

Multi-state SIS epidemic model

Gonzalo Solera

January 2021

1 Introduction

In this document we propose a new compartmental epidemiological model that has multiple states for the infected nodes. More specifically, each node that is infected goes through different sub-states, each one with different characteristics/behaviour inspired in the Covid-19. This new model, namely multi-state SIS, is based in the SIS model in the sense that there are two compartments (susceptible and infected) from which the nodes transition back and forth.

This model is naturally applicable to networks and all the experimentation done in this work consists in applying the model over different kind of network topologies. We focus on analyzing through simulations the impact that two measurable and controllable variables have over the spread of the epidemic: preventive confinement when appreciating symptoms and time to obtain a result of a medical test.

The document is structured as follows: In section 2 a formal definition of the model is given. In section 3 we introduce the network models that we use to experiment with. The results of such experiments are shown in section 4 and a few details of the implementation are mentioned in section 5.

2 The model

2.1 Motivation

Due to the recent coronavirus pandemic (Covid-19), we realized that there are some important variables that are not properly captured by simple epidemic models. For example, it takes some time for an infected node to show symptoms, and when it does, it will preemptively self-isolate and take a medical test. Then, it takes some other time to know the result of the medical test, at which point the infected node will know for sure that it's infected and it will quarantine. During the aforementioned phases, the infected node will not infect at the same rate. The goal of the model is to take into account this type of casuistic in order to perform a sensitivity analysis of the new introduced variables.

2.2 Definition

A node can be in one of the following states:

- S (susceptible): A node is not infected.
- I_a (infected asymptomatic): A node is infected but it's asymptomatic during the following t_a time steps. During this time, it can infect each neighbor with some probability β .
- I_s (infected symptomatic): A node is infected and shows symptoms. It self-isolates, so the probability of infecting a neighbor decreases by a factor of α . It takes a medical test for which it takes t_s time to know the result. After t_s it can die with probability γ . Otherwise, it goes to the next state.
- I_q (infected quarantined): A node is infected and it knows it thanks to a medical test. It quarantines so the probability of further infections is 0.
- D (dead): A node is dead.

Visually, the state transitions are shown in Figure 1:

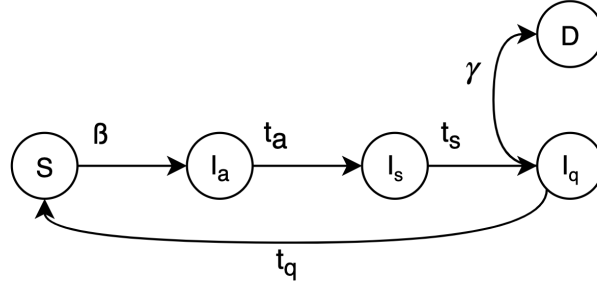


Figure 1: State transitions diagram

Consider the following situations:

- $t_i = 0$ removes state I_i .
- $t_a = \infty$ converts the model to a SI model, in which an infected node remains infected and infectious forever.
- $t_q = \infty$ converts the model to a SIR inspired model, in which an infected node that stops being infectious, remains like that.
- Other configurations yield a SIS-like model.

A possible extension of this model would consist in replacing the t_i parameters by some probability distribution of mean t_i . Another interesting extension would

be to define a transition probability $p_i = 1/t_i$. With this variation, the theoretical analysis would be much easier since we would be able to analytically obtain the differential equations fully describing the model. However, this approach is not implemented for this project since the main focus was to implement an efficient simulation over uncommon and arbitrary graphs.

2.2.1 Application of this model to networks

In this document we are focused on the application of this model over networks. Hence, we will assume an undirected weighted graph G with integer weights.

Under this assumption, on each time step every infected node u tries to infect each neighbor $v \in S$ with probability β or β/α (depending on whether $u \in I_a$ or $u \in I_s$), and it does so $w_{u,v}$ many times.

The rest of the behaviour is the same as stated previously: A node u that is infected at time t will be in I_a until $t + t_a$. Then it will remain in I_s until $t + t_a + t_s$. Then, with probability γ it will die, and with probability $1 - \gamma$ it will go into I_q . It will remain there until time $t + t_a + t_s + t_q$, when it will go back to S .

3 Networks

Since the main goal of this project was to obtain an epidemiological model inspired in the Covid-19 and it's impact in the population, we tried to obtain a network model with a topology similar to the social interactions of the Spanish population during this last year. We identify 2 important aspects:

- Home size: An important concept that has affected greatly in the spread of the Covid-19 is the home size (cohabiting nucleus). These are amount of people living together and an unit used to delimit social interactions by the local authorities.
- Locality: Another important aspect related to the spread of Covid-19 is the geographical distribution of people. The social interactions were greatly delimited by the physical distance due to legal restrictions, and hence they show a lot of locality.

With these ideas in mind, we decided to develop a network model that captures them:

To model the home sizes we first define a probability distribution of house sizes. We have collected statistics from the INE (Instituto Nacional de Estadística), where they report that in Spain 2011 there were:

- 25% of 1-person-homes

- 30% of 2-person-homes
- 21% of 3-person-homes
- 17% of 4-person-homes
- 4% of 5-person-homes
- 1% of +6-person-homes

We then use this distribution to iteratively generate cliques (houses) with the previous distribution of sizes until n nodes are created. The position of the nodes is the same among each clique (house). The position of each clique follows an uniform or a Gaussian bivariate distribution. The later distribution is very useful to model the distribution of population over a big territory, where a lot of people is concentrated around some highly-dense zones (big cities) and a few people is dispersed around big chunks of territory.

These created and positioned nodes are then passed to the **Geometric Geographical Threshold Graph** model (available in NetworkX) that creates edges following a probability distribution depending on the Euclidean distance between them, limiting the number of those edges per node to some number. This graph model is very useful to model the locality of social interactions as we intend to.

The last step consists in setting a weight of 24 to the edges inside each house (each clique created in the first step). The idea is to represent the number of hours per day that a node interacts with another with the weight of the edges. This way, treating the unit of the times t_a , t_s , and t_q as days, we would have a reasonable model of the duration of the social interactions. Hence, in this context the parameter β indicates the probability that an infected node infects a susceptible one in one hour of close physical interaction.

4 Experimentation

4.1 Setting

For the following experiments, we have used a network generated by the previous method of $n = 10000$ nodes. Furthermore, we have used $t_a = 3$ (days) since following a WHO (World Health Organization) report, this is the incubation period of Covid-19. We have used the 2-weeks quarantine guideline to set $t_q = 14$. We have set $\gamma = 0.002$ since this is the death rate of Covid-19 in Spain. The simulation time is one year: $T = 365$. The initial infected nodes are chosen at random, with a proportion of $p_0 = 0.001$.

The network is shown in Figure 2:

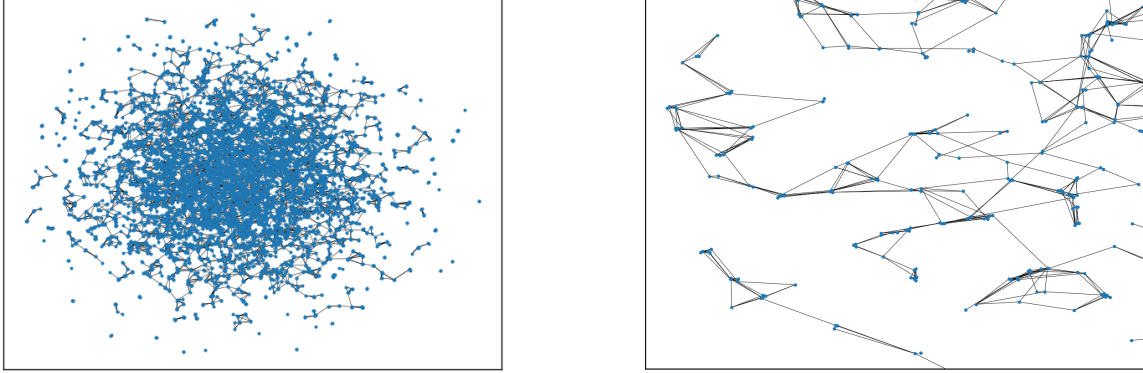


Figure 2: Realistic network of $n = 10000$ with different levels of zoom

4.2 Parameters under analysis

We aim to perform a very informal sensitivity analysis of the variables α and t_s , since these are measurable variables that local authorities can track and modify (with some limitations) in order to control the spread of the pandemic:

- Interesting values of t_s are:
 - 1 day: Fast testing.
 - 3 days: A reasonable value since this was the average time to obtain a PCR result.
 - 5 days: Slow testing.
- Interesting values of α are:
 - 1: No change in social interactions after experiencing symptoms.
 - 2: A reduction of the infection risk by 2, due to a “light self-isolation”.
 - 10: A notable reduction of infection risk by 10, due to a “heavy self-isolation”.

4.3 Results

The results of each simulation (plots and video) are stored under the folder **plots** attached to this document. The reader can find a folder for each simulation. The folder name contains the exact parameters used to run the simulation in order to ease reproducibility.

For each simulation performed, we automatically generate a video visually showing the spread of the virus over the network. It’s a very informative and interesting way of appreciating the spread of the virus, and we highly recommend the reader to watch some of the videos attached to this document under the `plots` folder.

4.3.1 t_s sensitivity analysis

As one can appreciate in Figure 3, shortening the time to test (t_s) can help a lot, making possible to stop the spread. The same effect happens when considering the SIR-like model setting $t_q = \infty$ as shown in Figure 4.

4.3.2 α sensitivity analysis

Obviously, the more preemptive self-isolation that an infected node does when realizing that it’s infected, the better the spread is going to be controlled. In Figure 5 we show that in order to stop the spread, a radical change in the amount of social interactions needs to be taken once a node suspects that it’s infected.

5 Implementation

Regarding the implementation of this project: the code is efficient, well structured and clean. The user can provide a set of epidemic model parameters and a set of network model parameters, which depend on the selected network model.

The generated networks are stored in a “cache” folder in order to speed-up subsequent simulations that use the same network model parameters.

`NetworkX` is heavily used to perform the simulation. This simulation is efficient and makes use of external data-structures to speed-up the computation. The biggest bottle-head is related to the network generation and the video showing the spread of the epidemic.

The reproducibility of the experiments is taken seriously by storing the parameters when saving the results of the simulation.

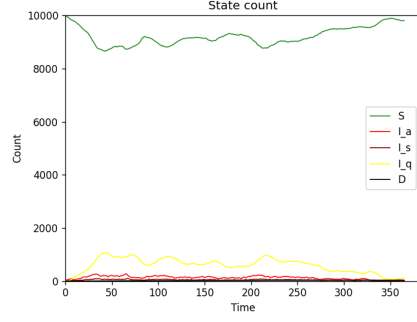
6 Conclusions

Although we haven’t developed theoretically the proposed model, we believe that the ideas behind it are interesting and applicable to an epidemic of such large scale as Covid-19.

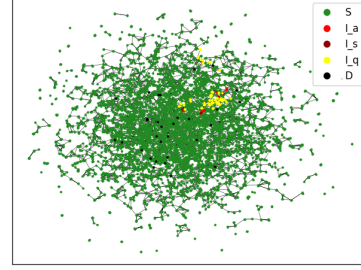
The main efforts of this work have been:

- Defining the model in a simple but powerful and expressive way.

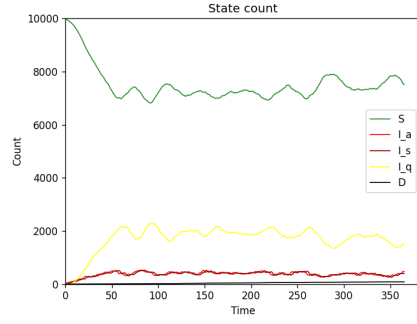
- Building a network model with the topological properties that we were looking for.
- Implementing an efficient code that allows us to run simulations on large networks and to generate videos of them.



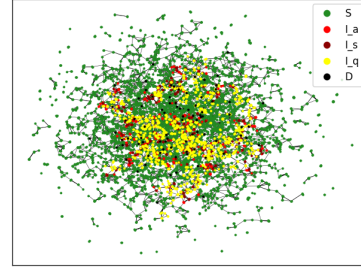
(a) State counts with $t_s = 1$



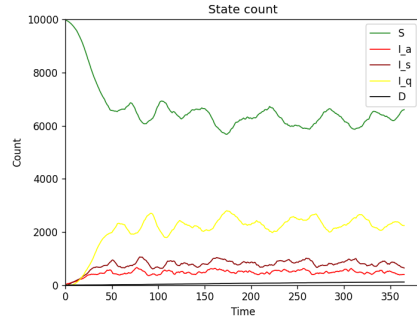
(b) Final network



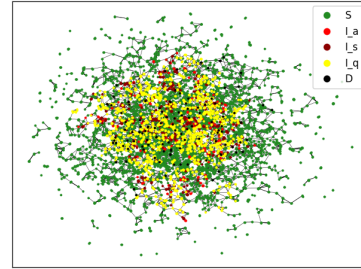
(c) State counts with $t_s = 3$



(d) Final network with $t_s = 3$

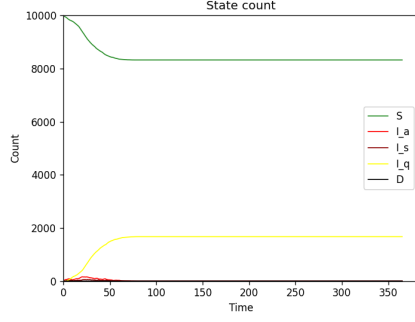


(e) State counts with $t_s = 5$

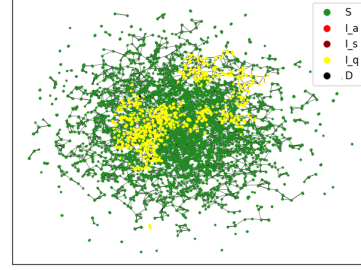


(f) Final network with $t_s = 5$

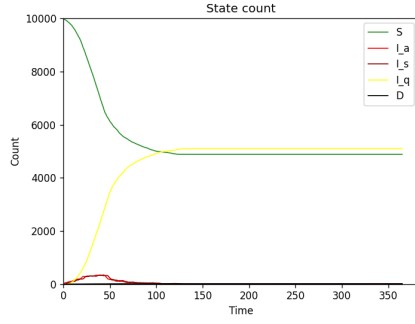
Figure 3: Effect of t_s with $t_q = 14$



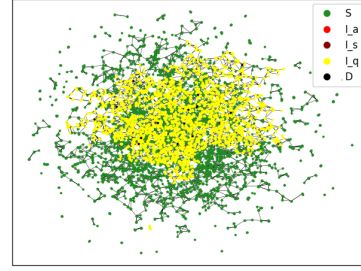
(a) State counts with $t_s = 1$



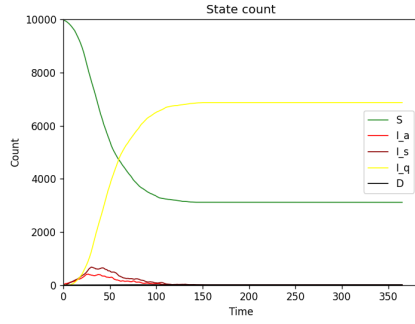
(b) Final network



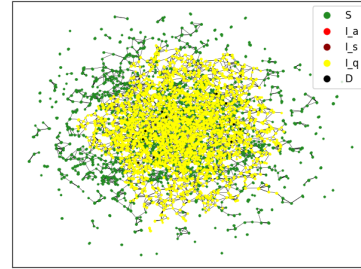
(c) State counts with $t_s = 3$



(d) Final network with $t_s = 3$

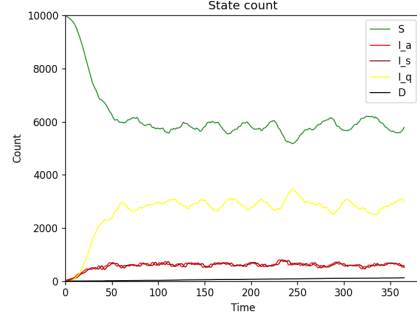


(e) State counts with $t_s = 5$

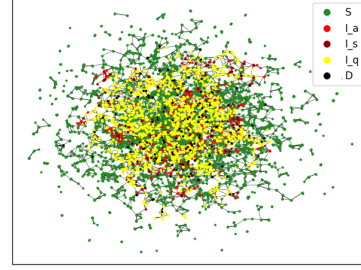


(f) Final network with $t_s = 5$

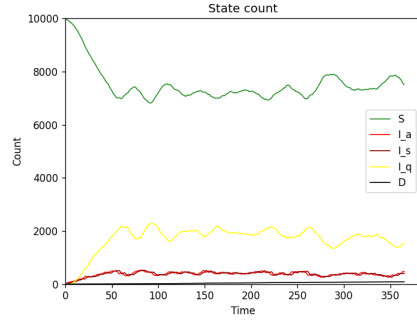
Figure 4: Effect of t_s with $t_q = \infty$



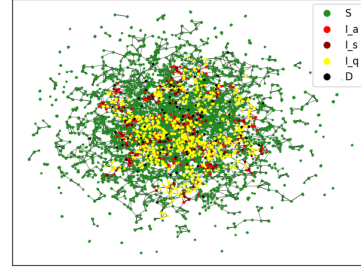
(a) State counts with $\alpha = 1$



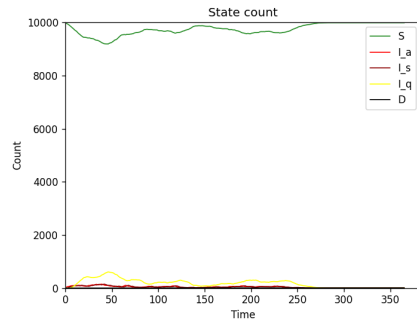
(b) Final network with $\alpha = 1$



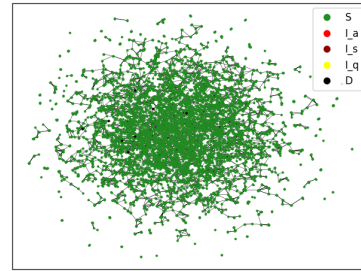
(c) State counts with $\alpha = 2$



(d) Final network with $\alpha = 2$

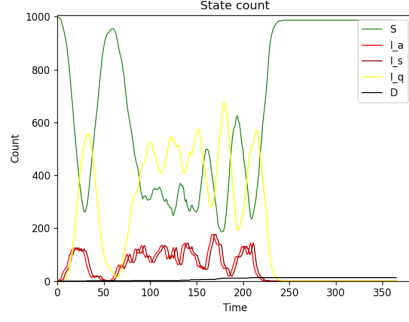


(e) State counts with $\alpha = 10$

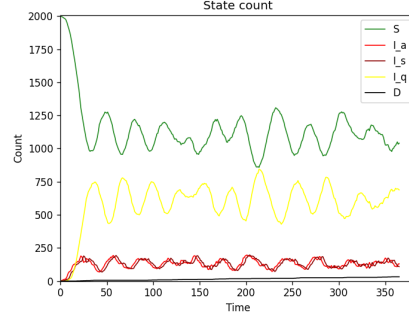


(f) Final network with $\alpha = 10$

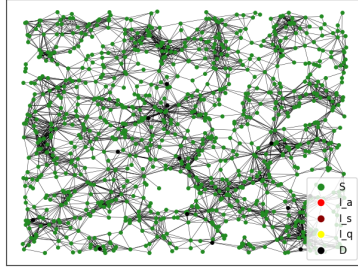
Figure 5: Effect of α



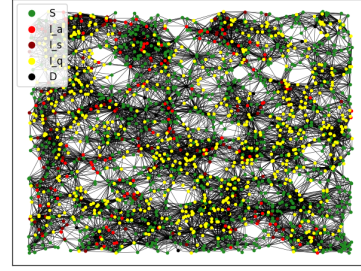
(a) State counts with $n = 1000$



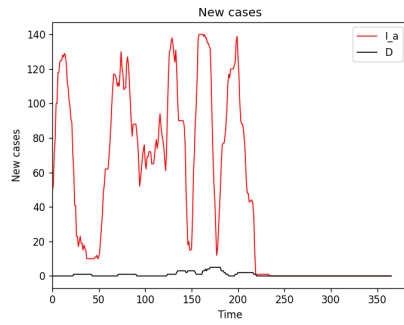
(b) State counts with $n = 2000$



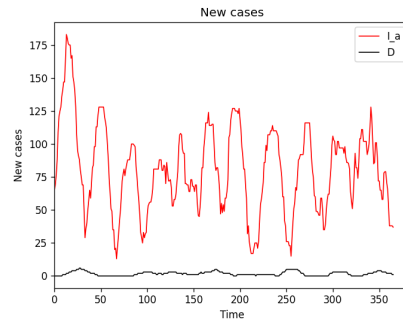
(c) Final network with $n = 1000$



(d) Final network with $n = 2000$



(e) Daily new cases with $n = 1000$



(f) Daily new cases with $n = 2000$

Figure 6: Different sizes of Geometric Threshold Graph