Modeling Statistical phenomena
Assignment 2

Ali Setareh Kokab (95100491)

Department of Physics, Sharif University of Technology

April 9, 2021

Contents

1 Summary

In this assignment, we will study the network-based SIR model in which infection is transmitted through a network of contacts between individuals. We also try to find the parameters' numerical values in the SIR model's solution (page 23) [1] for weekly deaths data from the plague in Bombay. Finally, we review some articles from recent literature about models used to model the COVID-19 pandemic.

2 SIR on lattice

In this exercise, we make a square lattice and run a SIR model on it.

2.1 Build a lattice

The network that we consider here is a two-dimensional square lattice in which each node is connected with its four nearest neighbors. We also use periodic boundary conditions. To make our lattice, we used `lattice.grid_graph` function from the Python NetworkX library [2] with periodic parameter set true.

2.2 SIR on lattice

To simulate SIR on the lattice we have written two main functions. The first function is the `init` function, which is responsible for producing the initial condition for us. It gets the total number of nodes (N) and the number of initially infected nodes (k). In this function for each node, we define two attributes representing the node today condition and tomorrow condition, respectively. By "condition," we mean susceptible (S), Infected (I), and recovered (R). In this function, we choose k nodes randomly and set both of their attributes "I."
The function `sir` is responsible for time evolution of the lattice. It gets the simulation's number of time steps and probability to get infected (p) and recovered (r). We loop through infected nodes and produce four random numbers for each of their neighbor nodes in each time step. If the random number is less than p, we change the "tomorrow" attribute of that node from "S" to "I." Moreover, we loop through every node that their "today" attribute is infected and produce a random number for each of them. If this random number is less than r, we change the "tomorrow" attribute of that node from "I" to "R." Eventually, we set the "today" attribute of changed nodes to their "tomorrow" attribute and go to the next time step. This function runs until there is no infected node in the system. This function returns three lists containing number of the susceptible individuals, infected and recovered individuals in each time step.
Using random numbers in our model we will see a different result, every time we run our code. To draw conclusion from our simulation, we should run our code several times and report the **average** of our outputs.

2.3 Results

In fig 1 you see the histogram of the R(∞) for p=0.25 and r=1. For plotting this graph, we have ran our code two thousands times in a for loop. As you can see, the final number of recovered nodes in most executions is less than ten. This result is expected as the probability of getting infected is much less than the probability of getting recovered. By looking at the plot of I versus time (figure 2), we see that in this scenario we have endemic.

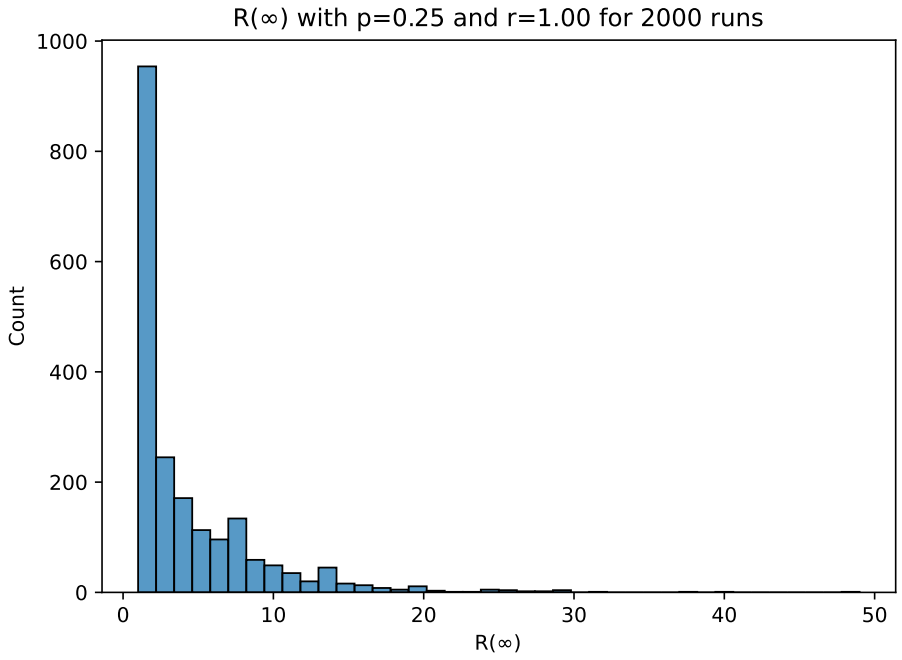


Figure 1: Histogram of R(∞) for 16×16 lattice with p=0.25 and r=1.

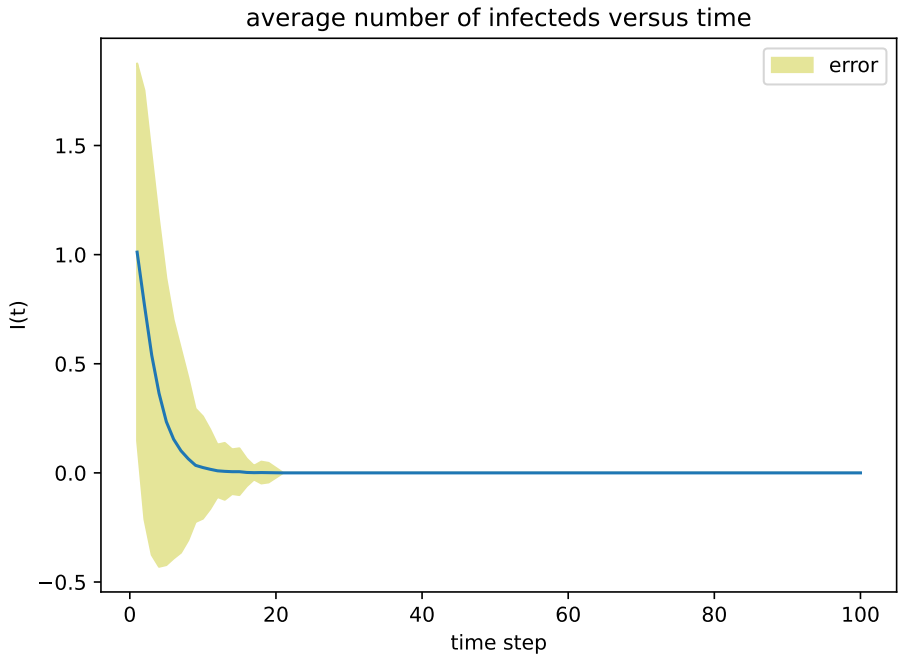


Figure 2: Average number of the infected individuals versus time for p=0.25 and r=1. Each point is averaged over 2000 ensembles. As it is seen, we are facing endemic here.

We run our program for p=0.4, 0.5, 0.6, 0.7. In fig 3 you see the histograms of R(∞) for these values of p. As it is seen, for p=0.4, most R(∞) falls between 0 and 50. By comparing with I(t) for this value of p (figure 5), we see that in p=0.4 we are close to the phase transition from endemic to pandemic. By raising the value of p, we see that in fig 3, the histograms' maximum move towards higher values as we expected. For p=0.5 we see that we have almost two peaks. One is about 10 and the other is about 220. The reason behind this is the stochastic behavior of our system. Unlike the differential equations that specify the system's output in a deterministic way, here we use random numbers which leads to different outputs every time we run our code. Therefore, the first peak in this histogram comes from the fact the infection dies very soon that in some of our ensembles the disease dies very soon, but it could affect most of our population in the other ensembles. We can also see this fact in fig 6 by the band around each curve, representing the standard deviation of that point. However, when we increase p above 0.5, we could clearly see that there is only one peak in 3. As it is shown, in this scenario the infection reach to nearly every individual. In fig 5 you see the average number of the infected individuals in time. By increasing the value of p, we see that disease's peaks happen sooner

and the maximum number of the infected individuals increases.

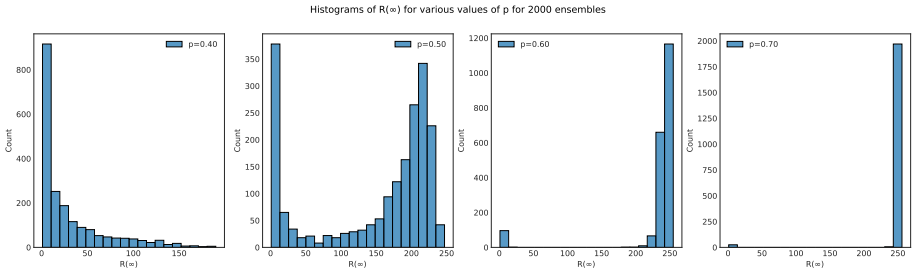


Figure 3: Histograms of R(∞) for various values of p.

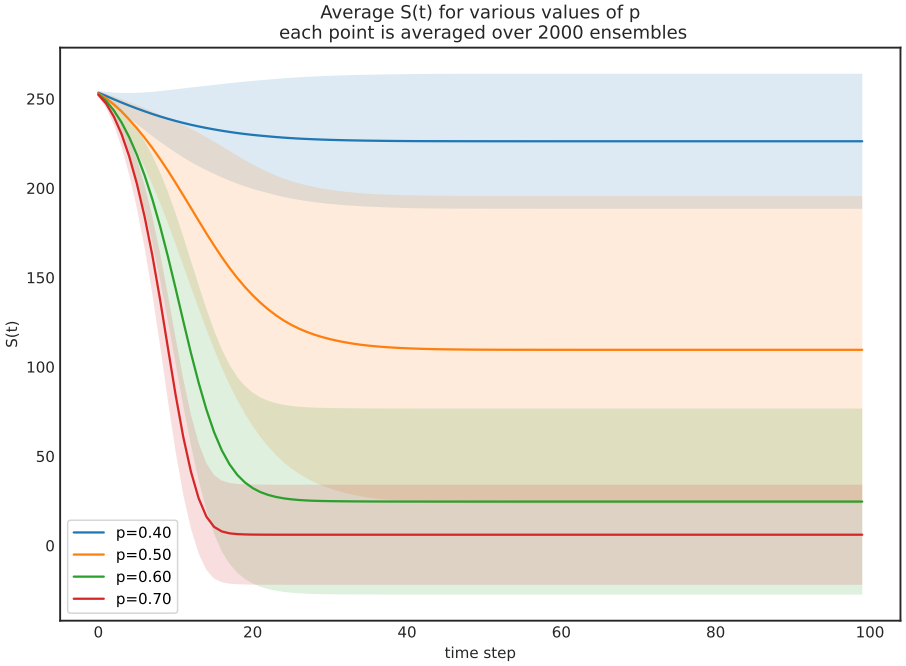


Figure 4: Average values of S(t) for different values of p for a 16 $\times$ 16 lattice. Each point is averaged over 2000 ensembles

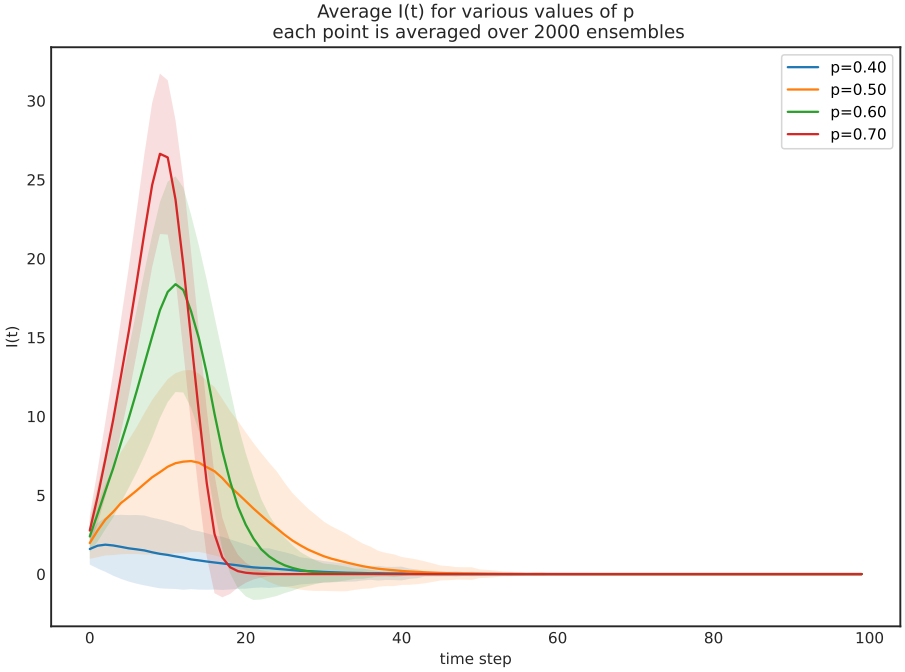


Figure 5: Average values of I(t) for different values of p for a 16 $\times$ 16 lattice. Each point is averaged over 2000 ensembles

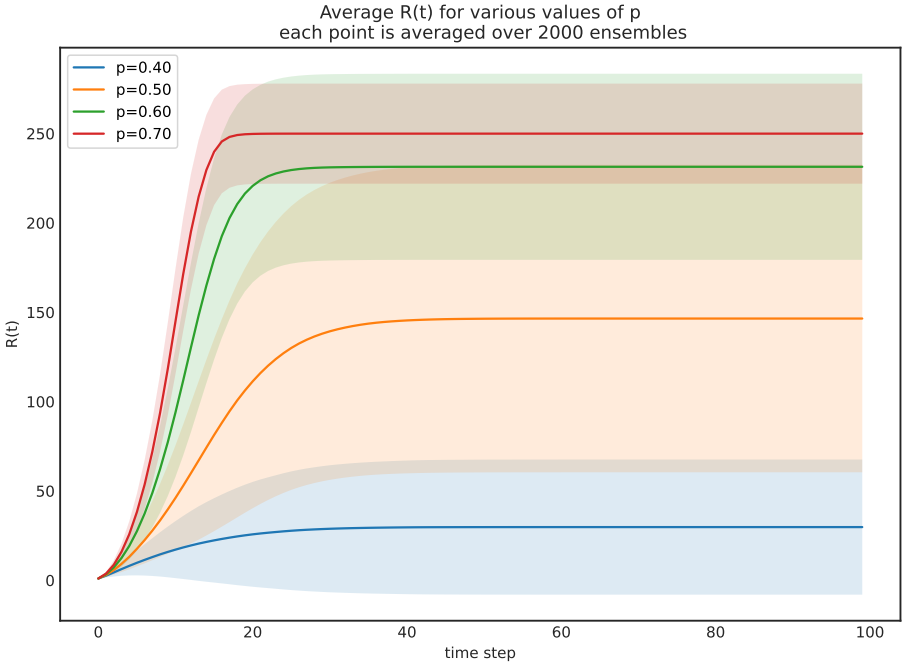


Figure 6: Average values of R(t) for different values of p for a 16 $\times$ 16 lattice. Each point is averaged over 2000 ensembles

2.4 Bonus

2.4.1 Phase transition

To see phase transition, we loop through different values of p from 0 to 1. For each p we run our code 2000 times and save the final value of R. In fig 7 you see the phase transition diagram for this system. To plot this diagram we choose 30 values for p between 0 and 1 with equal distance and for each p, we averaged over 2000 ensembles. Evidently, the phase transition happens around p=0.4. If we compare this diagram with the diagram we obtain in the previous assignment, we see that in that chart for R(t=0)=0 and I(t=0)=1 (which is the case here), the phase transition happens at R_0=1. Note that in previous assignment we had **rate** of getting sick and recovered which both of them have the dimension of time^{-}1. Nevertheless, here have the **probability** of getting sick and recovered which are the dimensionless quantities. The other difference between these two diagrams is that in the previous phase transition diagram, each point's value (R(∞)) was exact. But here because we averaged over different ensembles, each point has an error. This error increases as we approach the phase transition point and decrease when we pass this point.

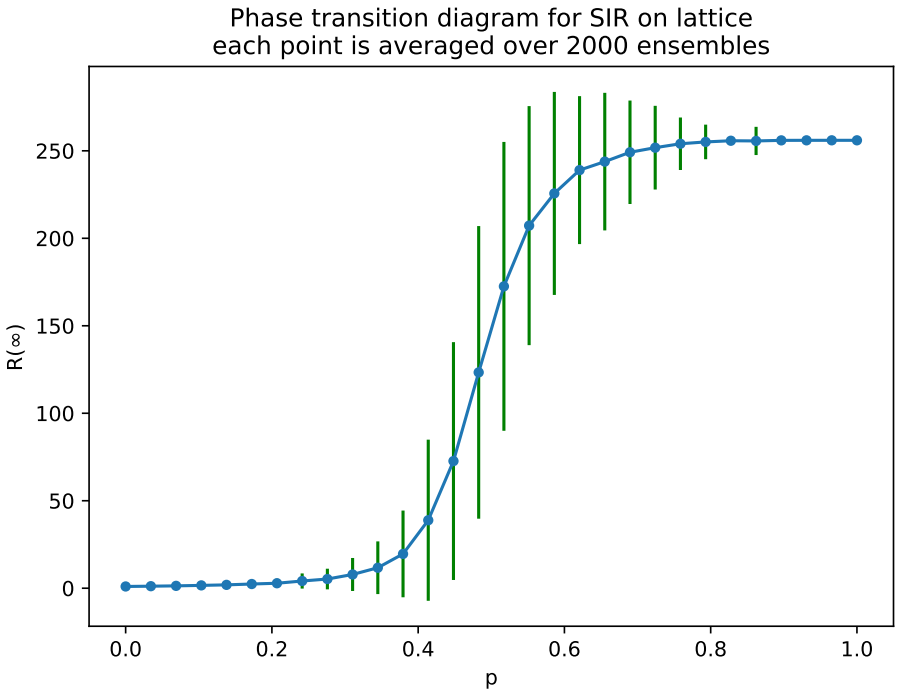


Figure 7: R(∞) versus p for a 16×16 lattice with I(0)=1 and R(0)=0

We could run our code for larger number of nodes and see the results. We do this for a lattice with 1024 nodes. in fig 8 you see the phase transition for this system. As it is shown, the resulting diagram is similar to 8.

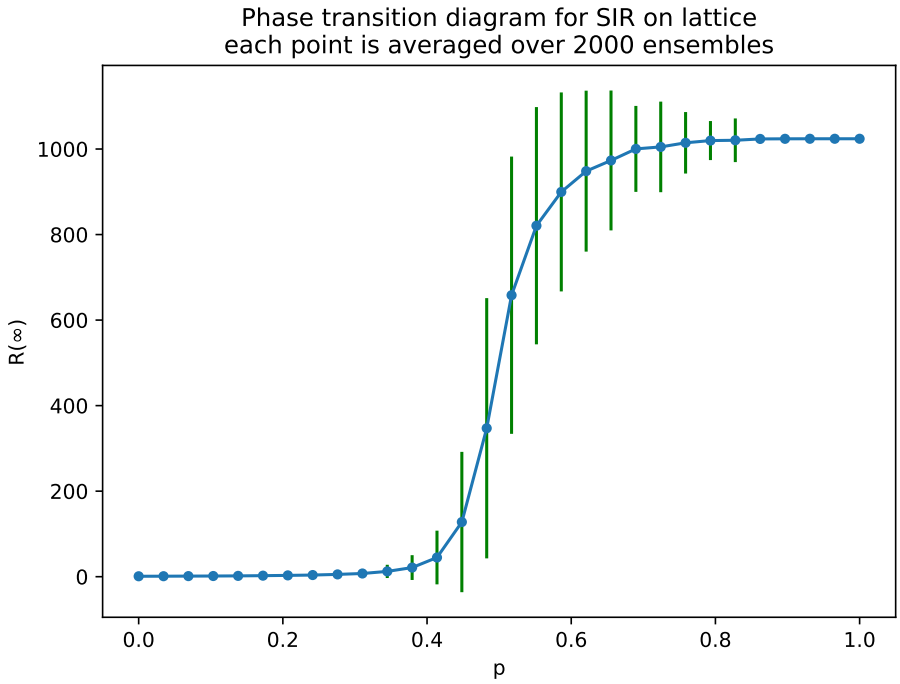
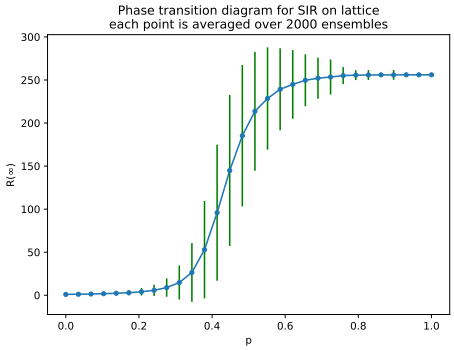
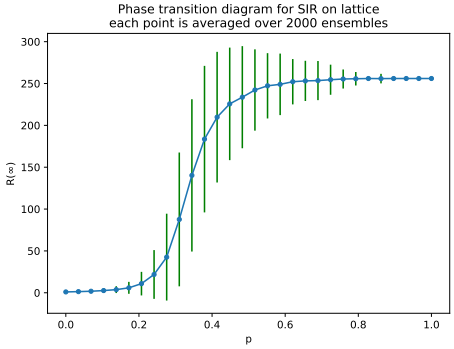


Figure 8: R(∞) versus p for a 32×32 lattice with I(0)=1 and R(0)=0

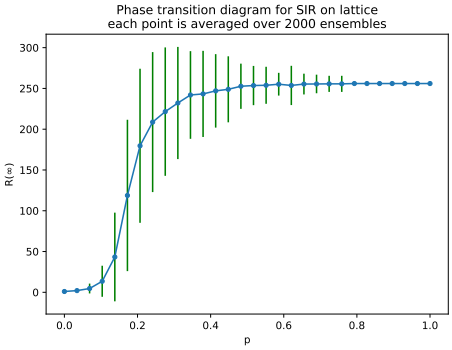
We could also change the value of r and examine the results. In figure 9 you see the phase transition for different values of r for a 16×16 square lattice. It is evident by decreasing the value of r the phase transition happens at lower values of p.



(a) r=0.8



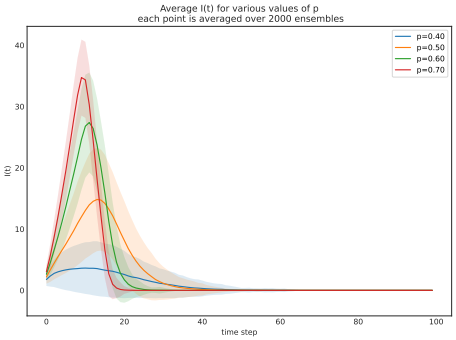
(b) r=0.5



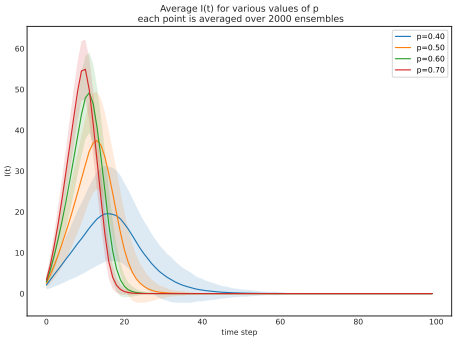
(c) r=0.2

Figure 9: Phase transition diagram for different values of r for 16×16 square lattice

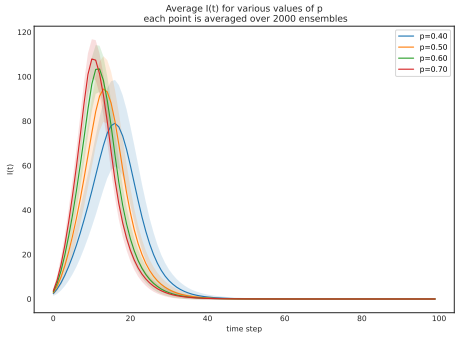
In figure 10 you see the epidemic curve for these different values of r. As you can see, the lower probability of getting recovered the higher the maximum of the epidemic curve.



(a) r=0.8



(b) r=0.5



(c) r=0.2

Figure 10: Epidemic curve for different values of r for a 16×16 lattice

3 Bombay plague Data

In this section we are going to fit a function in the form of a $\text{sech}^2(\text{bt+c})$ to the provided Bombay plague weekly death rate Data. This function is an approximate solution of the SIR system of equations for $\frac{dR}{dt}$ (page 23 [1]). To fit this function to our data, we use Scipy[2] library's `curve_fit` function. You see the result in fig 11. Thus the parameters of our function are:

$$a = 8.74 \times 10^2 \pm 2 \times 10^1$$
$$b = 1.93 \times 10^{-1} \pm 6 \times 10^{-3}$$
$$c = 2.98 \pm 9 \times 10^{-2}$$

As you can see, this data is for **weekly** time-frame. However we can use it for daily death rate too due to the fact that this is the **rate** of deaths and not the number of deaths itself. As a result we can use this data for daily time frame as well.

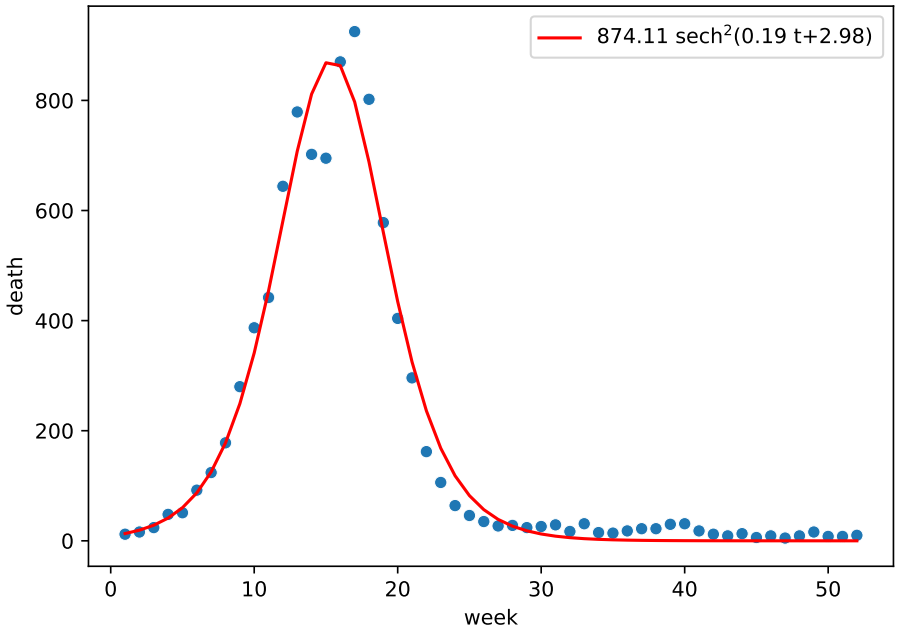


Figure 11: The filled circles represent weekly deaths from plague in Bombay from December 17, 1905 to July 21, 1906. The solid line is the fitted approximate solution given by $\frac{dR}{dt}$ = 874.11 sech2(0.19 t+2.98).

4 Review recent articles about COVID-19

In this section we review two articles in which used pandemic models to model the COVID-19 pandemic.

4.1 Pandemic in Brazil

Scabini et al. [3] have proposed an improved SIR on the network to study Brazilian epidemic. Their network consists of several layers: Home, Work, Transport, School, Religious activities, and Random. There are same nodes in every layer but the connections between nodes are different. Unlike traditional SIR model composed of 3 groups, their model categorizes individuals into seven different groups: Susceptible, Infected - asymptomatic, Infected - Mild, Infected - Severe, Infected - Critical, Recovered and Dead. For building the structure of the layers they have used real data. For instance to make the Home layer they have used demographic data of the country.

- **pros**:
    1. realistic model by considering the different layers of the interaction between people and use real-world data to build these layers.
    2. using improved version of SIR by considering different infection stages like mild infected and severe infected.
- **cons**:
    1. High computational cost due to the complexity of the simulating multi-layer networks
    2. Need lots of sociological information which can be hard to obtain

4.2 SIR model assumption for the spread of COVID-19 in different communities

Cooper et al. [4] have developed a SIR model to study the spread of COVID-19 in different communities. Their model does not keep the population constant, so unlike the traditional SIR model, the number of susceptible individuals does not decrease monotonically. In many communities, a spike in the number of infected individuals has been seen, representing the second wave of infection. To account for such phenomena, they define S_{surge}, representing the number of susceptible individuals at the spike. Moreover, at any time the value of S can change to S_{surge}. Their analysis considers data from January to Jun 2020 for China, South Korea, India, Australia, USA, Italy and Texas.

- **pros:**
    1. The small number of parameters of this model makes it easy to use on real data.

2. Consider the surge in the number of infected individuals which makes their model more realistic.
3. low computational cost.

- **cons:**

1. not considering the demographic structure of the population.
2. constant rate of getting sick and recovered

References

1. Keeling, M. J. & Rohani, P. *Modeling Infectious Diseases in Humans and Animals* https://doi.org/10.2307/j.ctvcm4gk0 (Princeton University Press, Sept. 2011).
2. Hagberg, A. A., Schult, D. A. & Swart, P. J. *Exploring Network Structure, Dynamics, and Function using NetworkX* in *Proceedings of the 7th Python in Science Conference* (eds Varoquaux, G., Vaught, T. & Millman, J.) (Pasadena, CA USA, 2008), 11–15.
3. Scabini, L. F. *et al.* Social interaction layers in complex networks for the dynamical epidemic modeling of COVID-19 in Brazil. *Physica A: Statistical Mechanics and its Applications* **564,** 125498. ISSN: 0378-4371. http://dx.doi.org/10.1016/j.physa.2020.125498 (Feb. 2021).
4. Cooper, I., Mondal, A. & Antonopoulos, C. G. A SIR model assumption for the spread of COVID-19 in different communities. *Chaos, Solitons Fractals* **139,** 110057. ISSN: 0960-0779. https://www.sciencedirect.com/science/article/pii/S0960077920304549 (2020).