CSN 06 - Simulation of SIS model over networks

Carolina Jorge, Laura Cebollero

17th of December, 2018

1 Introduction

In this delivery we are asked to study the SIS model infection. To do so, we are asked to create a function that simulates an SIS infection on a given network.

After having our function implemented, then we have to check what the epidemic threshold ($\frac{\beta}{\lambda}$) is.

Theoretically speaking, it should be

$$\frac{1}{\lambda_1}$$

, where λ_1 is the largest eigenvalue of the underlying contact network.

So we are going to check that among other things.

The graph models we are going to use in this study are five and always with size $n = 1000$:

1. Erdos-Renyi (ER) graph. 1.1. Probability of generating an edge between two nodes of 0.5.
2. A fully connected graph.
3. Barabasi-Albert (BA) graph. Power of preferential attachment = 1.
4. A star.
5. Watts-Strogatz (WS) Small world. 5.1. Dimension 1, neighborhood 4, rewiring probability of 0.4.

Let's see their basic metrics:

Table 1: Summary table of all 5 graphs

Graph	N	E	$\langle k \rangle$	δ	Diameter
Erdos-Renyi	1000	249250	498.5	0.50	2
Full graph	1000	499500	999.0	1.00	1
Barabasi-Albert	1000	999	2.0	0.00	19
Star	1000	999	2.0	0.00	2
Watts-Strogatz	1000	4000	8.0	0.01	6

With the parameters used, the full graph is the one that contains the largest amount of edges. This is obvious since the diameter is 1, all nodes are connected to all the other nodes. Erdos-Renyi follows with half of the edges, justifiable since we gave it the probability of 0.5 to generate an edge between two nodes (thus, the diameter will be 2). Barabasi-Albert model is the one with the smallest diameter (20), meaning it's the one that is worst connected.

2 Threshold

We have computed for each graph its lambda and, for a fixed gamma, we are computing the beta such that:

$$\frac{\beta}{\gamma} = \frac{1}{\lambda}$$

$$\beta = \frac{\gamma}{\lambda}$$

Graph	lambda	beta	gamma
Erdos-Renyi	500.548283	0.0007991237	0.4
Full graph	999.000000	0.0004004004	0.4
Barabasi-Albert	6.159719	0.0649380242	0.4
Star	31.606961	0.0126554399	0.4
Watts-Strogatz	8.827153	0.0453147262	0.4

3 Task 01: Spread of the disease

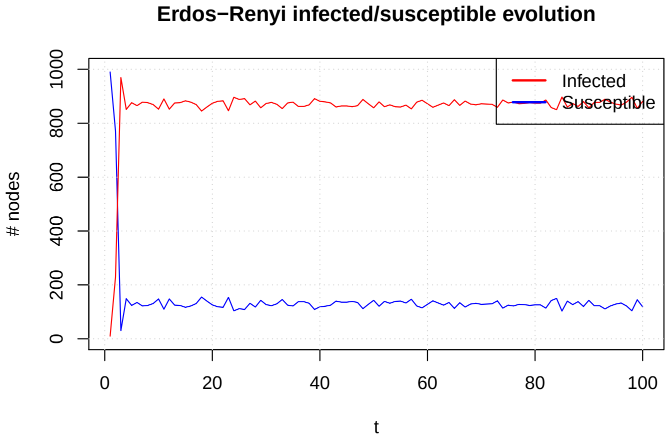
Let's first address Task 01. Having implemented the function that simulates the spread of an infection in a SIS model for a given graph, we are going to check the evolution of the infected nodes and susceptible ones.

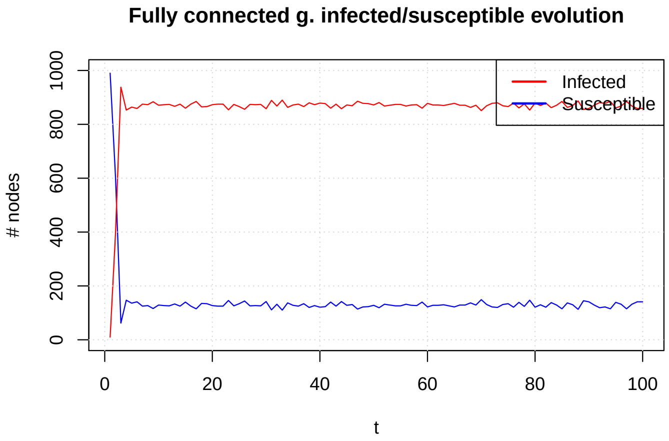
For fixed parameters such that:

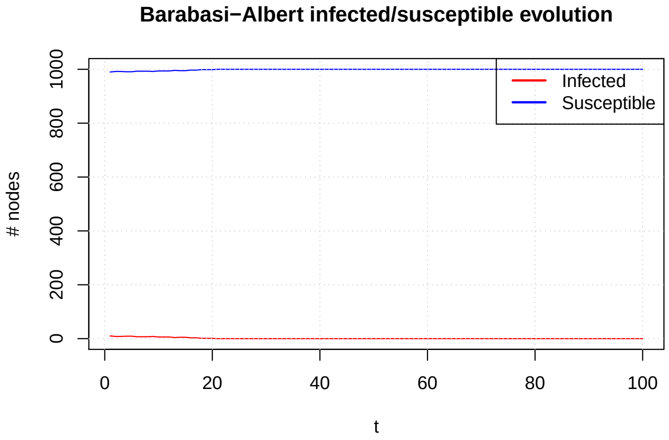
- $\beta = 0.05$
- $\gamma = 0.15$
- $p = 10$; 1% of the total nodes start on t_0 as infected.

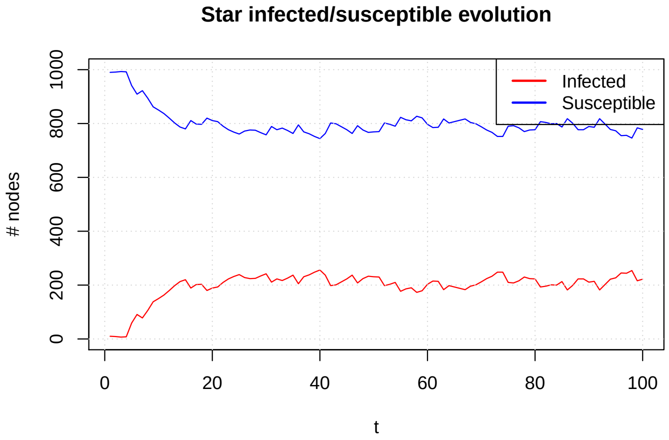
below are the evolutions of the number of infected nodes vs. the number of susceptible nodes.

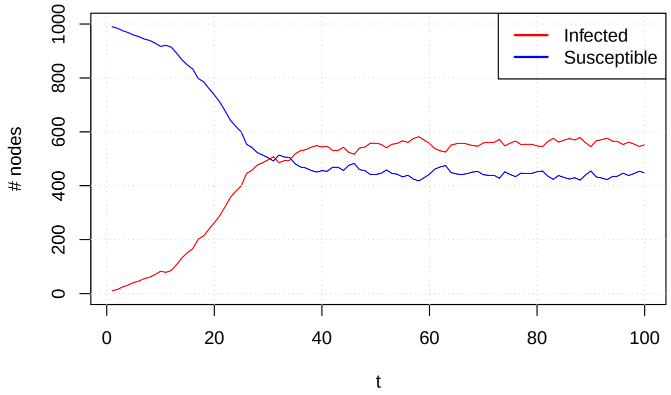
We have fixed the parameters such that $\beta << \gamma$, meaning that there should be more recoveries than infected nodes.

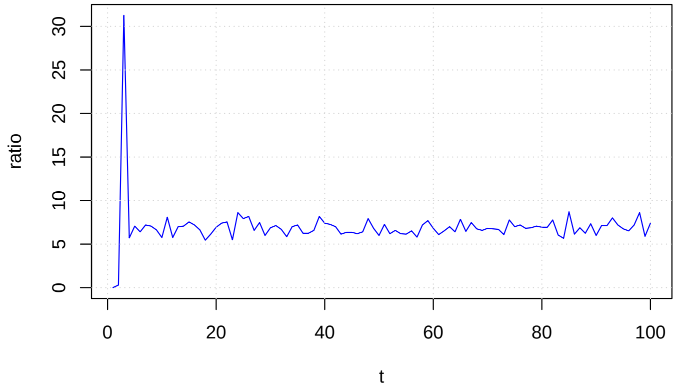


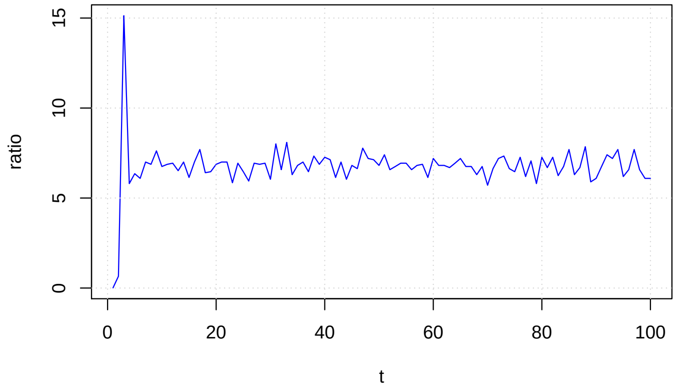

Fully connected g. infected/susceptible evolution

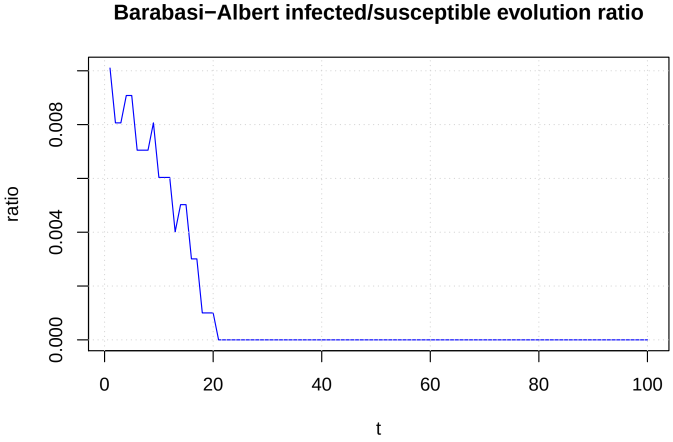

Barabasi−Albert infected/susceptible evolution

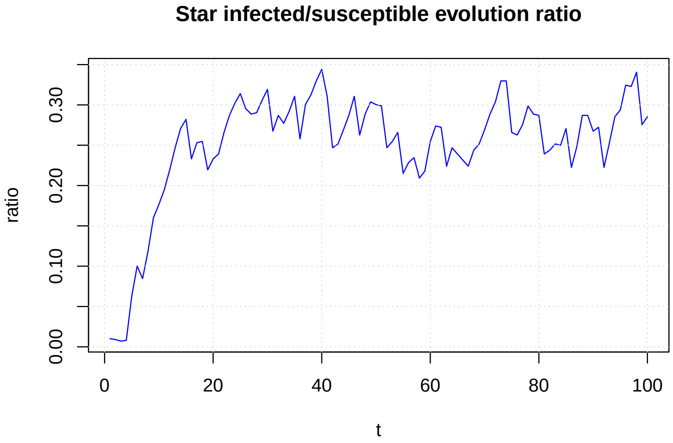

Star infected/susceptible evolution

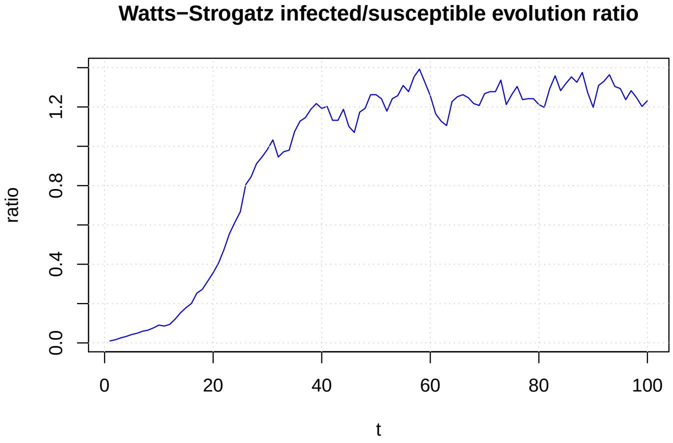

Watts−Strogatz infected/susceptible evolution


Erdos−Renyi infected/susceptible evolution ratio


Fully connected g. infected/susceptible evolution ratio







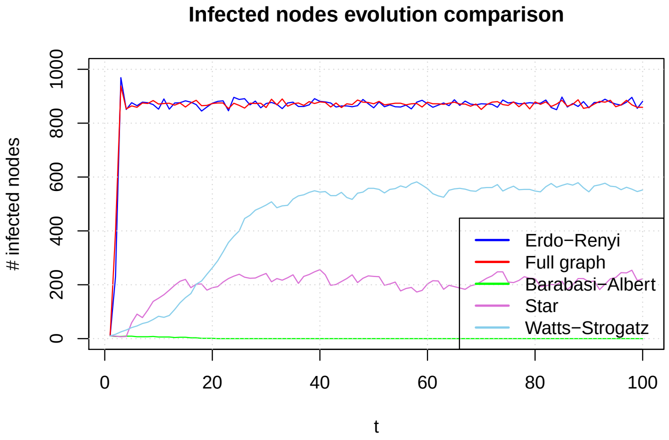
We can see how the ER and Full graph are very prone on infection, since they are pretty well-connected. The infection spreads really fast and the number of infected nodes steadies on a very large number, meaning that many nodes are going to be infected onwards.

In the BA model the contrary happens. The disease does not really spread and it actually dies a little bit before the 20 iteration.

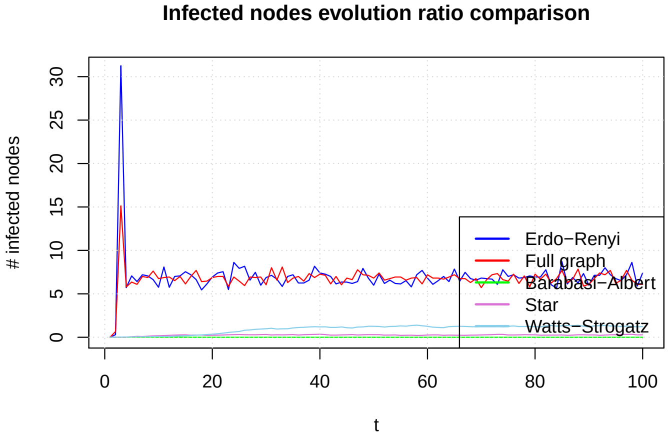
On the star model, the disease does not spread much and does it slowlier than on ER and the full graph. We can see how this network is pretty resistant and the number of susceptible (thus, not infected) is really higher than those infected.

Finally, on the last model, the WS, we can see how the disease spreads even slowlier than on the star network, but it actually reaches a higher number of infected nodes. It then steadies around $t = 50$.

For a better comparison, we can plot the evolution of infected together:



And their ratios:



We can see that the graph that, based on the first table, we verifyed was the worst connected (Barabasi-Albert) is the one that is the hardest to infect. In the star graph, we can imagine that it takes some time until it reaches an important node (one with many connections), but when it does it spreads fast and stabilizes. Watts-Strogatz takes the longest to reach its peak and estabilize. This can be due to its connection not being as goos as ER or full graph, but having a more distributed degree sequence. ER and full-graph behaved the same way. We were not expecting this result.

3.1 Results commentary

The threshold does not apply. The epidemy stabilizes always (expect for Barabasi-Albert) algorithm and the expression doesn't hold.

We think that the infection spreads too fast on the ER and full graph, for example, and this is due an error from our part on the implementation, but we have been unable to correct it so far.

4 Task 02: Use the threshold

Let's remember what the thresholds are for a fixed γ:

Graph	lambda	beta	gamma
Erdos-Renyi	500.548283	0.0007991237	0.4

Graph	lambda	beta	gamma
Full graph	999.000000	0.0004004004	0.4
Barabasi-Albert	6.159719	0.0649380242	0.4
Star	31.606961	0.0126554399	0.4
Watts-Strogatz	8.827153	0.0453147262	0.4

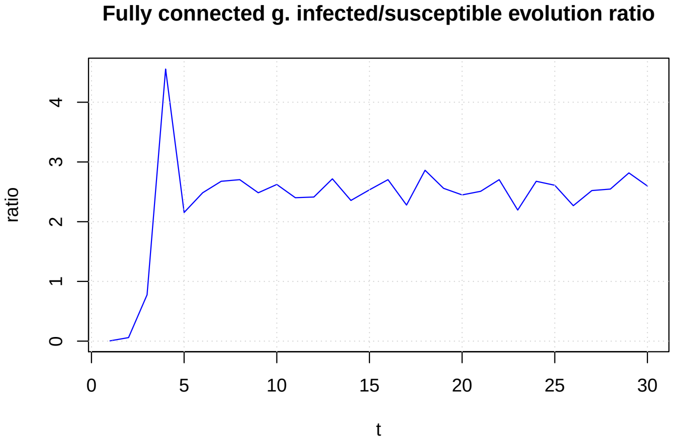
Since the Full-graph one is an homogenic system, the beta/gamma threshold should hold.

We are going to decrease the beta so that

$$\frac{\beta}{\gamma} < \frac{1}{\lambda}$$

and thus, the infection should die. We have setted the simulation to work for 30 iterations and:

- $\beta = 0.0002$
- $\gamma = 0.4$
- $p = 5$; 0.5% of the total nodes start on t_0 as infected.



We can see how the threshold does not hold. Again, we are certain that the problem resides in our implementation. After 2 weeks of debugging and asking the professor, tho, we have been unable to pinpoint the exact problem.

5 Conclusions: Problems we have encountered

The problems we have encountered with this delivery have been, ironically, many.

- We think we have the concepts very clear but our implementation fails somewhere and we have not been able to pinpoint why.
- Also, the computation of the eigenvalues table has been tricky. We have, randomly, encountered errors when using the `spectrum` function. We found that this bug has been long reported but is still open[1], so we had to make the table manually, taking the values when the ARPACK package did not fail.
- Finally, as mentioned, our results are not correct and we know about it, but we have not been able to correct them or pinpoint the problem for 2 weeks. We will probably still debug this code for more weeks since we want to focus our final project on epidemics, so we will probably face this problem again.

[1] https://stackoverflow.com/questions/35316187/arpack-error-maximum-number-of-iterations-reached