Homework 3

STAT 201A - Fall 2023

Author: Frederik Stihler

Department of Statistics

Instructor: Adrian Gonzalez Casanova Soberon

GSI: Facundo Sapienza

Collaborators: Max Medina, Isabel Moreno

30th of November, 2023

1. Simulation of Markov Process.

a) Write the Markov process in matrix representation, that is, define the matrix $P \in \mathbb{R}^{3\times3}$ such that P_{ij} is the probability of transitioning from the node i to j.

We have the following transition matrix:

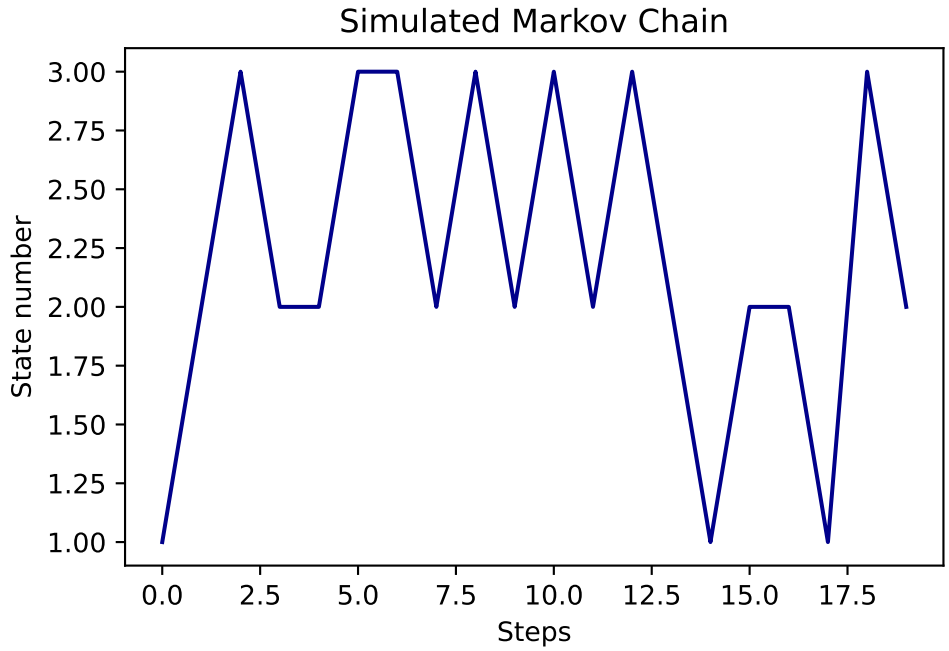
$$P = \begin{bmatrix} 0.2 & 0.7 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.2 & 0.4 & 0.4 \end{bmatrix}$$

b) Simulate one single realization of the chain, that is, starting from $X_0 = 1$, update the value of X_i using the probabilities defined by the process.

```python
import numpy as np

X_0 = 1
iter_max = 19
P = np.array([[0.2, 0.7, 0.1], [0.2, 0.5, 0.3], [0.2, 0.4, 0.4]])

def MC(initial_state, transition_matrix, iterations):
    current_state = initial_state
    simulation = [current_state]

    for _ in range(iterations):
        probs = transition_matrix[current_state - 1, :]
        next_state = np.random.choice(
          np.arange(1, len(probs) + 1), p=probs)
        simulation.append(next_state)
        current_state = next_state

    return simulation

# Simulation
result = MC(X_0, P, iter_max)

# Results
print(f"Simulated Markov Chain ({iter_max} iterations):")
print(result)

import matplotlib.pyplot as plt
```

```
_ = plt.plot(result, color = "darkblue")
_ = plt.title("Simulated Markov Chain")
_ = plt.xlabel("Steps")
_ = plt.ylabel("State number")
```

```
Simulated Markov Chain (19 iterations):
[1, 2, 3, 2, 2, 3, 3, 2, 3, 2, 3, 2, 3, 2, 1, 2, 2, 1, 3, 2]
```



2. Stationary Distribution.

The goal of this section is to show the convergence of the probability distribution of the Markov process.

a) Calculate theoretically the stationary state of the process by finding the vector $\pi_\infty \in \mathbb{R}^3$ such that $\pi_\infty^T = \pi_\infty^T P$. Notice that this is the same as finding the eigenvector with eigenvalues equals one of the matrix P^T . This is the same as solving $(P^T - I)\pi_\infty = 0$. You can solve the linear system of equation numerically or analytically.

Recall that:

$$P = \begin{bmatrix} 0.2 & 0.7 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.2 & 0.4 & 0.4 \end{bmatrix}$$

with transpose:

$$P^T = \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.7 & 0.5 & 0.4 \\ 0.1 & 0.3 & 0.4 \end{bmatrix}$$

Now we want to solve:

$$(P^T - I)\pi_\infty = 0;$$

which is equal to:

$$(P^T - I)\pi_\infty = \begin{bmatrix} -0.8 & 0.2 & 0.2 \\ 0.7 & -0.5 & 0.4 \\ 0.1 & 0.3 & -0.6 \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

We can bring this in the form (Gaussian elimination):

$$\begin{bmatrix} -0.8 & 0.2 & 0.2 \\ 0 & -2.6 & 4.6 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Then, we set $\pi_3 = t$ and solve the system of equations with infinite solutions:

$$\begin{aligned} -0.8\pi_1 + 0.2\pi_2 + 0.2t &= 0 \\ -2.6\pi_2 + 4.6t &= 0 \end{aligned}$$

This yields:

$$\pi_\infty = \begin{bmatrix} \frac{0.9t}{1.3} \\ \frac{2.3t}{1.3} \\ t \end{bmatrix}$$

As we know that the elements of π_∞ must sum to 1, we solve the following for t:

$$\begin{aligned} \frac{0.9t}{1.3} + \frac{2.3t}{1.3} + t &= 1 \\ t(\frac{0.9}{1.3} + \frac{2.3}{1.3} + 1) &= 1 \\ t &= \frac{1.3}{4.5} \end{aligned}$$

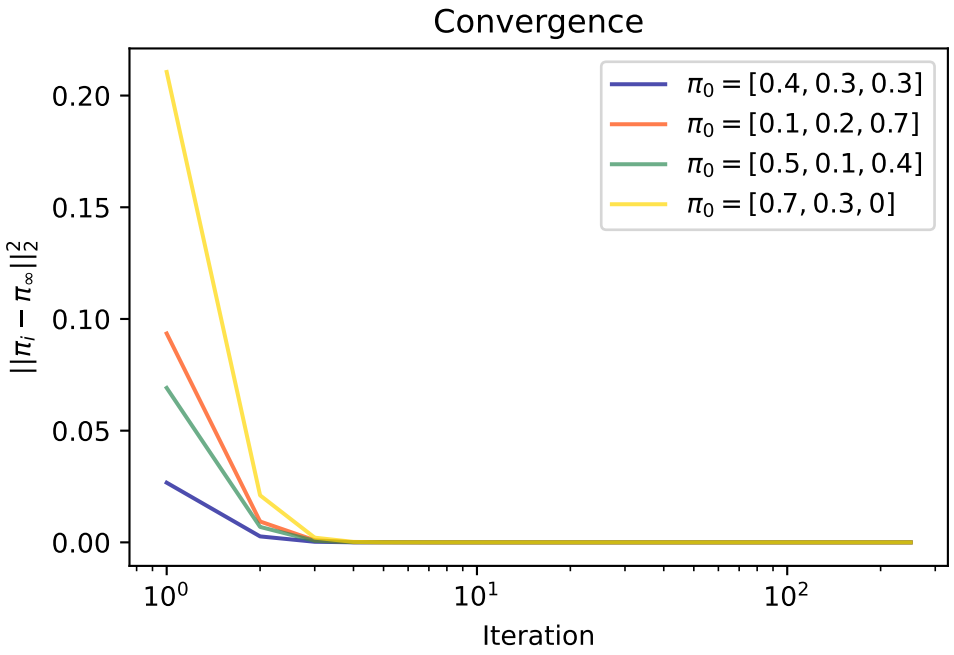
It follows that the stationary distribution π_∞ is:

$$\pi_\infty = \begin{bmatrix} 0.2 \\ \frac{2.3}{4.5} \\ \frac{1.3}{4.5} \end{bmatrix} \approx \begin{bmatrix} 0.2 \\ 0.5 \\ 0.3 \end{bmatrix}$$

b) Starting now from an initial probability distribution π_0 on the nodes, compute the value of $\pi_i^T = \pi_0^T p_i$ the probability distribution at time i. Show that $\pi_i \to \pi_\infty$ and make plot of i vs $||\pi_i - \pi_\infty||_2^2$. Generate this plot for at least two different initial conditions π_0 and compare.

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

# Transition matrix
P = np.array([[0.2, 0.7, 0.1],
              [0.2, 0.5, 0.3],
              [0.2, 0.4, 0.4]])

# Stationary distribution
pi_inf = np.array([0.2, 2.3/4.5, 1.3/4.5])

# Initial conditions
pi_0_1 = np.array([0.3, 0.4, 0.3])
pi_0_2 = np.array([0.05, 0.3, 0.65])
pi_0_3 = np.array([0.5, 0.1, 0.4])
pi_0_4 = np.array([0.8, 0.2, 0])
```

```
# iterations
iterations = 250

# Arrays to store L2 norm values for each iteration
l2_norm_1 = np.zeros(iterations)
l2_norm_2 = np.zeros(iterations)
l2_norm_3 = np.zeros(iterations)
l2_norm_4 = np.zeros(iterations)

# Calculate pi_i and L2 norm for each iteration
for i in range(iterations):
    pi_i_1 = np.dot(pi_0_1, np.linalg.matrix_power(P, i + 1))
    pi_i_2 = np.dot(pi_0_2, np.linalg.matrix_power(P, i + 1))
    pi_i_3 = np.dot(pi_0_3, np.linalg.matrix_power(P, i + 1))
    pi_i_4 = np.dot(pi_0_4, np.linalg.matrix_power(P, i + 1))

    # Calculate L2 norm
    l2_norm_1[i] = np.linalg.norm(pi_i_1 - pi_inf)
    l2_norm_2[i] = np.linalg.norm(pi_i_2 - pi_inf)
    l2_norm_3[i] = np.linalg.norm(pi_i_3 - pi_inf)
    l2_norm_4[i] = np.linalg.norm(pi_i_4 - pi_inf)

# Plot
plt.plot(range(1, iterations + 1), l2_norm_1,
label=r'$\pi_0 = [0.4, 0.3, 0.3]$', alpha=0.7, color = "darkblue")
plt.plot(range(1, iterations + 1), l2_norm_2,
label=r'$\pi_0 = [0.1, 0.2, 0.7]$', alpha=0.7, color = "orangered")
plt.plot(range(1, iterations + 1), l2_norm_3,
label=r'$\pi_0 = [0.5, 0.1, 0.4]$', alpha=0.7, color = "seagreen")
plt.plot(range(1, iterations + 1), l2_norm_4,
label=r'$\pi_0 = [0.7, 0.3, 0]$', alpha=0.7, color = "gold")
plt.xscale('log')
plt.xlabel('Iteration')
plt.ylabel(r'$||\pi_i - \pi_\infty||_2^2$')
plt.title("Convergence")
plt.legend()
plt.show()
```



All lines in the plot appear to be similar and differ only in their starting values along the y-axis. This shows that the Markov chain converges to the stationary distribution π_∞ regardless of the initial conditions. As the number of iterations increases, the probability distribution π_i for different initial conditions consequently tends to approach the stationary distribution π_∞ .

3. Absorbing state.

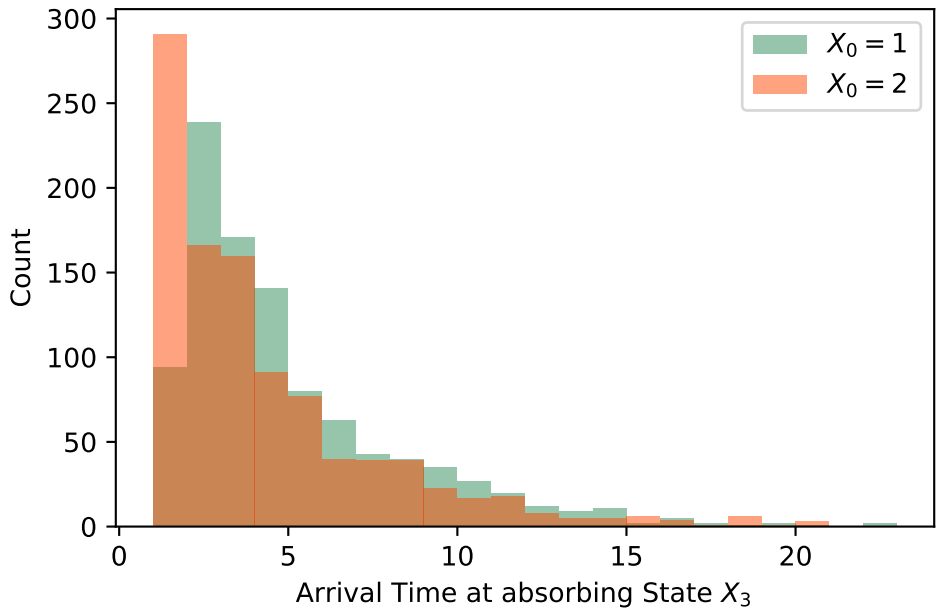
Consider now that node 3 is an absorbing state and we want to estimate the waiting time until the process arrives at $X_i = 3$ from any other node.

a) Starting from each one of $X_0 = 1$ and $X_0 = 2$, run multiple simulation of the Markov chain (Problem 1, part b) until $X_i = 3$ and store the arrival time untilthis happens. Make a histogram of the arrival time for both $X_0 = 1$ and $X_0 = 2$ and compute the mean.

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

def simulate_MC_until_absorbing(X_0, transition_matrix):
    current_state = X_0
    time_steps = 0
```

```
    while current_state != 3:
        probs = transition_matrix[current_state - 1, :]
        next_state = np.random.choice(
          np.arange(1, len(probs) + 1), p=probs)
        current_state = next_state
        time_steps += 1

    return time_steps

P = np.array([[0.2, 0.7, 0.1], [0.2, 0.5, 0.3], [0.2, 0.4, 0.4]])

# Initial states
initial_state_1 = 1
initial_state_2 = 2

# Iterations
iterations = 1000

# Simulate and record arrival times
arrival_times_1 = [simulate_MC_until_absorbing(
  initial_state_1, P) for _ in range(iterations)]
arrival_times_2 = [simulate_MC_until_absorbing(
  initial_state_2, P) for _ in range(iterations)]

# Plot
plt.hist(arrival_times_1, bins=range(min(arrival_times_1),
 ↪  max(arrival_times_1) + 1),
 alpha=0.5, label=r'$X_0 = 1$', color = "seagreen")
plt.hist(arrival_times_2, bins=range(min(arrival_times_2),
 ↪  max(arrival_times_2) + 1),
 alpha=0.5, label=r'$X_0 = 2$', color = "orangered")

plt.xlabel('Arrival Time at absorbing State $X_3$')
plt.ylabel('Count')
plt.legend()
plt.show()

# Mean arrival times
mean_arrival_time_1 = np.mean(arrival_times_1)
mean_arrival_time_2 = np.mean(arrival_times_2)
```

```
print("Mean Arrival Time (X_0 = 1):", mean_arrival_time_1)
print("Mean Arrival Time (X_0 = 2):", mean_arrival_time_2)
```




```
Mean Arrival Time (X_0 = 1): 4.533
Mean Arrival Time (X_0 = 2): 3.842
```

b. Compute theoretically the mean arrival time to the absorbing state and compare it with part a).

To do so, notice that if T_i denotes the random variable associated to the arrival time starting from $X_0 = i$, then:

$$\mu_i = 1 + \sum_{j=1}^{3} p_{ij}\mu_j,$$

with $\mu_i = \mathbb{E}[T_i]$ This is a linear system of equations that you can solve. Notice $T_3 = 0$.

For the 3 states (i=1,2,3) the system of equations is:

$$\mu_1 = 1 + p_{11}\mu_1 + p_{12}\mu_2 + p_{13}\mu_3$$
$$\mu_2 = 1 + p_{21}\mu_1 + p_{22}\mu_2 + p_{23}\mu_3$$
$$\mu_3 = 0$$

Now we substitute the values from our transition matrix P and leave out the terms with $\mu_3 = 0$. We solve this for μ_1 and μ_2:

$$\mu_1 = 1 + 0.2\mu_1 + 0.7\mu_2$$
$$\mu_2 = 1 + 0.2\mu_1 + 0.5\mu_2$$

We get:

$$\mu_1 = \frac{5}{4} + \frac{7}{8}\mu_2$$
$$\rightarrow \mu_2 = \frac{50}{13}$$
$$\rightarrow \mu_1 = \frac{60}{13}$$

Hence the theoretical mean arrival times to the absorbing state are $\mu_1 = 4.615$ and $\mu_2 = 3.846$. We observe that these are close to the simulated means. This highlights the accuracy of the estimates for the waiting time until the process reaches the absorbing state.