B.Tech V Semester (2025-26)
MC 303: Stochastic Processes
Assignment- III

Task 1: Discrete-Time Markov Chain for Customer Support System

Objective:

Simulate the transition of customers between different states in a customer support system and analyze the steady-state probabilities to optimize customer service efficiency.

Problem Description:

You are tasked with modeling a customer support system in which customers move between different states such as "Waiting," "On Hold," "Talking to Agent," and "Problem Resolved." Each state transition has a certain probability, and the customer's journey through these states can be represented by a discrete-time Markov Chain.

- States:
    - State 1: Waiting for an agent
    - State 2: On hold
    - State 3: Talking to an agent
    - State 4: Problem resolved
    - State 5: Exit (absorbing state)

The system operates such that once a customer's problem is resolved (State 4), they exit the system.

Expected Outcome:

- Determine the n-step transition probabilities for a customer who starts in the "Waiting" state.
- Classify the states as absorbing or transient.
- Compute the steady-state probabilities for each state, which shows the long-term behavior of the system (i.e., the fraction of time customers spend in each state).

Implementation Hints:

- Define a transition matrix where each entry represents the probability of moving from one state to another in a single time step.
- Use Markov Chain steady-state equations to calculate the limiting probabilities.
- Simulate the system for a large number of iterations to estimate the steady-state behavior (Markov Chain Monte Carlo).
- Use Python's numpy or pandas libraries for matrix operations and Monte Carlo simulation.

Task 2: Continuous-Time Markov Chain for Queueing Systems (M/M/1)

Objective:

Simulate a single-server queue (M/M/1) model and determine the steady-state probabilities for the number of customers in the system.

Problem Description:

You are tasked with modeling a queue at a bank where customers arrive randomly according to a Poisson process with rate λ, and each customer is served one at a time with an exponential service rate μ. The system consists of a single server.

- States:
  - The number of customers in the system (0, 1, 2, …).
- Parameters:
  - λ (arrival rate)
  - μ (service rate)

The system is modeled as a continuous-time Markov process, where the transitions between states occur at rates λ (for arrival) and μ (for service).

Expected Outcome:

- Determine the steady-state distribution of the system (probability of having 0, 1, 2, … customers).
- Calculate the mean queue length, mean waiting time, and system utilization (the fraction of time the server is busy).

Implementation Hints:

- The Chapman-Kolmogorov equation is key to modeling the system. For an M/M/1 queue, the system's steady-state probabilities can be derived using the birth-death process.
- Use Little's Law to relate the average number of customers, the average waiting time, and the arrival rate: $L = \lambda W$, where L is the average number of customers, λ is the arrival rate, and W is the average waiting time.
- Use exponentially distributed random variables for arrival and service times. You can generate them using Python's numpy.random.exponential.
- Simulate the queueing process using a discrete-event simulation approach and estimate the steady-state properties after a large number of events.

Task 3: Birth-Death Process for Population Growth

Objective:

Simulate a birth-death process for population dynamics, modeling the random birth and death rates of a species over time.

Problem Description:

You are tasked with modeling a population of animals where each individual in the population can either be born (with rate β) or die (with rate δ). The state of the system is the number of individuals in the population.

- States:
    - The number of individuals in the population (0, 1, 2, 3, …).
- Rates:
    - β (birth rate)
    - δ (death rate)

The birth-death process is a continuous-time Markov Chain, where transitions occur at rates β (birth) and δ (death).

Expected Outcome:

- Estimate the steady-state distribution of the population size.
- Compute the expected population size in the long run.
- Analyze how the birth and death rates affect population dynamics and stability.

Implementation Hints:

- Define the birth-death rates as parameters and use them to generate transition rates between population states.
- Use continuous-time Markov chains to model transitions between states with exponential holding times.
- Compute the steady-state probabilities for the population sizes using the balance equations for the birth-death process.
- Use simulation techniques to observe population size fluctuations over time and calculate the expected population size by averaging over a large number of simulation runs.

Task 4: Disease Spread Modeling with Absorbing States

Objective:

Model the spread of a disease in a population using a discrete-time Markov Chain with absorbing states to simulate the disease dynamics.

Problem Description:

You are tasked with modeling the spread of an infectious disease in a community where individuals can be in one of three states:

- State 1: Susceptible
- State 2: Infected
- State 3: Recovered (absorbing state)

Individuals can transition from Susceptible to Infected with a certain probability, and from Infected to Recovered with a different probability. Once an individual recovers, they no longer contribute to the disease spread (absorbing state).

Expected Outcome:

- Compute the probabilities of an individual being Susceptible, Infected, or Recovered after a large number of transitions (steady-state probabilities).
- Analyze the time until absorption (i.e., the time until all individuals recover or the disease dies out).
- Study how varying the transition probabilities (infection and recovery rates) affects the disease's spread in the population.

Implementation Hints:

- Use transition matrices to represent the probabilities of moving from one state to another.
- Use absorbing Markov chains to calculate the probability of eventually reaching the Recovered state.
- Simulate the system using Monte Carlo techniques by iterating the transitions over many runs.
- Study the limiting distribution to see the long-term fraction of Susceptible, Infected, and Recovered individuals.