Group 38 - Dragos Pop & Jimmy Gijsel

Finding the Optimal Vaccination Strategy for Epidemics through Simulation
- Final Assignment: Modelling Epidemics –

Introduction

Summary of the report

This report exposes each action taken in order to model an epidemic, with the final goal of providing the ideal approach for the vaccination strategy. In the beginning, the given data is presented together with the assignment description. Next, the assumptions extracted from the documentation are transposed into code and new methods are added to build an accurate simulation algorithm. At this point, the narrative thread divides into three parts, illustrating the implementation of each treatment plan, namely: vaccinating random individuals, injecting the persons with the most connections and immunizing after a certain number of steps. Last part of the report draws the conclusions, making a comparison between the three scenarios.

Problem description and the given data

An epidemic of an infectious disease is generally spreading to a large number of persons through the connections between individuals. To address this, we received a document that contains all the links between the people of a community (1000 persons), that is all the possible ways through the plague might spread.

In addition, the assignment description provided the manner somebody's condition is transformed at every time-step in relation to the infection. First, all people are prone to contacting the virus with probability 1/3 for each ill neighbor they have. If this happens, there is 1/8 probability to recover. Moreover, a recovered individual becomes susceptible again with probability 1/20. Besides these, there is the vaccinated state which offers total immunity against the disease. Hence, it breaks the ties with the neighbors and removes the injected from the network.

Setting up the environment

Transposing the assumptions into code

In this part of the report is discussed the implementation of the previously mentioned assumptions. As stated in the assignment description, infected people spread the disease with probability 1/3 to each of their neighbors. In other words, every susceptible individual has a chance of 0.333 of contacting the virus for each ill friend he/she has. Subsequently, it was designed a method able to count the number of infected neighbors a vulnerable person has and another one called *getsInfected()*, which generates a random number for each infected neighbor and compares it with the probability of getting infected. If at one of the iterations the generated number is lower than 1/3, the respective individual becomes infected.

Then, it followed the construction of the *recovers()* and *turnSusceptible()* functions. Both were composed in a similar way, as it follows: at the beginning, a number was randomly generated and later compared with the possibility of recovering, respective, becoming susceptible again. In case the second number is higher than the first one, the *recovers()* function cures the disease of an infected individual, while the *turnSusceptible()* changes the state of a person as the name of the method suggests.

Taking into account that the methods presented above cover the status of all the persons present in the network, the simulation of a single time-step run has become possible. By fitting each citizen according to his/her status and calling the respective function to the group in which the person was assigned led to the simulation of a single round. Next, the simple addition of a loop made the method adequate to simulate the spread of the disease over a predefined period of time.

Understanding how the disease spreads

Initially, it is assumed that the infection occurs at a single person chosen randomly. This means that all the persons are equally likely to contact the virus, even the citizens who are not connected with the network. The only limitation is that the already vaccinated people cannot get infected anymore and it applies to the first two injection strategies, but not to the delayed one when initializing the disease happens ten time-steps before the vaccination. Accordingly, a random number between 0 and 999 was generated to denote the index of the person who will get the infection in the first place, keeping into account that he/she was not vaccinated previously.

The next step approached the optimal value for the number of time-steps such that the results of the simulations are not altered. For discovering this number, it was essential to program a function that would graphically represent the evolution of the infected population during the specified amount of time slots. With the help of the already programmed getFractionInfected() and simulateOneRun(), the percentage of ill people was stored in an array and then plotted. As can be seen in Fig. 1, for t equals 100, the infected fraction goes into a periodic cycle at the half of the graph. This means that regardless of the vaccination strategy used, the average of the disease proportion would be negatively influenced by the cyclic sequence thus, the impact of the applied tactics diminished. Therefore, the value we decided to use for the following simulations is 50, as it does not show any sign of periodicity (See Fig. 1).

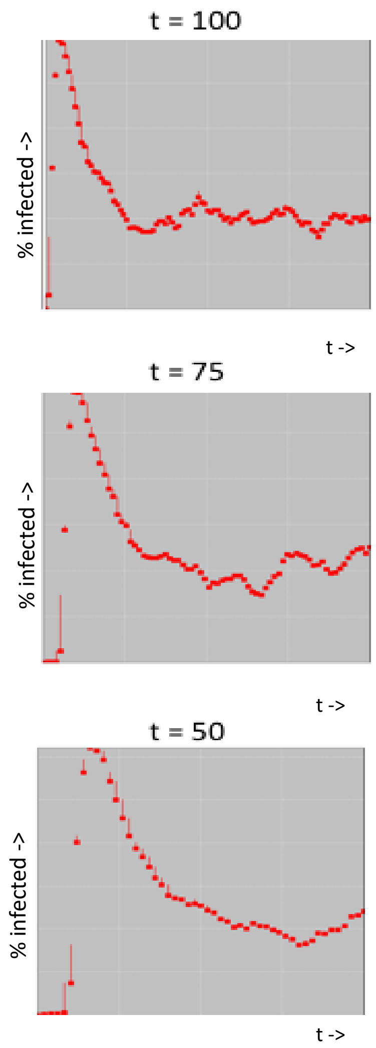

Fig. 1 – The spread of the disease for different time-steps

Vaccinating randomly choosen persons

Randomly picking the people to be vaccinated

According to the name of the plan, the vaccinated people in this strategy are selected uniformly at random. However, we assumed that, from an organizational and financial point of view, it is easy to keep track of the people previously treated such that the authorities will not inject the same person twice, even if this rarely happens when the fraction of humans to be vaccinated is low. Consequently, the desired proportion of individuals to be injected was transformed in an integer by multiplying it with the size of the population. Then, each time someone has to be treated, a random number is generated for the index of the chosen person. If he/she satisfies the condition of not being immunized already, that person is vaccinated and removed from the network.

Impact when the fraction of vaccinated people is 5%

To measure the influence on the spread of the disease when people are vaccinated arbitrarily required initializing a Tally, such that all information is stored and used in the final result. Next, as a single simulation is very unpredictable, the solution found to deal with randomness was running 5000 scenarios, each of 50 time-steps, and adding the proportion of infected people at the end of each interval to the Tally. Accordingly, a single simulation timeline debuts with loading the network, calling the previously mentioned vaccinateRandomFraction() method and then simulating a single run of every time-step given. Therefore, the results showed that when 0.05 of the population is randomly injected against the disease, the virus spreads to at most 52% of the total community, with an average bellow 0,001 (See Fig. 2).

```
REPORT on Tally stat. collector ==> null
    num. obs.      min          max         average      variance     standard dev.
    255000         0.000        0.521       7.4E-4       8.9E-5        9.4E-3
```

Fig. 2 – Result of 5000 simulations when 0.05 of the population is randomly vaccinated

Maximal spread of the infection

The next part approached finding a value for the fraction of persons to be vaccinated such that the disease does not spread at more than 10% of the population. As the algorithm was prepared already to support user input when it comes to the fraction, the answer to the question followed after trials and errors (see Appendix 1). Apparently, for 50% of the people being arbitrarily injected before the disease spreads, four out of five simulations show a favorable outcome. However, depending on how damaging the disease and how large the cost of the treatment is, we recommend further analysis of the problem in order to avoid endangering the lives of many citizens.

Vaccinating the most popular persons in the community

Selecting the most popular persons

Considering that an epidemic spreads through the connection between the people, it makes relevant immunizing the most socially active individuals in the network so that the virus has fewer channels to be transmitted. In other words, the popular persons are more likely to spread the disease, hence, we decided to implement an algorithm which records all the citizens based on their amount of neighbors and vaccinate the first fraction of this list.

Impact when the fraction of vaccinated people is 5%

Similar to the first strategy, another Tally was created, before the vaccinateNodesAlternative() function was called. Different from the case when the people are injected randomly is that the vaccinate method is called only once, outside the loop, since the most notorious persons are always the same in the given network. Then, for each simulation, the disease was initialized and the infected proportion collected after each round was reproduced. Therefore, as the image below suggests, the maximum spread of the virus when 50 persons are treated against the disease (approximately 0,157) decrease drastically in parallel to the first strategy. However, when it comes to the average, the same picture shows that the proportion of infected people is around 0,066.

```
REPORT on Tally stat. collector ==> null
    num. obs.       min        max       average     variance    standard dev.
     51000         0.000      0.157       0.066       3.7E-4        0.019
```

Fig. 3 – Result of 1000 simulations when the first 0.05 most popular populations within the network is vaccinated

Maximal spread of the infection

As expected, this strategy is far more efficient compared to the other one when it comes to the fraction of people that have to be treated such that the maximal spread of the disease is lower than 10%. One more time, using the trial and error technique (See Appendix 1) on the developed algorithm concluded that by only injecting the first 70 most notorious individuals in the network limits the germ to spread to at most 8% of the total population.

```
REPORT on Tally stat. collector ==> null
    num. obs.       min        max       average     variance    standard dev.
     51000         0.000      0.080       0.011       1.7E-4        0.013

Maximum is 0.0800
```

Fig. 4 – Result of 1000 simulations when the first 0.07 most popular populations within the network is vaccinated

Delayed vaccinating

Diminishing the consequences

As is to be anticipated, by letting the virus spread without putting any efforts against it leads the disease to become further prevalent and more difficult to isolate. In addition, the vaccine only prevents the virus from spreading to a susceptible or recovered individual, but, unfortunately, cannot heal an infected person.

In these conditions, the strategy we considered as being the most optimal when it comes to minimizing the greater virus growth is vaccinating the neighbors of the contaminated people with the most healthy connections. As it is illustrated in Fig. 5, by treating a socially active person with at least one infected neighbor would stop the illness to be transmitted through his/her links to the others susceptible acquaintances.

In order to transpose this into code, we first checked all the healthy neighbors of infected persons. Then, the number of their healthy connections was stored and used to rank them. Lastly, the first fraction of people was vaccinated and, thus, erased from the network.

In conclusion, to get the final results required adding the solution to each simulation to the Tally. Subsequently, the disease was initialized, let to spread for ten time-steps until applying the previously described vaccination method and then simulated until it reached the predefined number of steps. Using this strategy, the outcome depicted in the figure below was obtained.

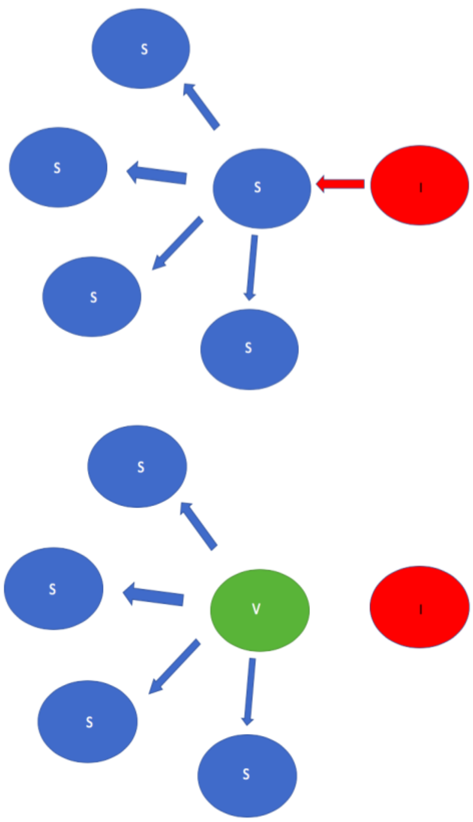

Fig. 5 – Delayed vaccination strategy

```
REPORT on Tally stat. collector ==> null
    num. obs.       min        max       average     variance    standard dev.
       51          2.0E-3     0.579       0.228       0.022         0.148
```

Fig. 6 – Solution of the delayed vaccination tactic

Conclusions

Therefore, based on the above-presented figures, it can be concluded that, in the first place, timing places a central role when it comes to isolating the spread of the disease. This fact is strongly supported by the discrepancy between the results of the latter two strategies. Even if both are choosing the persons to be vaccinated according to valid theories, it can be seen that the delayed vaccination leads to an infected population almost eight times bigger than the other tactic for the two, average and maximum spread.

Regarding the first and second approach to vaccination, namely random injection and the treatment of the most notorious persons in the community, we emphasize the benefits of the latter. Even if it can be more costly from a financial point of view, it is needless to say, after the presented results, that leads to enhanced protection.

Appendix 1

<u>Question 3</u>

Result of 5000 simulations when fraction is 0,30 and people are vaccinated randomly

```
Answer to Question 3:
REPORT on Tally stat. collector ==> null
    num. obs.      min         max        average      variance     standard dev.
    255000         0.000       0.330      2.6E-4       7.0E-6        2.6E-3

Maximum is 0.3300
```

Result of 5000 simulations when fraction is 0,35 and people are vaccinated randomly

```
Answer to Question 3:
REPORT on Tally stat. collector ==> null
    num. obs.      min         max        average      variance     standard dev.
    255000         0.000       0.299      2.3E-4       4.8E-6        2.2E-3

Maximum is 0.2990
```

Result of 5000 simulations when fraction is 0,40 and people are vaccinated randomly

```
Answer to Question 3:
REPORT on Tally stat. collector ==> null
    num. obs.      min         max        average      variance     standard dev.
    255000         0.000       0.289      2.1E-4       4.4E-6        2.1E-3

Maximum is 0.2890
```

Result of 5000 simulations when fraction is 0,45 and people are vaccinated randomly

```
Answer to Question 3:
REPORT on Tally stat. collector ==> null
    num. obs.      min         max        average      variance     standard dev.
    255000         0.000       0.243      1.9E-4       2.8E-6        1.7E-3

Maximum is 0.2430
```

<u>Question 4</u>

Result of 1000 simulations when the first 0.07 most popular populations within the network is vaccinated

```
REPORT on Tally stat. collector ==> null
    num. obs.      min         max        average      variance     standard dev.
     51000         0.000       0.022      1.2E-3       4.5E-6        2.1E-3

Maximum is 0.0220
```