STAT 753: Stochastic Models and Simulations

Jaleesa Houle
University of Nevada, Reno - Spring 2024

Homework 5

Theory

1. Consider the continuous-time Markov chain on the state space with the generator

$$A = \begin{bmatrix} -2 & 1 & 1 \\ 3 & -4 & 1 \\ 1 & 0 & -1 \end{bmatrix} \tag{1}$$

a) Find its stationary distribution.

Solution
The stationary distribution for A can be found by using the equations $\pi A = 0$ where $\pi = \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 \end{bmatrix}$ and $\pi_1 + \pi_2 + \pi_3 = 1$.

$$\begin{bmatrix} \pi_1 & \pi_2 & \pi_3 \end{bmatrix} \begin{bmatrix} -2 & 1 & 1 \\ 3 & -4 & 1 \\ 1 & 0 & -1 \end{bmatrix} = 0$$

$$\begin{Bmatrix} -2\pi_1 + 3\pi_2 + \pi_3 & = 0 \\ \pi_1 - 4\pi_2 & = 0 \\ \pi_1 + \pi_2 - \pi_3 & = 0 \\ \pi_1 + \pi_2 + \pi_3 & = 1 \end{Bmatrix} \tag{2}$$

$$\begin{Bmatrix} \pi_1 = 2/5 \\ \pi_2 = 1/10 \\ \pi_3 = 1/2 \end{Bmatrix}$$

The corresponding stationary distribution is $\begin{bmatrix} 0.4 & 0.1 & 0.5 \end{bmatrix}$.

b) Find the probability that at state 2, we spend more than 0.4 time at this state before jumping away.

Solution

The exit rate for state 2 is given as $T_2 := min(T_{21}, T_{23}) \sim Exp(4)$. We can find the probability of a time spent at state 2 by integrating over this pdf.

$$
\begin{aligned}
p(T) &= 4e^{-4t}, t > 0 \\
P(T > 0.4) &= \int_{0.4}^{\infty} 4e^{-4t} dt \\
&= -e^{-4t} \Big|_{t=0.4}^{\infty} \\
&= 0 + e^{-4(0.4)} \\
&= 0.2019
\end{aligned} \tag{3}
$$

.

The probability that we spend more than 0.4 time at state 2 before jumping away is 0.2019.

c) Find the mean, the median, and the standard deviation of time it spends at state 1 before jumping away.

Solution

The exit rate for state 1 is $T_1 := min(T_{12}, T_{13}) \sim Exp(2)$. The mean of an exponential random variable with rate r is $E(X) = 1/r$, the variance is $1/r^2$, and the median is found by solving $P(X \leq m) = 1/2$. The exit rate for state 1 is 2, so the corresponding mean is $1/r = 1/2$ and the standard deviation can be seen as $\sqrt{1/r^2} = \sqrt{1/2^2} = 1/2$. Lastly, the median can be computed as

$$
\begin{aligned}
p(T) &= 2e^{-2t}, t > 0 \\
P(T \leq m) &= \int_{0}^{m} 2e^{-2t} dt = 1/2 \\
&= -e^{-2t} \Big|_{0}^{m} \\
1/2 &= -e^{-2m} + 1 \\
1/2 &= e^{-2m} \\
m &= \frac{ln(0.5)}{-2} \\
m &= 0.3466.
\end{aligned} \tag{4}
$$

To summarize, the mean, median, and standard deviation of time spent in state 1 are 1/2, 0.3466, and 1/2, respectively.

d) Find the corresponding discrete-time Markov chain, and its own stationary distribution.

Solution

The corresponding discrete-time Markov chain can be found by reducing the diagonal elements of the generating matrix to zero and normalizing the off-diagonal elements in each row by the exit rate so that the values of each row sum to one. The exit rates for states 1, 2, and 3 are 2, 4, and 1, respectively. Reducing the generating matrix in this way yields the transition matrix

$$A = \begin{bmatrix} 0 & 1/2 & 1/2 \\ 3/4 & 0 & 1/4 \\ 1 & 0 & 0 \end{bmatrix}. \tag{5}$$

The corresponding stationary distribution can be found using the relations $p = pA$ and $p_1 + p_2 + p_3 = 1$.

$$\begin{aligned} \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix} &= \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 & 1/2 & 1/2 \\ 3/4 & 0 & 1/4 \\ 1 & 0 & 0 \end{bmatrix} \\ &= \begin{Bmatrix} p_1 = 3/4p_2 + p_3 \\ p_2 = 1/2p_1 \\ p_3 = 1/2p_1 + 1/4p_2 \end{Bmatrix} \\ &=> \begin{Bmatrix} p_1 = 8/17 \\ p_2 = 4/17 \\ p_3 = 5/17 \end{Bmatrix} \end{aligned} \tag{6}$$

Thus the corresponding stationary distribution to the discrete-time Markov chain is $\begin{bmatrix} 8/17 & 4/17 & 5/17 \end{bmatrix}$.

2. Take a discrete-time Markov chain with 3 states and the transition matrix

$$P = \begin{bmatrix} 0 & 0.7 & 0.3 \\ 1/3 & 0 & 2/3 \\ 1 & 0 & 0 \end{bmatrix} \tag{7}$$

Take exit intensities from states 1, 2, 3 are $\lambda_1 = 20$, $\lambda_2 = 3$, $\lambda_3 = 5$. Find the generator of the corresponding continuous-time Markov chain.

Solution

For a discrete-time Markov chain with given exit intensities λ, the continuous-time Markov chain generating matrix is found by multiplying each row by the corresponding exit intensity. The diagonal values are then taken as $-\lambda_i$ for each state i so that each row sums to zero. Doing so with matrix P gives the generating matrix

$$P = \begin{bmatrix} -20 & 14 & 6 \\ 1 & -3 & 2 \\ 5 & 0 & -5 \end{bmatrix}. \tag{8}$$

Coding

1. Simulate the continuous-time Markov chain from Theory Problem 1 and answer all questions in that problem using this simulation.

Solution

Part a: Find A's stationary distribution.

```
import numpy as np
A = np.array([[-2,1,1],[3,-4,1],[1,0,-1]])
# Calculate the eigenvalues and eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(A.T)
# Find the index of the eigenvalue corresponding to the largest real part
↪   (should be 0 for CTMC)
idx = np.argmax(np.real(eigenvalues))
# Extract the corresponding eigenvector
v = np.real(eigenvectors[:, idx])
# Normalize the eigenvector to get the stationary distribution
v = v / np.sum(v)
print('Stationary distribution = ', v)

>> Stationary distribution =  [0.4 0.1 0.5]
```

We can see that the stationary distributions calculated from part a matches those found in the theory section above.

Part b: Find the probability that at state 2, we spend more than 0.4 time at this state before jumping away.

```
sim = [np.random.choice([0,1,2])]
jump_time = [0]
time1 = []
time2_long=[]
time2_short = []
N = 1000

np.random.seed(1)

for k in range(N):
    if sim[-1] == 0:
        clock01 = np.random.exponential(1/A[0,1])
        clock02 = np.random.exponential(1/A[0,2])
        if clock01 < clock02:
            jump_time.append(jump_time[-1]+clock01)
            sim.append(1)
            time1.append(clock01)
        else:
            jump_time.append(jump_time[-1]+clock02)
            sim.append(2)
            time1.append(clock02)
```

```

    if sim[-1] == 1:
        clock10 = np.random.exponential(1/A[1,0])
        clock12 = np.random.exponential(1/A[1,2])
        if clock10 < clock12:
            jump_time.append(jump_time[-1]+clock10)
            sim.append(0)
            if clock10 > 0.4:
                time2_long.append(clock10)
            else:
                time2_short.append(clock10)
        else:
            jump_time.append(jump_time[-1]+clock12)
            sim.append(2)
            if clock12 > 0.4:
                time2_long.append(clock12)
            else:
                time2_short.append(clock12)

    if sim[-1] == 2:
        clock20 = np.random.exponential(1/A[2,0])
        sim.append(0)
        jump_time.append(jump_time[-1]+clock20)

print('Probability of spending more than 0.4 time in state 2:', len(time2_long )/ (len(time2_long) + len(time2_short)))

>> Probability of spending more than 0.4 time in state 2: 0.1893939393939394
```

The probability of spending more that 0.4 time in state 2 over a simulation of 1000 steps was approximately 0.1894, whereas the theoretical probability was found to be 0.2019. The error between the empirical and theoretical results is approximately 6.2%, demonstrating these answers are in agreement.

Part c: Find the mean, the median, and the standard deviation of time it spends at state 1 before jumping away.

```
print('Mean time spent in state 1:', np.mean(time1))
print('Median time spent in state 1:', np.median(time1))
print('Standard deviation of time spent in state 1:', np.std(time1))

>> Mean time spent in state 1: 0.5033819972561435
>> Median time spent in state 1: 0.3505871329268793
>> Standard deviation of time spent in state 1: 0.4971136157767725
```

The empirical estimates of mean, median, and standard deviations of time spent in state 1 were found to be 0.5034, 0.3506, and 0.4971, respectively. The theoretical mean, median, and standard deviations are 0.5, 0.3466, and 0.5, respectively. The errors between these values are all around or less than 1%, which demonstrates strong agreement between the empirical and theoretical solutions.

Part d: Find the corresponding discrete-time Markov chain, and its own stationary distribution.

```
l = -np.diagonal(A) #exit rates
P = A / l[:,None] # scale each row by corresponding exit rate
np.fill_diagonal(P, 0) #make diagonal elements 0
print('Transition matrix \n P= \n', P)

>> Transition matrix
>>  P=
>>  [[0.   0.5  0.5 ]
>>  [0.75 0.   0.25]
>>  [1.   0.   0.  ]]

# Calculate the eigenvalues and eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(P.T)
# Find the index of the eigenvalue corresponding to the largest real part
idx = np.argmax(np.real(eigenvalues))
# Extract the corresponding eigenvector
v = np.real(eigenvectors[:, idx])
# Normalize the eigenvector to get the stationary distribution
v = v / np.sum(v)
print('Stationary distribution = ', v)

>> Stationary distribution =  [0.47058824 0.23529412 0.29411765]
```

We can see that the stationary distributions calculated from part d matches those found in the theory section above.