Modelling and analysis of COVID-19 epidemic in India with graphs

Arvind Nishanth Srivathsan Sukumar Thyagaraajan

I. Introduction

COVID-19 is a disease caused by a novel virus called SARSCoV-2 (severe acute respiratory syndrome coronavirus 2) which has spread over more than 200 countries infecting more than 42,966,344 people worldwide (as of 26 October 2020) . The outbreak of this disease was announced by World Health Organization (WHO) after one month from the reporting of first case on Dec. 31, 2019, in Wuhan, China and later as a pandemic on March 11, 2020 [2]. India is also in the thick of this highly infectious viral disease with over 7,909,959 (as of 26 October 2020) people caught in its tentacles [2].

The symptoms of COVID-19 range from mild to severe, which are indicated by mainly fever, cough, and respiratory distress[2]. The two most important modes of transmission of corona virus are respiratory droplets and contact transmission (contaminated hands) with an incubation period of 2-14 days [2] . The virus spread rapidly around the world and several large-size clusters of the spread have been observed worldwide including outbreaks in China, USA, Spain, Russia, UK, Italy and India [2].

In this paper we present a mathematical model using graphs that simulate the spread of this virus on a small scale. This model helps us analyse the effectiveness of social distancing, wearing masks in public and the use of sanitizers. We have used data published in research papers[2] to help improve our model.

II. Related work

The SIR model is one of the simplest and most used models to simulate disease outbreaks[3]. The SIR (susceptible-infected-removed) model,developed by Ronald Ross1, William Hamer, and others in the early twentieth century[4], consists of a system of three coupled non-linear ordinary differential equations, which does not possess an explicit formula solution. Simple tools from calculus allow us to extract a great deal of information about the solutions. A lot of diseases have been modeled this way [5],[6],[7].

III. Methods

In this section, we briefly discuss the method employed in this paper, including a definition of the basic reproduction number.

Identify applicable funding agency here. If none, delete this.

IV. Basic Reproduction Number

The basic reproduction number is defined as the number of secondary cases that one infected primary subject causes on average in an uninfected and totally susceptible population, over the infectious period[8-10]. In this paper we will not calculate the value of R_0 instead using publicly available data [2] and research material available to us [1] for this model. Data that is being used is limited to India.

V. Brief Overview

After calculating the value of R_0, we generate a random graph using the secrets module in python which is a strong pseudo-random number generator. In this graph, the vertices represent people and edges between vertices represent contact between the two persons. We then infect one person/node and see the transmission of virus from the single node. The factors that affect the transmission rate are mask, sanitization and social distancing. The number of nodes then infected depends on the Basic Reproduction Number R_0.

The value of R_0 is determined using the normal distribution. The normal distribution is constructed using the mean and variance calculated from the research article[1]. Each iteration is counted as one day so the program calculates the number of days required to completely infect every node that is the part of the connected graph.

VI. Graph Definition

A. Adjacency matrix for Non cautious community

1) Pseudo code:

- for i in range(0,population):
  for j in range(i,population):
  normal_community[i][j]=normal_community[j][i]
  = secrets.randbelow(2)
  if(i==j):
  normal_community[i][j] =normal_community[j][i] = 0

  Since You Can't Transmit Disease to your self so Matrix at i==j is zero

B. Graph 1:Non cautious community

Graph is defined as $V : \{i\ for\ 0 <= i < n\}$ i.e the labelled vertices correspond to people (0,1,2,3...)
Two vertices have an edge if they are related i.e. they come in contact with each other. This is got from the randomly generated adjacency matrix from the secret module in Python $E = \{e_{ij}\ exists\ from\ v_i\ to\ v_j if A[i][j] = 1 \& A[j][i] = 1\ \}$

C. Graph 2:Cautious community

The Weighted Graph is defined as $V : \{i\ for\ 0 <= i < n\}$ i.e the labelled vertices correspond to people (0,1,2,3...)
Two vertices have an edge if they are related i.e. they come in contact with each other. This is got from the randomly generated adjacency matrix from the secret module in Python $E = \{e_{ij}\ exists\ from\ v_i\ to\ v_j if A[i][j] = 1 \& A[j][i] = 1\ \}$

D. Edge weight for Cautious community

In our algorithm the weightage for an edge is given 3 factors:wearing masks,social distancing and hand sanitization The Edge value lies from [0,1] Lower the value lesser probability of getting infected.

Table		
Precaution	Population Percentage	Effectiveness
Masks	40%	50%
Social Distancing	20%	90%
Sanitization	40%	60%

According to Table using secret module that much percentage of people are assigned with those edge weight values

VII. How R_0 is calculated for each Vertex

Theoretical R_0 definition:R_0 tells you the average number of people who will contract a contagious disease from one person with that disease.
In our model :Since in real world we don't know how many people a person interacts with, We use the R_0 value.
So we use the Gaussian distribution concept for such random event with mean(μ) $= 2.06$ and $\sigma = 2.04$ is the variance around the mean. Refer Fig1.
For the code R_0 is the value taken randomly taken weighted around the mean from the standard normal distribution($N(\mu, \sigma)$). Refer Fig 2

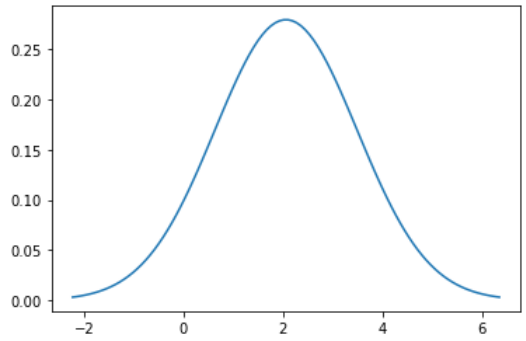

Fig. 1. Normal distribution ($N(\mu, \sigma)$)

Table 1. The various parameters calculated using SIR model. R_0, β, γ, $C_{[end]}$ and $S_{[end]}$ represent basic reproduction number, average contact frequency, average removal frequency, epidemic size and final number of susceptible individuals left.

India and State	R_0	β	γ	$C_{[end]}$	$S_{[end]}$	Outbreak (year-2020)	Start of acceleration (year-2020)	Start of steady growth (year-2020)	Start of ending phase (year-2020)	End of epidemic (1 case) (year-2020)
India	1.15	0.66	0.573	176126	523659	03-Mar	23-Apr	10-Jun	04-Jul	16-Oct
Maharashtra	1.11	1.008	0.908	55631	234823	11-Mar	25-Apr	05-Jun	26-Jun	09-Sep
Gujrat	1.58	0.282	0.179	16409	9558	20-Mar	19-Apr	01-Jun	22-Jun	13-Sep
Tamil Nadu	3.1	0.371	0.12	12555	701	18-Mar	03-May	20-May	29-May	29-Jul
Delhi	1.19	0.486	0.409	23635	55146	03-Mar	26-Apr	19-Jun	16-Jul	15-Oct
Rajasthan	1.72	0.269	0.156	4900	2080	03-Mar	09-Apr	19-May	08-Jun	13-Aug
Madhya Pradesh	1.68	0.347	0.206	4053	1866	22-Mar	09-Apr	11-May	27-May	16-Jul
Uttar Pradesh	1.77	0.273	0.154	4531	1748	04-Mar	11-Apr	19-May	07-Jun	10-Aug
West Bengal	1.96	0.18	0.092	17522	4751	19-Mar	11-May	01-Jul	26-Jul	24-Nov
Andhra Pradesh	1.12	0.979	0.874	2500	9462	19-Mar	09-Apr	17-May	05-Jun	15-Jul
Punjab	1.28	0.797	0.621	3237	4708	18-Mar	27-Apr	21-May	02-Jun	01-Jul
Telangana	2.44	0.298	0.122	1180	153	14-Mar	31-Mar	26-Apr	09-May	22-Jun
Jammu and Kashmir	1.41	0.275	0.195	1330	1211	13-Mar	06-Apr	28-May	24-Jun	19-Aug
Karnataka	1.09	0.789	0.723	1216	6031	12-Mar	03-Apr	30-May	28-Jun	16-Aug
Bihar	7.92	0.18	0.023	823	0	23-Mar	18-Apr	15-May	28-May	17-Jul
Kerala	2.55	0.271	0.106	494	55	05-Mar	19-Mar	16-Apr	30-Apr	06-Jun
Odisha	1.74	0.19	0.109	10921	4449	21-Mar	28-May	22-Jul	19-Aug	05-Dec
Jharkhand	1.86	0.298	0.16	191	61	02-Apr	14-Apr	16-May	31-May	22-Jun
Tripura	1.87	2.101	1.123	148	47	10-Apr	06-May	11-May	13-May	16-May
Uttarakhand	1.03	2.264	2.204	71	1071	20-Mar	24-Mar	06-May	27-May	04-Jun

Fig. 2. Data for R_0

VIII. Infection And Day-wise Spread for Non-cautious community

We start by infecting v_0. We define a set called $TI = \{v :\ v\ is\ infected\}$ Initially only vertex 0 is infected $\implies TI = \{v_0\}$
Define another set $M : \{v_1, v_2, ..., v_n\}$
Algorithm:
Until TI==M or we get the maximal Graph with V=TI
For each vertex in TI:
Infect its Neighbours
Have a count of the number of iterations.
The number of iterations = No of days

IX. Infection And Day-wise Spread for Cautious Community

We start by infecting v_0. We define a set called $TI = \{v :\ v\ is\ infected\}$ Initially only vertex 0 is infected $\implies TI = \{v_0\}$

Define another set $M : \{v_1, v_2, ..., v_n\}$

Algorithm:
Until TI==M or we get the maximal Graph with V=TI
For each vertex in TI:
Infect its Neighbours **if** Probability of infection is **True**
Have a count of the number of iterations.
The number of iterations = No of days

X. Example data set and understanding the output:

Lets start with a small example so that data set can be viewable and understandable.Take n=5.Check Fig 3.
If we run The non cautious community algorithm the entire population gets affected in **1 day**.

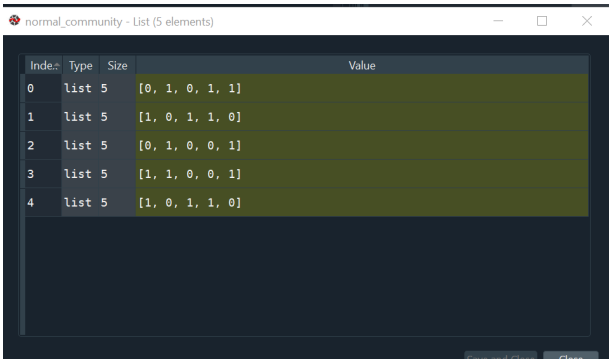

Fig. 3. Adjacency matrix:

For Cautious community lets include the edge weightage in the adjacency matrix for comfort. Check Fig 4.

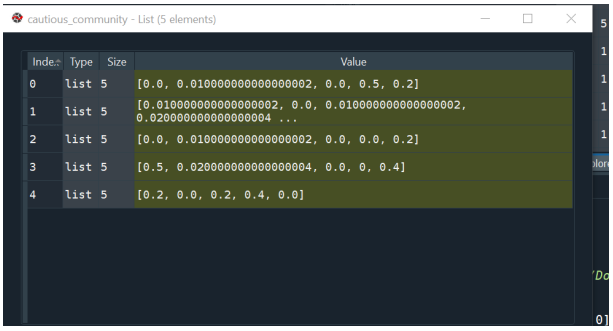

Fig. 4. Matrix with edge weights(cautious community)

If we run The Cautious Community algorithm the entire population gets affected in **11 days**.
We can see the effect of following precautions even if followed partly.

Now Lets see for a bigger data set but since we can't visualize the matrix we can draw,
A plot of **infected people** vs **no of days**
N=1000

(Refer Fig 5 and Fig 6)

A. Remarks

For n=100 The non Cautious community took 12 days and The Cautious community took 50+days. This is a huge increase and slows down the exponential growth.

XI. Parallel edges and cycles

In randomly generated graph G ,the nodes represent the people and the edges represents recent physical contact with each other.Since an infected person cannot be re-infected (since we did not take into account the fact that a person who is infected is later cured), The spread of infection in G will end up being a tree so the discussion on cycles and parallel edges becomes irrelevant.

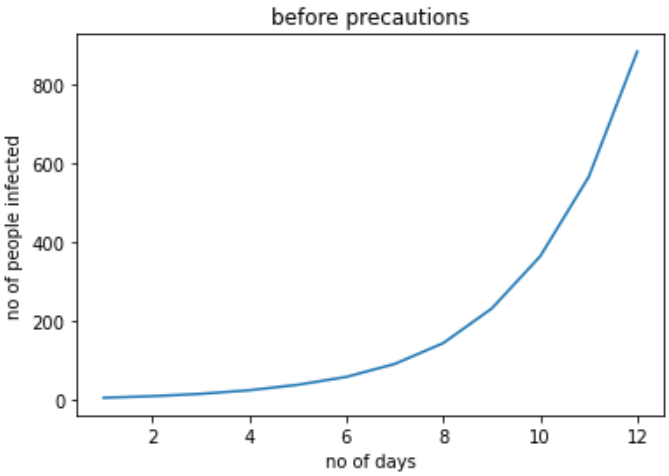

Fig. 5. Non cautious community spread

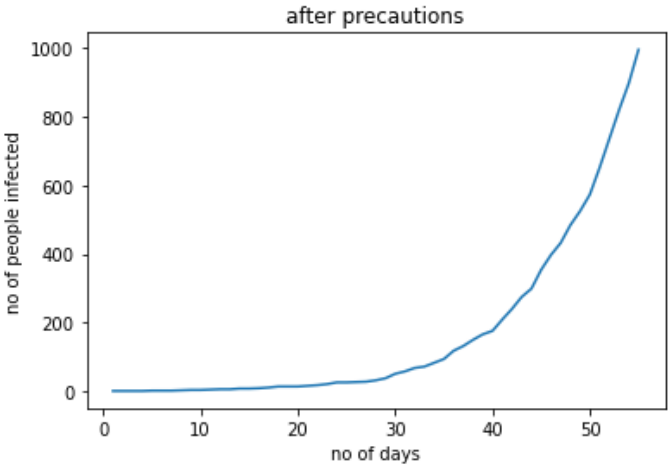

Fig. 6. Cautious community spread

XII. Longest Path

the spread of infection modelled as a graph is an undirected tree, the longest path between any two nodes of the tree is the diameter of the tree. The algorithm for finding the diameter of a graph is (BFS method) :
Step 1: Start a breadth first search from any vertex 'V1' and find a vertex 'V2' with the longest distance from 'V1', 'V2' is an end vertex of the longest path in the tree Step 2: Start a breadth first search from 'V2'. The end-point 'V3' of this breadth first search will be the other end-vertex of the longest path on the graph G.

For viewing the algorithm in Python Click here: Longest Path Algorithm.

XIII. Eigenvector Centrality

Eigenvector centrality is a measure of the influence of a node in a network. Relative scores are assigned to all nodes

in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. A high eigenvector score means that a node is connected to many nodes who themselves have high scores.
Click here for the link to the eigenvector centrality algorithm.
Upon running for limited number of nodes as in our model, it can be easily found that the first node is the node with the highest eigenvector centrality.
Logically, it can be argued that the first node contributes the most to the spread of the virus as it influences a large number of nodes and hence the consequential result.

XIV. Small World phenomenon

The small-world phenomenon – the principle that we are all linked by short chains of acquaintances -is fundamental in social networks
In informal terms For e.g You and Donald trump might be related within like say 5-6 people.
In terms of our graph in COVID for any two vertices might have a path within length of 3-8.
The path length is logarithmic to number of people.This is due to tree like structure of the virus which is related to R_0 value.
In our plot for 100 people we could see that plot 1 with no precautions people got infected in 6-7 days.
The day count is our largest path length between 2 vertices.
In the pre-cautious community graph i.e plot 2 the number of days must increase-we got 47 days for that particular example
The sample set consisted of 100 people. For the given sample set, the adjacency matrix is a matrix consisting of 100 rows and 100 columns and it becomes hard to visualise. As a result we use the plot to keep track of the number of people who have been infected vs the day count. Notice that we expect as per WHO's claim that a significant amount of population wearing masks, maintaining social distancing and using sanitizers significantly impacts the spread and the curve takes a much longer time to rise. The reason for the curve not flattening out even in the second case is due to the lack of recovery factor. This implies that an infected person will infect a fellow individual at some point of time.

XV. Limitation of the graph model

In this model we assume that each person has the same age and has no previous medical complications so it is not a very accurate representation of the of the general populace. We could then take local data that indicates the age and count of the population. We have also avoided the fact that people can get reinfected and that they get recovered. We are running the probability function until the person can get infected.This is not not completely true because of the fact that it could be larger than recovery time But if we try to include these in our model then the time to simulate the model becomes very large and It is difficult to do this with the limited computing resources that we have.

XVI. Conclusion

Modelling and analysis of COVID-19 data has led to significant breakthroughs in formulating effective management strategies at the height of the pandemic crisis. Forming graphs is an effective way of processing data and deriving meaningful inferences from it. The ever-increasing movement of people has facilitated the spread of the virus to an alarming extent. This project illustrates the importance of employing methods such as social distancing and, the usage of masks and hand sanitizers and how they consequentially retard the spread of the infection keeping in mind the WHO's mandate on the usage of such methods. By assigning effectiveness values based on the method employed, we are translating the real-world effectiveness of each method into this project. By this model we can make a compelling argument for the increased use of social distancing norms and sanitization measures to effectively combat the spread of the virus. Strict use of such methods are the need-of-the-hour until a vaccine is approved. More research and predictive models can provide further insights and can result in interesting and novel approaches towards managing this pandemic-induced halt of the entire globe.

Acknowledgment

We are extremely grateful to our professor, Dr R.S.Lekshmi for giving us the golden opportunity to do this interesting project on the topic, "Modelling and analysis of COVID-19 epidemic in India with graphs". This project would not have been possible without her valuable teachings.

We would also like to thank our friends, whose competitive nature helped us set high expectations for ourselves.

References

[1] Mukesh Jakhar1, P. K. Ahluwalia2 and Ashok Kumar1 - COVID-19 Epidemic Forecast in Different States of India using SIR Model.
[2] https://covid19.who.int/ WHO data on COVID – 19 pandemic.
[3] Hyuk Jun Chang - Estimation of basic reproduction number of the Middle East respiratory syndrome coronavirus (MERS CoV) during the outbreak in South Korea, 2015.
[4] Heesterbeek JAP - A brief history of R_0 and a recipe for its calculation. Acta Biotheor. 2002;50(3):189–204
[5] Thomas H Cormen, Charles E Leiserson, and Ronald L Rivest, "Introduction to Algorithms", MIT Press, 2015.
[6] Jennifer Golbeck, "Eigenvector Centrality", https://www.sciencedirect.com/topics/computer-science/eigenvector-centrality, 2013.