Virus spread on networks

Magno Alessandro

S4478234@studenti.unige.it

1: Introduction

This project takes inspiration from the pandemic situation we have lived in for two years. The idea is to analyse how a disease spreads inside a network using the SIR model. The tool that has been used is Netlogo, which we saw during the course.

1.1: Netlogo

NetLogo is an agent-oriented platform used for simulating different scenarios. It combines the characteristic of being a potent programming tool with a simple and user-friendly interface, thanks to practical graphical support and several modes of interaction for programming. The world of NetLogo uses an agent-oriented approach, which consists of agents capable of following instructions. Each agent is an entity which carries out its actions independently of the others but simultaneously with them. There are three different types of agents:

- **Turtles**: they are the agents which can perform actions; they move into the world of Netlogo, obeying assigned rules
- **Patches**: they are the ground on which the turtles can move; the world consists of a two-dimensional space divided according to a grid, each element of which is a patch
- **Observer**: the observer has no specific location but is as if it were placed outside the world of turtles and patches and looks at the scenario

2: SIR epidemic model description

The SIR model [1] supposes that an individual can be in one of three states/compartments:

- **Susceptible (S)**: healthy individuals who have not yet contacted the pathogen
- **Infectious (I)**: contagious individuals who have contacted the pathogen and hence can infect others
- **Recovered (R)**: individuals who have been infected before but have recovered from the disease (or died) hence are not infectious

The contact network structure and two additional quantities, the probability of contagion (p) and the duration of the infection (T_i), control the epidemic's progress. Initially, some nodes are in the state I and all others are in state S. Each node v that enters state I remain infectious for a fixed number of steps T_i.

During each T_i step, each infected node has a probability p of passing the disease to each of its susceptible neighbours. After T_i steps, the node is no longer infectious or susceptible, and it is described as recovered(R), which can no longer either catch or transmit the disease.

2.1: Implementation of the SIR model

The implemented scenario starts with importing a network (chosen by the user), which can be a simple one or a complex one. Initially, all the nodes are in state S; their label is “S”, and their colour is yellow. Then the user can select the number of nodes to start the diffusion of the virus and can update the view to see them. These nodes move to the state I; their label is “I”, and their colour is red. The user has to fix the other model parameters:

- the disease transmission probability (p)
- the recovery probability (q)
- the duration of the infection (T_i). For simplicity, we assume that each node spends a minimum of T_i time ticks in the state I, and after that, it can move to state R with the given recovery probability (q)

When the user begins the simulation, the following algorithm is executed:

- while all nodes are not in compartment I (e.g., Infected, this means that all nodes are in state S or R), do:
  - From S to I: for each infected node, look at their neighbours in state S and spread the contagion with probability p. For each neighbour of an infected node, we sample a random number, and if the result is less than p, a contagion occurs, and the neighbour moves to compartment I. The T_i of the infected node is decreased by one. We change the neighbour node label to “I”, its colour to red and T_i is updated with the value selected by the user.
  - From I to R: for each infected node, if a minimum of T_i time ticks have elapsed (this means that T_i is equal to 0), move the node to the compartment R with probability q. We sample a random number and if the result is less than q, we move the node to the compartment R. We change the node label to “R” and its colour to green.

while there are nodes in compartment I

2.1.1: Networks data

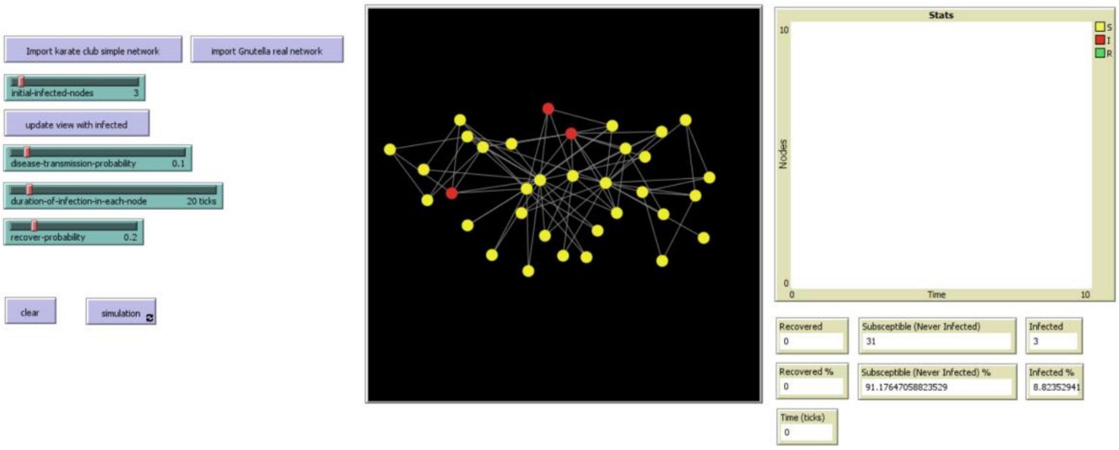
The simple network is called Zachary's karate club [2]. These are data collected from the members of a university karate club by Wayne Zachary. This social network captures 34 karate club members (nodes), documenting 78-links between pairs of members who interacted outside the club.

The complex network used to perform the task comes from a peer-to-peer file-sharing network named Gnutella, collected in august 2002 [3]. The network is directed but has been treated as undirected because the virus can be transmitted in both directions. The graph is generated from a sequence of 9 snapshots of an unspecified peer-to-peer network, which uses the Gnutella protocol. The nodes represent hosts in the Gnutella network topology, and edges represent connections between the Gnutella hosts to send and receive data. I can notice that the network is well connected; in fact, it is almost a unique giant component (6301 total nodes, 6229 nodes of the giant component, only 2 nodes are outside).

3: Instructions

The software implemented is simple to use. The figure below shows the primary User Interface that users can use to interact with the software.

3.1: User interface Netlogo


Import karate club simple network
import Gnutella real network
update view with infected
clear
simulation
Stats
S
I
R
Nodes
10
0
Time
0
10

initial-infected-nodes 3

disease-transmission-probability 0.1

duration-of-infection-in-each-node 20 ticks

recover-probability 0.2

Recovered 0

Subsceptible (Never Infected) 31

Infected 3

Recovered % 0

Subsceptible (Never Infected) % 91.17647058823529

Infected % 8.82352941

Time (ticks) 0

Netlogo represents the world as a square where turtles can move over the patches. This project has used just one kind of turtle to represent the nodes, which are static (they do not move) and can perform three different actions: move to state S, move to state I, and move to state R.

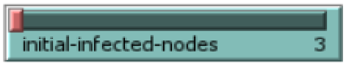
3.2: Input parameters set by UI

3.2.1:

It is a button that imports the simple network karate club from a file. In particular, we import first the nodes from a file organized as a column of ids which are integers, then the links from another file organized as two columns of integers representing the ids of nodes between existing an edge.

3.2.2:

It is a button that imports the complex network Gnutella from a file organized as the command above. (increase the time of ticks to faster; otherwise, it will take too much to upload it. Also, it is not well visible due to the size.)

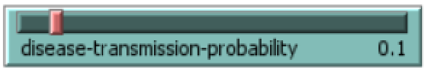
3.2.3: 

initial-infected-nodes 3

It is a slider button that sets the number of initially infected nodes selected. It is possible to set it after we have imported a network.

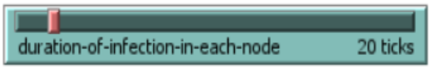
3.2.4:

It is a button that updates the view of the network, showing the infected nodes with the colour red. These nodes are selected randomly and move to the state Infected.

3.2.5: 

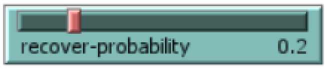
disease-transmission-probability 0.1

It is a slider button that sets the disease transmission probability (number from 0 to 1), from which the disease is spread from one node infected to a neighbour that is not infected.

3.2.6: 

duration-of-infection-in-each-node 20 ticks

It is a slider button that sets the duration of the infection in each node (measured in tick).

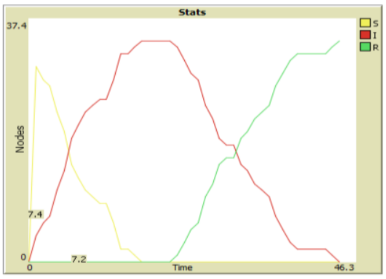
3.2.7: 

recover-probability 0.2

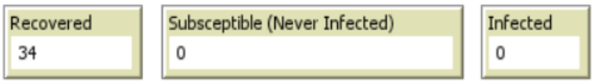
It is a slider button that sets the recovery probability (number from 0 to 1), from which an infected node can become recovered after the duration of the infection has been spent.

3.3: Analytics

A set of different items in the UI are included to show some analysis.

3.3.1: 

This graphic represents the number of susceptible (S), infected (I), and recovered (R) nodes in real-time.

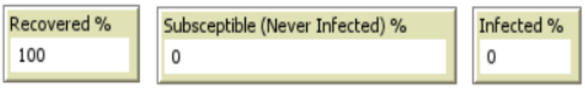
3.3.2: 

Recovered
34

Subsceptible (Never Infected)
0

Infected
0

These three windows represent the current number of nodes susceptible, recovered, and infected in real-time.

3.3.3: 

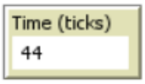
Recovered %
100

Subsceptible (Never Infected) %
0

Infected %
0

These three windows represent the percentage of nodes susceptible, recovered, and infected in real-time.

3.3.4:



Time (ticks)
44

This window shows the time spent from the beginning of the simulation (measured in ticks of Netlogo).

4: Results

Since the epidemic in the SIR model is controlled by the disease transmission probability and the length of the infection, we can try different values to see how the diffusion of the pathogen change and which final configurations we obtain. We fix the following parameters:

- recovery-probability ½
- number of initial infected nodes 3 (Karate Club graph) and 80 (Gnutella graph)
- disease probability transmission (p) 0.8 (high value) and 0.1 (low value)
- duration of the infection for each node (T_i) 50 (high value) and 10 (low value)

4.1: Karate Club

p	**T_i**	**state S**	**state R**	**Time (ticks)**
high	high	0	34	56
high	low	0	34	17
low	high	0	34	66
low	low	4	30	25

4.2: Gnutella

p	**T_i**	**state S**	**state R**	**Time (ticks)**
high	high	2	6299	66
high	low	2	6299	57
low	high	14	6287	114
low	low	783	5518	39

We can see from the previous tables the analysis of different scenarios. For each scenario (determined by the fixed parameters), we found the number of nodes still in state S, so they never got infected, the number of nodes recovered in R and the total time in ticks of the execution. In the Gnutella, two nodes are expected to remain susceptible because they are not connected with the rest of the graph.

5: Conclusion

This project aims to map the SIR model into a network. We have three different states (S, I, R) and several parameters. The goal is to see how things evolve for different values of these parameters. Analysing the tables of results, we can say:

- if the probability of contagion and length of infection are both high, the virus can reach all the nodes of a network easily

- if the probability of contagion is high and the length of infection is low, the virus can reach all the nodes as in the previous case, but the execution takes less time than before because the nodes will spend less time in the state I
- if the probability of contagion is low and the length of infection is high, the virus acts like in the first case; it can reach almost all the nodes but requires more time
- if the probability of contagion is low and the length of infection is low, the virus cannot reach all the nodes in the network, so it dies out more quickly

Additional part

As requested, a check has been done to see if the SIR model has already mapped into a network in Netlogo. We can find a solution in the library of Netlogo [4], also reported in this paper [5]. The difference between this solution and the one presented above is that they generate every time random graphs while in our solution, we import graphs from files. Furthermore, in their SIR model, the nodes infected do not become directly recovered in state R. If a node does recover, it moves to state S, and there is some probability that it will become resistant and move to state R.

The SIR model has also been used to simulate the diffusion of Covid-19 in a network with Netlogo [6]. This paper (like many others) was published in 2020. At that time, we did not have much information about this virus. Nowadays, we can say that the SIR model is not ideal for analyzing the diffusion of the Covid-19. Because we know that when an individual recovers from it, he/her does not get a permanent immunity, so after some time, the individual can get the virus again. A model that can simulate better the spread of Covid-19 is the SIRS model, which considers temporal immunity. With respect to the SIR model, in the SIRS, we also track the length of the immunity for each individual. When it is passed, the individual becomes again susceptible, moving to state S.

References

[1] Albert-László Barabási. 2016. Network Science, chapter 10.

[2] Karate Club network, taken from python library: https://networkx.org/documentation/stable/auto_examples/graph/plot_karate_club.html

[3] Gnutella network. https://snap.stanford.edu/data/p2p-Gnutella08.html.

[4] https://ccl.northwestern.edu/netlogo/models/VirusonaNetwork

[5] https://www.researchgate.net/profile/Las-Johansen-Caluza/publication/334415548_Epidemiological_Simulation_Of_A_Nonlinear_Computer_Network_Laboratories_Using_Kermack-Mckendrick_Model/links/5d282b6da6fdcc2462d692d7/Epidemiological-Simulation-Of-A-Nonlinear-Computer-Network-Laboratories-Using-Kermack-Mckendrick-Model.pdf

[6] https://dergipark.org.tr/en/download/article-file/1147407