scale)` uses the scale parameter $\beta = 1/\lambda$ (mean), not the rate λ.
- **C++ (Standard Library):** The class `std::exponential_distribution(lambda)` uses the rate parameter λ directly.