PES University, Bangalore

Established under Karnataka Act No. 16 of 2013

UE20CS312 - Data Analytics - Worksheet 5 Course instructor: Gowri Srinivasa, Professor Dept. of CSE, PES University

Designed by Yashas Kadambi, Dept. of CSE - yashasks@pesu.pes.edu

Markov Chains and AB Testing

Prerequisites

- Revise the following concepts
    - Markov Chains
    - Markov Chains with Absorbing states
    - Calculation of eventual probability of aborbtion
    - A/B Testing
- Install the following software
    - pandas
    - numpy

Points

The problems in this worksheet are for a total of 10 points with each problem having a different weightage.

- Problem 1: 2 points
- Problem 2: 4 points
- Problem 3: 4 points

Scenario 2

Its a freezing day in New york, Commisioner Wench has sent a report to Captain Holt that the 99th precinct has much lower reported crimes compared to other precincts. Upon Analysis Captain Holt decides to add feedback unit along with the 4 major units to analyse this descripency. All the units are mentioned below

1. Major Crimes
2. Traffic
3. General crimes

4. Feedback
5. Theft

The initial probablity of a person going to a particular unit on a particular day is given as follows

Major Crimes	Traffic	General crimes	Feedback	Theft
0.3	0.4	0.1	0.15	0.05

To measure how many people will go to the feedback unit, the personel files of all employees are give to the ***Move-o-Tron 99*** and it gives us the following matrix which shows us the probability of people moving from one unit to another on a particular day . It is known that the _ **Move-o-Tron 99**_ alwasy outputs matices which follow a first order Markov chain.

	Major Crimes	Traffic	General crimes	Feedback	Theft
Major Crimes	0.002	0.666	0.31	0.0	0.022
Traffic	0.466	0.102	0.222	0.0	0.21
General crimes	0.022	0.11	0.502	0.0	0.366
Feedback	0.0	0.0	0.0	1.0	0.0
Theft	0.11	0.122	0.066	0.0	0.702

As the people of New York are smart the will learn where all the units are present and hence the next days (day 1) distribution will be the distribution present at the end of the current day (day 0). Captain holt want to check if the matrix given by the ***Move-o-Tron*** can be used to model the footfall.

Problem 1 (2 points)

1. What technique can be used to model the probability of people going to the correct unit to report thier crime after N days? (0 points)
2. Is the chain irreducible? Justify (0.5 point)
3. What will be the intital probability of a person going to a particular unit after 1 day, 2 days, 10 days, 1000 days and 1001 days. (1 point)

   Hint: Use the Chapman−Kolmogorov relationship

```
# C = A.B
matrix_C = np.dot(matrix_A, matrix_B)

# C = A.(B^4) can be replaced by
matrix_C = matrix_A
for _ in range(4):
    matrix_C = np.dot(matrix_C, matrix_B)
```

```
# Useful function np.array.squueze()
print(a)
# [[1, 2, 3]]
print(a.squeeze())
# [1, 2, 3]
```

4. What can you say about the markov chain from state of intital probability of a person going to a particular unit after 1000 and 1001 days? (0.5 points)

```
# Importing Librarire
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt



# encoding the probabilities as a numpy array
trans_array = np.array([
    [0.002, 0.666, 0.31, 0, 0.022],
    [0.466, 0.102, 0.222, 0, 0.21],
    [0.022, 0.11, 0.502, 0, 0.366],
    [0, 0, 0, 1, 0],
    [0.11, 0.122, 0.066, 0, 0.702]
])
# ensures that the probabilities sum to 1
assert(trans_array[0].sum() == 1.0)
assert(trans_array[1].sum() == 1.0)
assert(trans_array[2].sum() == 1.0)
assert(trans_array[3].sum() == 1.0)
assert(trans_array[4].sum() == 1.0)



# encoding the initial probability of as a numpy array
state = np.array([[0.3, 0.4, 0.1, 0.15, 0.05]])
assert(state[0].sum() == 1.0)


# Solution
print("Question 1")
print("\tMarkov Chains\n")

print("Question 2")
print("\tNo the Markov Chain is reducible as Feedback unit is not accessable or communicable with the o
# https://stats.stackexchange.com/questions/186033/how-do-you-see-a-markov-chain-is-irreducible
print("Question 3")
# after 1 day
state_1 = np.dot(state, trans_array)
print(f'\tstate after 1 day = {state_1.squeeze()} ')
```

```
# after 2 days
state_2 = np.dot(state_1, trans_array)
print(f'\tstate after 2 days = {state_2.squeeze()} ')


# after 10 days
state_10 = state
for _ in range(10):
    state_10 = np.dot(state_10, trans_array)
print(f'\tstate after 10 days = {state_10.squeeze()} ')


# after 1000 days
state_1000 = state
for _ in range(1000):
    state_1000 = np.dot(state_1000, trans_array)
print(f'\tstate after 1000 days = {state_1000.squeeze()} ')


# after 1001 days
state_1001 = np.dot(state_1000, trans_array)
print(f'\tstate after 1001 day = {state_1001.squeeze()} \n')

print('Question 4')
print('\tThe state after 1000 and 1001 days shows that there is no change in the probability distributi
Hence These states represent the stationary probability distribution of the Markov Chain')
```

```
Question 1
        Markov Chains

Question 2
        No the Markov Chain is reducible as Feedback unit is not accessable or communicable with t

Question 3
        state after 1 day = [0.1947 0.2577 0.2353 0.15   0.1623]
        state after 2 days = [0.1435072 0.2016392 0.2463988 0.15      0.2584548]
        state after 10 days = [0.12139467 0.16406836 0.1973962  0.15       0.36714077]
        state after 1000 days = [0.1214373  0.16411091 0.19739717 0.15       0.36705462]
        state after 1001 day = [0.1214373  0.16411091 0.19739717 0.15       0.36705462]

Question 4
        The state after 1000 and 1001 days shows that there is no change in the probability distr:
        Hence These states represent the stationary probability distribution of the Markov Chain
```

After analysing the model Captain holt calls the squad and educates them to ask people to give feedbacks. And the details of the squad are given to the ***Move-o-Tron 99***. After reanalyising the report the ***Move-o-Tron 99*** gave out a new Matrix, which is shown below

	Major Crimes	Traffic	General crimes	Feedback	Theft
Major Crimes	0.002	0.666	0.01	0.02	0.302

	Major Crimes	Traffic	General crimes	Feedback	Theft
Traffic	0.466	0.102	0.02	0.032	0.38
General crimes	0.0	0.0	1.0	0.0	0.0
Feedback	0.0	0.0	0.0	1.0	0.0
Theft	0.11	0.122	0.066	0.004	0.698

Considering the new report the model has to be re-evaluated. The initial probablity of a person going to a particular unit on a particular day remains the same.

Problem 2 (4 points)

1. Is the chain irreducible? Justify (0.5 point)
2. What will be the intital probability of a person going to a particular unit after 1 day, 2 days, 10 days, 1000 days and 1001 days. (1 point)

   Hint: Use the Chapman–Kolmogorov relationship
3. What can you say about the markov chain from state of intital probability of a person going to a particular unit after 1000 and 1001 days? (0.5 points)
4. Summer Edgecombe is Confidential Informant (CI) to the Officer Kimbal Cho and comes in every day to the police station. If on the first day she goes to the Major crimes Unit what will be the probability that she gives a feedback? (2 points)

```
# np.delete()
# https://note.nkmk.me/en/python-numpy-delete/#:~:text=Using%20the%20NumPy%20function%20np,f

print(a)
# [[ 0  1  2  3]
#  [ 4  5  6  7]
#  [ 8  9 10 11]]

print(np.delete(a, 1, 0))
# [[ 0  1  2  3]
#  [ 8  9 10 11]]

print(np.delete(a, 1, 1))
# [[ 0  2  3]
#  [ 4  6  7]
#  [ 8 10 11]]

# Deleting multiple rows or columns
print(np.delete(a, [0, 3], 1))
```

```
    # [[ 1  2]
    #  [ 5  6]
    #  [ 9 10]]

    # Deleting rows and columns
    print(np.delete(np.delete(a, 1, 0), 1, 1))
    # [[ 0  2  3]
    #  [ 8 10 11]]
```

```
# encoding the probabilities as a numpy array
trans_array = np.array([
    [0.002, 0.666, 0.01, 0.020, 0.302],
    [0.466, 0.102, 0.02, 0.032, 0.38],
    [0.0, 0.0, 1, 0.0, 0.0],
    [0, 0, 0, 1, 0],
    [0.11, 0.122, 0.066, 0.004, 0.698]
])

# ensures that the probabilities sum to 1
assert(trans_array[0].sum() == 1.0)
assert(trans_array[1].sum() == 1.0)
assert(trans_array[2].sum() == 1.0)
assert(trans_array[3].sum() == 1.0)
assert(trans_array[4].sum() == 1.0)

# encoding the initial probability of as a numpy array
state = np.array([[0.3, 0.4, 0.1, 0.15, 0.05]])
assert(state[0].sum() == 1.0)



# Solution

print("Question 1")
print("\tNo the Markov Chain is reducible as Feedback unit or General crimes unit is not accessable or


print("Question 2")
# after 1 day
state_1 = np.dot(state, trans_array)
print(f'\tstate after 1 day = {state_1.squeeze()} ')

# after 2 days
state_2 = np.dot(state_1, trans_array)
print(f'\tstate after 2 days = {state_2.squeeze()} ')


# after 10 days
state_10 = state
for _ in range(10):
```

```
    state_10 = np.dot(state_10, trans_array)
print(f'\tstate after 10 days = {state_10.squeeze()} ')


# after 1000 days
state_1000 = state
for _ in range(1000):
    state_1000 = np.dot(state_1000, trans_array)
print(f'\tstate after 1000 days = {state_1000.squeeze()} ')


# after 1001 days
state_1001 = np.dot(state_1000, trans_array)
print(f'\tstate after 1001 day = {state_1001.squeeze()} \n')

print('Question 3')
print('\tThe state after 1000 and 1001 days shows that there is some change in the probability distribu


print('Question 4')

q_array = np.delete(np.delete(trans_array, [2, 3], 0), [2, 3], 1)
print(f'\tThe matrix Q is \n\t{q_array[0]}\n\t{q_array[1]}\n\t{q_array[2]}\n')

r_array = np.delete(np.delete(trans_array, [2, 3], 0), [0,1,4], 1)
print(f'\tThe matrix R is \n\t{r_array[0]}\n\t{r_array[1]}\n\t{r_array[2]}\n')

i_array = np.eye(3)
print(f'\tThe matrix I is \n\t{i_array[0]}\n\t{i_array[1]}\n\t{i_array[2]}\n')

f_array = np.linalg.inv(i_array - q_array)
print(f'\tThe matrix F is \n\t{f_array[0]}\n\t{f_array[1]}\n\t{f_array[2]}\n')

fr_array = f_array @ r_array
print(f'\tProbability of eventual apsorbtion\n\t{fr_array[0]}\n\t{fr_array[1]}\n\t{fr_array[2]}\n')

print(f'\tProbaility that CI will give a feed back if she enter Major crimes on day one is {fr_array[0,
```

```
Question 1
        No the Markov Chain is reducible as Feedback unit or General crimes unit is not accessabl

Question 2
        state after 1 day = [0.1925 0.2467 0.1143 0.169  0.2775]
        state after 2 days = [0.1458722 0.1872234 0.139474  0.1818544 0.345576 ]
        state after 10 days = [0.07693332 0.09688565 0.3362719  0.24330485 0.24660428]
        state after 1000 days = [8.97378472e-28 1.13004326e-27 6.60595331e-01 3.39404669e-01
 2.87688168e-27]
        state after 1001 day = [8.44851901e-28 1.06389804e-27 6.60595331e-01 3.39404669e-01
 2.70848815e-27]

Question 3
        The state after 1000 and 1001 days shows that there is some change in the probability dis
        Hence these states do not represent the stationary probability distribution of the Markov

Question 4
        The matrix Q is
```

```
[0.002 0.666 0.302]
[0.466 0.102 0.38 ]
[0.11  0.122 0.698]

The matrix R is
[0.01 0.02]
[0.02  0.032]
[0.066 0.004]

The matrix I is
[1. 0. 0.]
[0. 1. 0.]
[0. 0. 1.]

The matrix F is
[4.0279365  4.26333958 9.39240352]
[3.27006043 4.80437252 9.31529737]
[ 2.78814698  3.49371126 10.49546579]

Probability of eventual apsorbtion
[0.74544479 0.25455521]
[0.74359768 0.25640232]
[0.79045644 0.20954356]

Probaility that CI will give a feed back if she enter Major crimes on day one is 0.254555
```

Problem 3 (4 points)

It seems that there is a bug in ***Move-o-Tron 99*** which makes general crimes and feedback units as absorbing states. After updating the software of ***Move-o-Tron 99***, Captain Holt wants to find out the effect that Amy Santiago has on the probability of a person giving feedback. So one matrix is generated including Santiago and the other one without.

Matrix 1 (With Santiago)

	Major Crimes	Traffic	General crimes	Feedback	Theft
Major Crimes	0.002	0.232	0.31	0.434	0.022
Traffic	0.426	0.102	0.222	0.04	0.21
General crimes	0.03	0.11	0.2	0.294	0.366
Feedback	0.003	0.176	0.321	0.3	0.2
Theft	0.11	0.422	0.166	0.1	0.202

Matrix 2 (Without Santiago)

	Major Crimes	Traffic	General crimes	Feedback	Theft
Major Crimes	0.11	0.222	0.092	0.374	0.202
Traffic	0.03	0.11	0.2	0.294	0.366
General crimes	0.002	0.232	0.31	0.434	0.022
Feedback	0.466	0.102	0.02	0.032	0.38
Theft	0.003	0.176	0.321	0.3	0.2

1. How can you find out the effect that Santiago has on the probability of feedback? (1 point)
2. What effect does Santiago have one the probability of getting feedback? (1 point)

   Note: The initial probablity of a person going to a particular unit on a particular day remains the same
3. Name the test Captain Holt is performing. (0.5 points)

Lina Ginetti reports to Captain Holt that the there two kinds of days in the precicnt *"There are normal days and then there are days where workflow is dismal with a tiny dash of pathetic."*. Captain Holt decided to sample the initial probablity of a person going to a particular unit on a good day and a bad day.

4. Without the information about these inital probabilities, can you tell if there is any difference in the probability of getting a feedback? Explain. (1.5 points)

```
# encoding the probabilities as a numpy array
# With Santiago
trans_array_with_amy = np.array([
    [0.002, 0.232, 0.31, 0.434, 0.022],
    [0.426, 0.102, 0.222, 0.04, 0.21],
    [0.03, 0.11, 0.20, 0.294, 0.366],
    [0.003, 0.176, 0.321, 0.3, 0.2],
    [0.11, 0.422, 0.166, 0.1, 0.202]
])

# Without Santiago
trans_array_without_amy = np.array([
    [0.11, 0.222, 0.092, 0.374, 0.202],
    [0.03, 0.11, 0.20, 0.294, 0.366],
    [0.002, 0.232, 0.31, 0.434, 0.022],
    [0.466, 0.102, 0.02, 0.032, 0.38],
    [0.003, 0.176, 0.321, 0.3, 0.2]
])

# ensures that the probabilities sum to 1
assert(trans_array_with_amy[0].sum() == 1.0)
assert(trans_array_with_amy[1].sum() == 1.0)
assert(trans_array_with_amy[2].sum() == 1.0)
assert(trans_array_with_amy[3].sum() == 1.0)
assert(trans_array_with_amy[4].sum() == 1.0)

assert(trans_array_without_amy[0].sum() == 1.0)
assert(trans_array_without_amy[1].sum() == 1.0)
assert(trans_array_without_amy[2].sum() == 1.0)
assert(trans_array_without_amy[3].sum() == 1.0)
assert(trans_array_without_amy[4].sum() == 1.0)

# encoding the initial probability of as a numpy array
state = np.array([[0.3, 0.4, 0.1, 0.15, 0.05]])
assert(state[0].sum() == 1.0)
```

```
# Solution

print('Question 1')
print('\tStationaray distribution of each of the Transition matrix and then comp[aring the probability


print('Question 2')
print('\tVarient including Amy Santiago')
state_1000_with_amy = state
for _ in range(1000):
    state_1000_with_amy = np.dot(state_1000_with_amy, trans_array_with_amy)
print(f'\t\tstate after 1000 days = {state_1000_with_amy.squeeze()} ')
state_1001_with_amy = np.dot(state_1000_with_amy, trans_array_with_amy)
print(f'\t\tstate after 1001 days = {state_1001_with_amy.squeeze()} ')
print('\t\tThe state after 1000 and 1001 days shows that there is some change in the probability distri

print('\tVarient not including Amy Santiago')
state_1000_without_amy = state
for _ in range(1000):
    state_1000_without_amy = np.dot(state_1000_without_amy, trans_array_without_amy)
print(f'\t\tstate after 1000 days = {state_1000_without_amy.squeeze()} ')
state_1001_without_amy = np.dot(state_1000_without_amy, trans_array_without_amy)
print(f'\t\tstate after 1001 days = {state_1001_without_amy.squeeze()} ')
print('\t\tThe state after 1000 and 1001 days shows that there is some change in the probability distri
print(f'\t{state_1000_with_amy.squeeze()[3]} is lesser than {state_1000_without_amy.squeeze()[3]}')
print('\tHence the probability of getting a feedback with Amy Santigo working is lesser than the probab

print('Question 3')
print('\tAB Testing')

print('Question 4')
print('\tYes, as stationary distribution does not depend on the intial states. Stationary Distribution
print('\tHence there will be no changes in the probability of the feedback given.')
```

```
Question 1
        Stationaray distribution of each of the Transition matrix and then comp[aring the probabil

Question 2
        Varient including Amy Santiago
                state after 1000 days = [0.12001975 0.20607988 0.23649985 0.21700347 0.22039705]
                state after 1001 days = [0.12001975 0.20607988 0.23649985 0.21700347 0.22039705]
                The state after 1000 and 1001 days shows that there is some change in the probabil
                Hence these states represents the stationary probability distribution of the Mark

        Varient not including Amy Santiago
                state after 1000 days = [0.14495178 0.16283362 0.18658671 0.26400004 0.24162786]
                state after 1001 days = [0.14495178 0.16283362 0.18658671 0.26400004 0.24162786]
                The state after 1000 and 1001 days shows that there is some change in the probabil
                Hence these states represents the stationary probability distribution of the Mark

        0.21700346817098676 is lesser than 0.2640000386809766
        Hence the probability of getting a feedback with Amy Santigo working is lesser than the p

Question 3
        AB Testing
```

```
Question 4
        Yes, as stationary distribution does not depend on the intial states. Stationary Distribut
        Hence there will be no changes in the probability of the feedback given.
```