# UE20CS312 - Data Analytics

## Worksheet 5 - Markov Chains and AB Testing

PES University

SUNDEEP A , Dept of CSE - PES1UG20CS445

## 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 descrepency. 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)
   ```

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)

In [1]:
```
# Importing Libraries
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
# 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)
```

## Q1.1

Chapman-Kolmogorov Relationship Technique can be used to find the probability of people going to coorect unit to report their crime after N days.
That is p[i]*[p[j]^N]

## Q1.2

the chain can be reduced. we can say this because there are few Zero values in the First Order Markov Chain result. Which means that not all the states are communicable with each other. When all the states are communicable with each other then we can say that they are irreducible, but here since all the states are not communicable we can conclude that the state can be reduced.

## Q1.3

In [3]:
```
#Q3

import numpy as np
matrix_A=np.array([0.3,0.4,0.1,0.15,0.05])
matrix_B=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]
])
matrix_C = matrix_A
n=1002
for i in range(n):
    matrix_C=np.dot(matrix_C,matrix_B)
    if(i==1 or i == 2 or i==1000 or i==1001):
        print("intital probability of a person going to a particular unit after",i,"days :\n",matrix_C)
```

```
intital probability of a person going to a particular unit after 1 days :
 [0.1435072 0.2016392 0.2463988 0.15      0.2584548]
intital probability of a person going to a particular unit after 2 days :
 [0.12810168 0.17477835 0.23000135 0.15       0.31711862]
intital probability of a person going to a particular unit after 1000 days :
 [0.1214373  0.16411091 0.19739717 0.15       0.36705462]
intital probability of a person going to a particular unit after 1001 days :
 [0.1214373  0.16411091 0.19739717 0.15       0.36705462]
```

## Q1.4

The Probability of going to a particular state is same for both day 1000 and 1001.Hence, the Markov Chain Converges or in other words we can say the markov chains has reached a Steady State.
On comparing the initial values and the day 1000 values we can see that the probablity of a person going to a particular unit has increased for **Theft**,**General Crimes**, the probability has remained same for **Feedback** and probability has actually reduced for **Major Crimes**,**Triffic**
</br>

---

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 |
| 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)

1. 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,from%20the%20NumPy%20array%20ndarray%20.&text=Specify%20the%20axis%20(dimension)%20and,a%20slice%20or%20a%20list.

 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]]

 # matrix multiplication or cross pdt C = A*B
 matrix_C = matrix_A @ matrix_B # matrix_C = np.matmul(matrix_A, matrix_B)
```

In [4]:
```
# 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)
```

## Q2.1

Here the Markov Chain is Reducable.
A Markov chain is called irreducible if all states of the chain communicate with each other. All the states can Communicate with each other when there are no Zero values in the Markov state. But we have Zero values in the state so we can conclude that that the Markov Chain is not irreducible(it is reducible)

## Q2.2

In [5]:
```
#Q2
matrix_A = np.array([[0.3, 0.4, 0.1, 0.15, 0.05]])

matrix_B=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]
])



matrix_C = matrix_A
n=1002
for i in range(n):
    matrix_C=np.dot(matrix_C,matrix_B)
    if(i==1 or i == 2 or i==1000 or i==1001):
        print("intital probability of a person going to a particular unit after",i,"days :\n",matrix_C)
```

```
intital probability of a person going to a particular unit after 1 days :
 [[0.1458722 0.1872234 0.139474  0.1818544 0.345576 ]]
intital probability of a person going to a particular unit after 2 days :
 [[0.12555121 0.15840794 0.16748521 0.1921453  0.35641034]]
intital probability of a person going to a particular unit after 1000 days :
 [[8.44851901e-28 1.06389804e-27 6.60595331e-01 3.39404669e-01
  2.70848815e-27]]
intital probability of a person going to a particular unit after 1001 days :
 [[7.95399887e-28 1.00162452e-27 6.60595331e-01 3.39404669e-01
  2.54995126e-27]]
```

## Q2.3

From the above probabilities we can notice that there is a very small difference among the initial probabilities with respect to day 1000 and day 1001.
Hence the state probabilities do not converge till 1001 days. In other words we can say that the Markov chain is not in steady state

## Q2.4

In [6]:
```
# encoding the probabilities as a numpy array
trans_array = np.array([
    [1.0,0.0,0.0,0.0,0.0],
    [0.0,1.0,0.0,0.0,0.0],
    [0.01,0.02,0.002,0.666,0.302],
    [0.02,0.032,0.466,0.102,0.38],
    [0.066,0.004,0.11,0.122,0.698]
])

# 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)
#after changing the matrix to go to the canonical form initial probabilities :
#[0.0,0.0,1.0,0.0,0.0]
# encoding the initial probability of as a numpy array
state = np.array([[0.0,0.0,1.0,0.0,0.0]])
assert(state[0].sum() == 1.0)
```

In [7]:
```
# R and Q matrices
temp = np.delete(trans_array,[0,1],0)
temp = np.delete(temp,[2,3,4],1)

matrix_R= temp
print("Matrix R :")
print(matrix_R)
```

```
Matrix R :
[[0.01  0.02 ]
 [0.02  0.032]
 [0.066 0.004]]
```

In [8]:
```
temp = np.delete(trans_array,[0,1],0)
temp = np.delete(temp,[0,1],1)

matrix_Q= temp
print("Matrix Q :")
print(matrix_Q)
```

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

In [9]:
```
# calculation of fundamental matrix:
matrix_I=np.identity(3, dtype = None)

matrix_F = np.linalg.inv(matrix_I-matrix_Q)
print("fundamental_matrix :\n",matrix_F)
```

```
fundamental_matrix :
 [[ 4.0279365   4.26333958  9.39240352]
 [ 3.27006043  4.80437252  9.31529737]
 [ 2.78814698  3.49371126 10.49546579]]
```

In [10]:
```
#Calculating the eventual probability:
matrix_eventual = np.dot(matrix_F, matrix_R)
print("eventual probability : \n",matrix_eventual)
```

```
eventual probability :
 [[0.74544479 0.25455521]]
 [0.74359768 0.25640232]
 [0.79045644 0.20954356]]
```

Probability that she gives a feedback = matrix_eventual_prob [major crimes][feedback] = [0][1] = 0.2545

## 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.

1. 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)

In [11]:

```python
# 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)
```

### Q3.1

We can perform A/B testing to find out the effect that Santiago has on the probability of feedback.
That is different ppl are taken in two groups A and B and subjected to with and without santiago respectively and found out the effect

### Q3.2

In [12]:

```python
#Q2

#with santigo
m=np.matmul(state,trans_array_with_amy)
print("With Santiago:\n",m)
```

```
With Santiago:
 [[0.17995 0.1689  0.25825 0.2256  0.1673 ]]
```

In [13]:

```python
#without santigo
m=np.matmul(state,trans_array_without_amy)
print("Without Santiago:\n",m)
```

```
Without Santiago:
 [[0.11525 0.1579  0.15765 0.293   0.2762 ]]
```

Santiago's Presence Decreases the feedback.

### Q3.3

the test Captain Holt is performing is Anderson Goodman - Chi Squared Test.

### Q3.4

Without knowing the initial probability there would be a difference in the probability in getting the feedback.
Solution for this is the chi-squared test on the transition matrix and finding the chi squared value and the subsequent p-value to see if the transition matrix is following the markov chain.
we can get the steady state probability matrix and can estimate the probability of getting feedback.
Using this we can perform A/B testing.