Faculty of computers and Artificial Intelligence
Cairo University

Stochastic Models in Operations Research and Decision Support DS361 – Year 3 – Project Code: PID23867710

Prepared for: Dr.Olivia Morad

A proposed Markov Chain Model for Epidemic Propagation Prediction

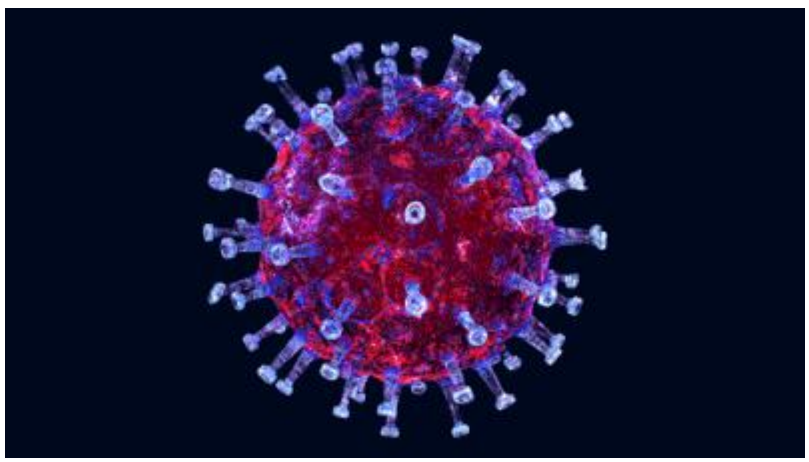
Prepared by:
Nada Abdelrahman Mabrouk … nadaabdelrahman776@gmail.com … 20170315
Athar Atef Hussien … athar.atef.99@gmail.com … 20170005
Khloud Khaled Gomaa … khloudkhaled19@gmail.com … 20170101
Mohamed Safi Ahmed … mody111safi@gmail.com … 20170237

Table of Contents

Chapter 1:

Problem Identification, classification:

COVID-19 is a novel coronavirus that has spread among humans since December 2019, mainly in China. It was first discovered at Wuhan, the Hubei province's capital city. Afterwards the virus spread to other provinces in China. Infection cases have also turned up in other countries. COVID-19 is a respiratory virus which is transmitted through contact with an infected person through droplets when a person coughs or sneezes, or through droplets of saliva.COVID-19 is a Respiratory virus transmitted through contact with an infected person through droplets when a person coughs or sneezes, or through droplets of saliva. The main clinical manifestations of the infection are fever, fatigue, respiratory symptoms, and emergence of dyspnea. Most patients with good prognosis have mild to moderate symptoms but some patients may die or be in critical condition. As of February 15, 2020, 68,500 people have been confirmed with the COVID-19 virus, and 1596 people have died. As the outbreak grows rapidly, India is imposing a more massive quarantine, all schools postponed spring semester commencement. Most companies also postponed the start date of their work after the Spring Festival holiday.



The country has issued restrictions on the entry of Chinese citizens. A considerable number of domestic and international flights were cancelled. There is no question that the spread of this infection with the disease has significantly affected the life, economy and people's health. We need a system or a model to let us predict the spread of this virus in the country to make some decisions based on the result of that system.

At the end we can classify the problem as a situation-based problem.

Problem Description and Statement:

Description: *We need to control the disease spread to evaluate the risk of returning to work at different timing and Provide suggestions on when people can start their routine life. But we cannot predict the Propagation.*

Statement:

We have five W's and H:

First → Who does the problem affect?

Answer: The problem mainly affect India citizens' life.

Second → What are the limits of the problem?

Answer: The whole country.

What is the issue?

Answer: We need to control the disease spread but we cannot predict the Propagation.

Third → When needs to fix the problem?

Answer: It must be fixed in a very short time.

Fourth → where is the problem occurring?

Answer: in India.

Fifth → Why this problem needs to be solved?

Answer: To evaluate the risk of returning to work at different timing and Provide suggestions on when people can start their routine life.

H → How we can solve the problem?

Answer: In the modern world, there are so many methods to solve the complex problem using forecasting approach. Seasonal Autoregressive Integrated Moving Average (SARIMA), Autoregressive Moving Average Model (ARIMA), and Artificial Neural Networks (ANN) are some of the forecasting strategies. Markov chain is an essential tool which has been prepared to solve complex problems, such as the use of peak power. The special case of the stochastic model for this complex problem is Markov chain model. Using probability matrices and Monte Carlo simulation, the 1st order markov models had been used to predict the spread of corona virus.

Chapter 2:
Data Collection and Analysis:
The data of the states of India were collected from the website, especially from the Indian government website as on 30th March 2020.

Table 1

CORONA VIRUS PANDEMIC IN INDIA BY STATE AND UNION TERRITORY AS ON 30TH MARCH 2020

S.No	State or Union territory	Active cases	Recoveries	Deaths	Total cases
1	Maharashtra	215	25	7	247
2	Kerala	202	20	1	223
3	Karnataka	83	5	3	91
4	Delhi	72	6	2	80
5	Uttar Pradesh	72	11	0	83
6	Telangana	70	1	1	72
7	Gujarat	69	1	6	76
8	Rajasthan	60	3	0	63
9	Tamil Nadu	67	4	1	72
10	Madhya Pradesh	39	0	2	41
11	Jammu and Kashmir	41	1	1	43
12	Punjab	39	1	1	41
13	Haryana	35	17	0	52
14	West Bengal	22	1	0	23
15	Andhra Pradesh	23	1	0	24
16	Bihar	15	0	1	16
17	Ladakh	13	3	0	16
18	Andaman and Nicobar island	10	0	0	10
19	Chandigarh	9	0	0	9
20	Chhattisgarh	7	0	0	7
21	Uttarakhand	7	2	0	9
22	Goa	5	0	0	5
23	Himachal Pradesh	3	1	1	5
24	Odisha	3	0	0	3
25	Manipur	1	0	0	1
26	Mizoram	1	0	0	1
27	Puducherry	1	0	0	1
	Total	1184	103	27	1314

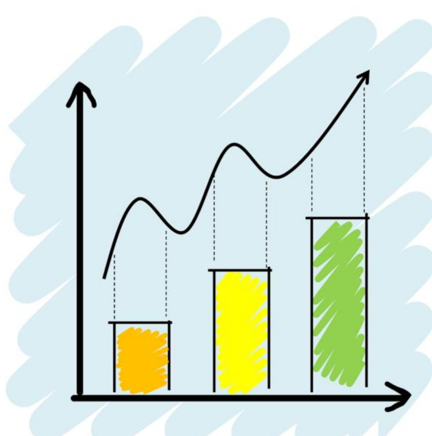
For the Corona pandemic COVID-19 we did some statistical analysis such as descriptive statistics, mean, median, mode, range, standard deviation, skewedness and kurtosis.

Mean, median, mode, range, standard deviation and variance are focused on the general public residing in India at some stage in the COVID-19 pandemic, and the dearth of symmetry to represent the extent of COVID-19 is highlighted in the skewness and kurtosis.

	Active cases	***Recoveries***	***Deaths***
N	*27*	*27*	*27*
Range	*214*	*25*	*7*
Minimum	*1*	*0*	*0*
Maximum	*215*	*25*	*7*
Sum	*1148*	*103*	*27*
Mean	*43.85*	*3.81*	*1.00*
Std. Deviation	*54.700*	*6.645*	*1.776*
Variance	*2992.131*	*44.157*	*3.154*
Skewness	*2.141*	*2.223*	*2.492*
Kurtosis	*4.820*	*4.214*	*6.140*

Table 2

DESCRIPTIVE STATISTICS OF CORONA VIRUS INFECTION IN INDIA AS ON 30TH MARCH 2020



Chapter 3:

Modeling and Formulation:

In this study, there is three states of the problem. The states are defined as 1) Infected, 2) Recovered, 3) Dead.

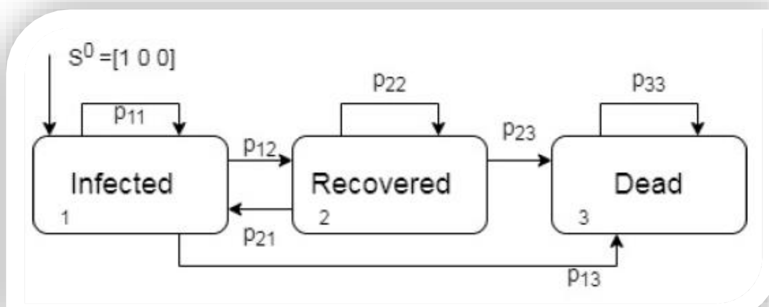


Figure 1-Illustration of the transition probabilities of the proposed model

Let S_0, S_1, S_2, S_3…… S_n be a random variables with times

T_0, T_1, T_2, T_3……… T_n is stated to be markov method and it's satisfy the following property:

$P_{ij}(t) = P\,[S_{t+1} = j \mid S_t = i]$ where $i, j \rightarrow S$ and $t = 0, 1, 2$…….. (I)

The above belongings is also referred to as one-step transition possibility from one state i at t-1 to t. i.e. P_{ij}.

Transition probabilities are allowed to be time – dependent, hence The need to Express $P_{ij}(t)$ as a function of the time instant t.

$0 \le Pij \le 1$; *Where, both i. j = 1, 2, 3……n and*

$$\sum_{j=1}^{n} Pij = 1 \quad (II)$$

$$Pij = Pij + \sum Pij\,; \quad for\ i,j = 1,2,3,4 \ldots\ldots\ldots..n \quad (III)$$

The first order markov chain of the probability transition matrix P is,

$$P = \begin{matrix} P11 & P12 & P13 \\ P21 & P22 & P23 \\ P31 & P32 & P33 \end{matrix}$$

The Transient Analysis:

$$\pi^n = \pi^0 * P^n \ldots\ldots\ldots$$

Where: π is the Probability distribution of states.

$\pi_i^n = P\{S_n = i\}$ *, where π_i^n is the Probability of being in state i at time n.*

$$\pi^n = [\pi_0^n, \pi_1^n, \pi_2^n, \ldots\ldots.]$$

Chapter 4:

Solution to the Model and Results analysis:

In order to generate the corona virus sequences (COVID-19) states, the starting state is that we say I arbitrarily selected. The random values between 0 and 1 were generated using random number generator. This is also referred to as Monte Carlo simulation.

State	Status	Frequency
1	*Active Cases*	*1184*
2	*Recoveries*	*103*
3	*Deaths*	*27*
Total		*1314*

Table 3 -FREQUENCY DISTRIBUTION OF CORONA VIRUS INFECTION AS ON 30TH MARCH 2020

The probability transition matrix P describes the Markov chain representing three states of corona virus (COVID-19) infection was obtained from the first order markov chain transition matrix,After some Calculations we get :

$$P = \begin{matrix} 0.9331 & 0.0483 & 0.0186 \\ 0.6628 & 0.2791 & 0.0581 \\ 0.4681 & 0.1064 & 0.4255 \end{matrix} \quad \text{(VI)}$$

And the markov chain transition matrix after 30 days we can get P^{30}*as:*

$$P^{30} = \begin{matrix} 0.89878701 & 0.06549072 & 0.03572228 \\ 0.89878701 & 0.06549072 & 0.03572228 \\ 0.89878701 & 0.06549072 & 0.03572228 \end{matrix} \quad \text{(VII)}$$

At the end we can get the The Transient Analysis as:

$$\pi^{30} = \pi^{0} * P^{30} = [0.89878701, 0.06549072, 0.03572228]$$

$$S = \begin{cases} Infected\ if\ \ v > 0.89878701 \\ Recovered\ if\ 0.03572228 < v < 0.06549072 \\ Dead\ if\ v < 0.03572228 \end{cases} \quad \text{(VIII)}$$

Where v is a random wide variety of uniform distribution.

As a markov model and Transient Analysis results we got that:

1 - The range of corona between 1 and 215 within the active stage with mean 43.85 and the standard deviation is 54.7.

2 - The skewness represents that the dearth of symmetry (i.e., 2.141) and the kurtosis is more than 3.

3 - The transition chances illustrates that, there is 89.8% chance in the first level

That (Active cases/Infected people) of corona number will increase in India which is a bad news.

4 - In the second level (Recoveries state) 6.6% chances are there to increase Recoveries people.

5 - In the third level (Death state) 3.6% of probabilities of there for the given statistics and it's a good thing that the death rate will be decrease in india.

From The markov model and Transient Analysis, the alternate of the corona virus (COVID-19) in which generated as the random range adjustments values and assumes the distinct states of corona infection.

Chapter 5:
Conclusion and Future Work:

We have developed a model that allows the study of an infection spreading in a society. We solved this model and calibrated it to India, and this model was fully assisted in capturing the Covid-19 pandemic dispersion's general behavior.

This study results revealed that markov chain is beneficial in simulating the corona infection in numerous stages. This type of analysis could be very much useful in generating the time period of corona virus infection. The corona infection assessment indicates that the markov chain approach provides us with modeling opportunity in future.

References

- Farhaoui, Y. (2020). Yousef Farhaoui Editor.Markov Chain Prediction
- Operations Research: Applications and Algorithms
- Statistical Inference Regarding Markov Chain Models
- A Markov Model for Prediction of Annual Rainfall
- Introduction to Markov chains
- Predicting the Weather with Markov Chains
- Markov Chain Monte Carlo Simulations and Their Statistical Analysis
- IndiaFightsCorona COVID-19

Appendix A – The Executable System/ Program Implementation

```
import numpy as np
import random as rm
import quantecon as qe
import tkinter
from tkinter import *
from tkinter import simpledialog
from tkinter import messagebox


root = tkinter.Tk()
root.geometry("300x300")
root.withdraw()
#############################################################################
######
def get_trans_matrix():
    #Function to get the probabilities and make the transistion matrix
    p = np.zeros((3,3),dtype=float)  # Initialize the transistion matrix
    for i in range(3):
        pi = p[i]
        for j in range(3):

            pi[j] = simpledialog.askfloat(title="MarkovChain
Prediction",prompt="Enter P" + str(i+1) + str(j+1) + " : ",minvalue=0.0,
maxvalue=1.0,parent=root)



        if((np.sum(p[i])) != 1.0):
            while(np.sum(p[i]) != 1.0):
                messagebox.showinfo("MarkovChain Prediction","Wrong Sum of
all Pi must be equal  1.!.!.Re Enter it please:")

                for j in range(3):
```

```
                    pi[j] = simpledialog.askfloat(title="MarkovChain Prediction",prompt="Enter P" + str(i+1) + str(j+1) + " : ",minvalue=0.0, maxvalue=1.0)


    return p


###############################################################################################
def check_MC_reducible(transitionmatrix): #check this markov chain is irreducible or reducible
    mc = qe.MarkovChain(transitionmatrix, ('infected','recovered', 'dead'))
    if(mc.is_irreducible):
        messagebox.showinfo("MarkovChain Prediction","This MC is irreducible")
    else:
        messagebox.showinfo("MarkovChain Prediction","This MC is reducible")
############################################################################################
def check_periodicity(transitionmatrix): # check the periodicity of the markov chain and the states
    mc = qe.MarkovChain(transitionmatrix)
    if(mc.is_aperiodic):
        messagebox.showinfo("MarkovChain Prediction","This MC is aperiodic and all states are aperiodic ")

    else:
        period = mc.period
        messagebox.showinfo("MarkovChain Prediction","This MC is periodic and all states are aperiodic with period = " + str(period))
#############################################################

def get_reachable_states(transitionmatrix): # Getting The reachable states and the not reachable
    for i in range(3):
```

```
        pi = transitionmatrix[i]
        for j in range(3):
            if(pi[j] > 0.0): # A state j is said to be [reachable] from a
state i if Pij > 0 for some n = 1, 2, ...
                messagebox.showinfo("MarkovChain Prediction","State " +
str(j) + " is reachable from state "+ str(i))
            else:
                messagebox.showinfo("MarkovChain Prediction","State " +
str(j) + " is not reachable from state "+ str(i))
##############################################################################
########
def get_absorbing_states(transitionmatrix): # Getting The absorbing states
and the not absorbing
    for i in range(3):
        pi = transitionmatrix[i]
        if(pi[i] == 1.0): #A state i is said to be absorbing if it forms a
single element closed set. If i is an absorbing state we have Pij =1
            messagebox.showinfo("MarkovChain Prediction","State " + str(i) +
" is an absorbing state")
        else:
            messagebox.showinfo("MarkovChain Prediction","State " + str(i) +
" is not an absorbing state")
############################################################################
def get_recurrent_transient_states(transitionmatrix): # Getting The recurrent
states and the transient
    for i in range(3):
        pi = transitionmatrix[i]
        if(pi[i] == 1.0): # A state i is said to be recurrent if Pi = 1
            messagebox.showinfo("MarkovChain Prediction","State " + str(i) +
" is recurrent state")
        elif(pi[i] < 1.0): # and if Pi < 1, The State is transient state
            messagebox.showinfo("MarkovChain Prediction","State " + str(i) +
" is transient state")
##############################################################################
```

```
def prediction(n,bi0,transitionmatrix,states): # Function to predict the next state
    # n : the period we want do predect the next state after it(n days, n months, n years...etc)
    # bi0 : The intial Bi
    # states : The states names
    transitionmatrix_n = transitionmatrix
    if(n == 1): # if the period is 1 so we already have the transitionmatrix of P
        transitionmatrix_n = transitionmatrix
    else: # if the Period more than 1 so we need to calculate the P power n
        for i in range(n):
            if(i == 0):
                transitionmatrix_n = transitionmatrix
            else:
                transitionmatrix_n = np.matmul(transitionmatrix_n, transitionmatrix_n)
    messagebox.showinfo("MarkovChain Prediction","P after "+str(n)+ " days is:\n "+str(transitionmatrix_n))
    bi_n = np.matmul(bi0, transitionmatrix_n) # calculate Bi power N
    messagebox.showinfo("MarkovChain Prediction","bi after "+str(n)+ " days is: " + str(bi_n))
    messagebox.showinfo("MarkovChain Prediction","The next state of the virus after " + str(n) + " days will be: ")
    for i in range(3):
        messagebox.showinfo("MarkovChain Prediction",str(i+1)+"- " + states[i] + " with probability " + str(bi_n[i]))
    messagebox.showinfo("MarkovChain Prediction","So with this probabilities we can predict that the number of " + states[np.argmax(bi_n)] + " people will be increased in the next " + str(n) + " Days") #Predict the next state
#############################################################################################

states = ["infected","recovered","dead"] # states names
bi0 = np.zeros(1)
bi0 = [1.0,0.0,0.0] # Initial Bi
```

```
p  = get_trans_matrix() #Getting the transistion Matrix
messagebox.showinfo("MarkovChain Prediction","The Transition matrix
is:\n"+str(p))
check_MC_reducible(p)
check_periodicity(p)
get_absorbing_states(p)
get_reachable_states(p)
get_recurrent_transient_states(p)
n = simpledialog.askinteger(title="MarkovChain Prediction",prompt="Enter The
period of time you want to predict the state after it")
prediction(n,bi0,p,states)
#############################################################################
##########
```

The python file of the code:

[Final project python Code file]   Note: Click on the link to download the code file.