Exercise 2
Determinig the Spreading of Epidemics

Paolo Frazzetto, Enrico Lorenzetti

Università degli Studi di Padova
Statistical Mechanics of Complex Systems
Prof. Samir Suweis, Prof. A. Maritan

23 April, 2019

Introduction

In recent years, epidemics forecasting is becoming more and more relevant in order to eradicate, or at least diminish, the impact of plausible epidemic outbreaks. To better understand and predict these phenomena we need reliable models that unavoidably hold for some specific cases and assumptions. In this assignment, we analyse the *SIS model*: a *Susceptible* person comes into contact with an *Infected* and gets sick with an infection rate λ but, after some time, he heals with recovery rate μ and becomes susceptible again. Even though this model may look overly simplified, it can be used to study infections such as the common cold or influenza.

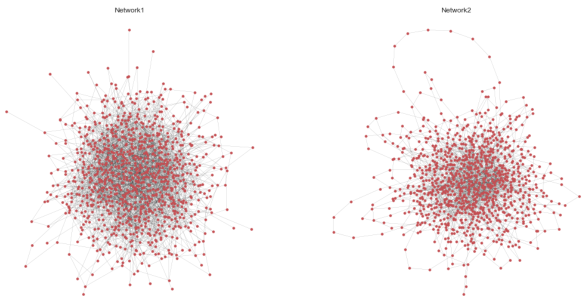

Figure 1: Pictorial representation of Network 1 (left) and Network 2 (right).

All the calculations, simulations and plots have been carried out by means of the attached `Python` notebook. Besides the most common packages, we used also `NetworkX` to import and analyse the graphs and `NDlib` to simulate the SIS dynamics. Moreover, a `C++` program has been used to implement the Gillespie algorithm to compare the two simulation approaches.
Only at the end of this work we utilise a different model, *SIR* (susceptible, infected, removed), in order to understand if a certain disease will spread among the population, given data of the its early stages.

1 Epidemic Thresholds

In the *SIS* model there are two parameters which regulate the dynamics, λ and μ. They are respectively the the proportional constant of the transition rate from $I + S \longrightarrow I + I$ and $I \longrightarrow S$. The first quantity of interest, is the *epidemic threshold* λ_c, i.e. the infection rate above which the epidemic gets viral keeping a certain μ fixed. For this purpose, two contact networks have been provided with one thousand nodes each and we set $\mu = 0.5$ throughout this work. Under different levels of approximations, the epidemic threshold can be computed analytically as described in the following subsections.

1.1 Homogeneous

Starting from the simplest but most naive case, we assume that *all* the nodes interact with each other without any spatial structure, such as in a Mean Field approximation. In this case we have that

$$\lambda_c = \frac{\mu}{<k>} \tag{1}$$

and for the two networks $\lambda^1_{c,MF} = 0.097$, $\lambda^2_{c,MF} = 0.139$.

1.2 Heterogeneous

We can add some complexity to our model if we consider that all the nodes *with the same degree* are statistically equivalent. The non-trivial solution of the critical parameter is then given by

$$\lambda_c = \frac{\mu <k>}{<k^2>} \tag{2}$$

and it results that $\lambda^1_{c,H} = 0.081$, $\lambda^2_{c,H} = 0.076$. Notice that now $\lambda^1_{c,H} < \lambda^2_{c,H}$, meaning that Network 2 has some nodes with many edges such that the second moment of the degree distribution is greater than in Network 1.

1.3 Quenched Mean Field

For the final and more realistic scenario, we take into account the network structure and connections among individuals. Using the following expression

$$\lambda_c = \frac{\mu}{\Lambda_{max}(A)} \tag{3}$$

where $\Lambda_{max}(A)$ is the largest eigenvalue of the $A_{i,j}$ adjacency matrix of the network and it can be proved that it is bounded above by the maximum degree. We have that $\lambda^1_{c,QMF} = 0.080$, $\lambda^2_{c,QMF} = 0.066$.

2 Stochastic Simulation

We would like to compare these theoretical results with the numerical values specific for our networks. To simulate the stochastic process of the epidemic spreading we followed two approaches: in the first one we relied on the library `NDlib`[1] that allows simple and flexible simulations of networks diffusion processes, and in the second one we implemented the *Gillespie* algorithm by ourselves in `C++`. Satisfyingly, they both provide similar outcomes.

2.1 NDlib Simulations

This library has a ready-to-use SIS model class that perfectly suites our needs. In addition, by looking at the source code we found that the implementation is quite simple and straightforward: at each iteration it updates the status of all the nodes comparing a uniform random float with the rates, also by taking into account the fact that a node with more infected neighbours will more likely get sick. Thanks to this synchronous update, the amount of infected populations reaches a stationary value after few tens of iterations. In our analysis first we decided to start with a 5% randomly chosen infected population I. Indeed, as it is shown in Fig.5, initial conditions do not affect so much the stationary state but only the time required to reach it. Nevertheless it's obvious that fluctuations could play an important role if we are close to $I \approx 0$, since once it reaches zero the epidemic spread must stop. So if we started with a number of infected very low, we could have a random noise and the simulation would be over without reaching the stationary state. Even so, we want to keep our simulation as close to reality as possible, therefore we chose a plausible scenario as initial condition. We avoided the problem previously remarked by repeating the simulation for the same parameters many times in order to have a sufficient statistics. We picked 50 evenly spaced values in the range $\lambda \in [0, 1]$ and launched 50 runs for each of them, storing the number of infected after 100 iterations. In this way we got the phase diagrams and, as expected, they both show a phase transition behaviour (Fig.2). Now, we focus our simulations closer to the critical point in order to find a numerical estimate of λ_c. As mentioned before, we start with a random 5% infected population but we reduce the effect of fluctuations by averaging over 100 runs for 40 λ variables in the range given by the theoretical results. Since we are looking for a λ_c that leads to a statistically significant value of infected percentage such that $I(\lambda_c) > 0$ (or in other words, the value at which all the population will not totally heal after some transient time), we have to face the issue of how to estimate the uncertainty of the mean infected population. Considering that we are close to a critical point with finite-size effects and $I \geq 0$ are lower bounded therefore not normally distributed, we cannot trust the standard deviation as a measure of error. Instead, we will count on bootstrapping to get reliable asymmetric 90% confidence intervals. The choice of this confidence interval is plausible since we are interested in a cautious and safe estimate, even if this may lead to a underestimation of the true λ_c.

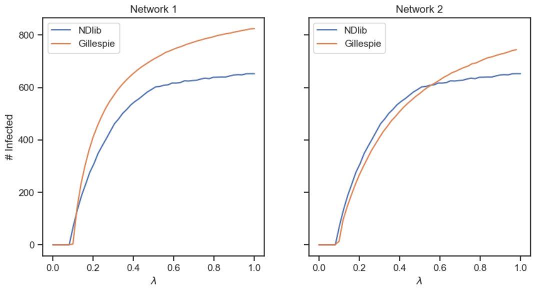

Figure 2: Phase transition of the two networks and comparison of the two simulations fashions.

Concerning the first Network (Fig.3) observe that it manifests an irregular transition characterised by large fluctuations and $I > 0$ for some isolated λ. The lowest infection rate with confidence interval that does not include zero is for $\lambda^1_{c,NDlib} = 0.093$: at and above this value we can affirm that, with at least 90% probability, if we start with 5% infected population the epidemic will not stop by itself but dangerously it will follow its course, with average fraction of infected being $I = 0.0012^{+0.0025}_{-0.0001}$. This estimate is closer to the one of the Homogeneous approximation. This might reflect the fact that the individuals in this Network are somewhat alike, with the same social structure. We also speculated that such a irregular phase transition could be caused by too few iterations, but since it is a stochastic process close to the critical point and to an absorbing state $I = 0$ that will be reached eventually, we stuck to this estimate of λ_c. So even if we are dealing with a system whose finite-size effects could lower λ_c, the fact that there might be too many initial configurations which go to zero and other features previously shown when we built the algorithm, we obtained slightly different results from what we expected. On the other hand, Network 2 has a well shaped transition that could be caused by some specific nodes that are

[1] ⟨https://ndlib.readthedocs.io⟩

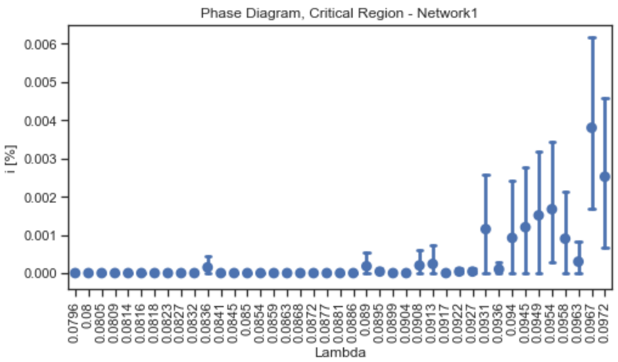

Figure 3: Phase transition at the critical point for Network 1.

decisive in the epidemic spread, since I also increases fast with $\lambda > \lambda_c$. (Fig.4) With the same considerations mentioned above, it results that $\lambda^2_{c,NDlib} = 0.094$ for $I = 0.0020^{+0.0036}_{-0.0006}$. Also this estimation is consistent with the theoretical results.

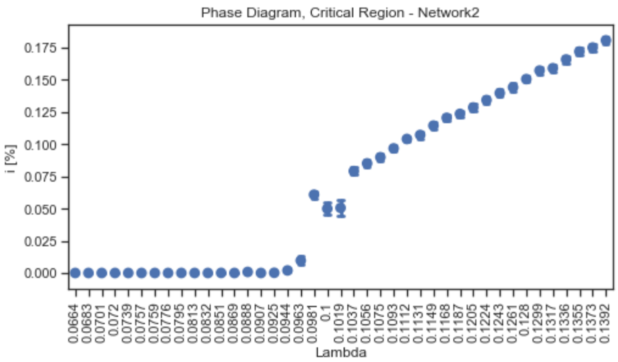

Figure 4: Phase transition at the critical point for Network 2.

2.2 Gillespie Algorithm

An alternative and efficient way to simulate this system is to resort to Gillespie Algorithm. This method allows the compiler to avoid useless steps in which no interactions occur, thus being faster. Given a set of parameters and a certain initial configuration as seen previously, it works with the following iterations:

I It computes the propensity rates of the current configuration as

$$\begin{cases} a_1 = \lambda l \\ a_2 = \mu I \end{cases}$$

where l is the number of edges between I–S. a_1 is associated to $I + S \longrightarrow I + I$ reaction, while a_2 to $I \longrightarrow S$. We define $a_0 = \sum_i a_i$;

II It samples t, time at which the next reaction occurs, from the exponential distribution $\frac{dp(t)}{dt} = a_0 e^{-a_0 t}$;

III It chooses which reaction takes place with probability $P(I + S \longrightarrow I + I) = \frac{a_1}{a_0}$ and $P(I \longrightarrow S) = \frac{a_2}{a_0}$. Then, with an uniform distribution it decides in which edges/nodes among all the possible ones the infection/recovering occurs;

IV It updates the state.

Following this procedure, at each step we have a reaction but, obviously, they are different time long.

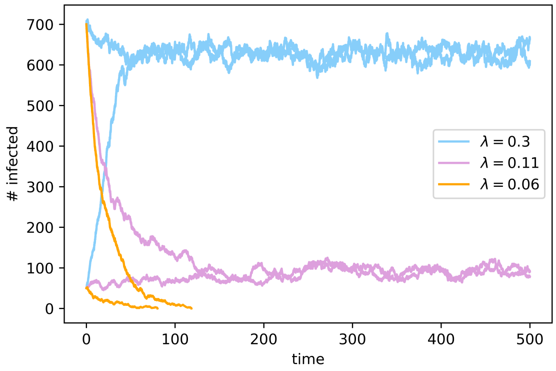

Figure 5: Different simulations with different λ with Gillespie Algorithm. We notice that even if they start from different initial condition they reach they reach the same stationary state as theory predicts.

As before, to obtain the phase diagram (Fig.2), we varied λ from 0 to 1 with steps of 0.02. At fixed parameter, we iterated 50 simulations of the Gillespie algorithm, 10000 interactions long. In order to estimate the $< I >$ of the stationary state, we averaged only the last 1000 interactions.

To focus on finding the critical points λ_c (Fig.6) we repeated the same procedure with $0.08 \leq \lambda^1 \leq 0.11$ and $0.08 \leq \lambda^2 \leq 0.11$ varying the parameter with 0.001 of precision. Due to noise in this delicate region, we decided to iterate 1000 times for each λ instead of 50, in order to have a better statistics taking advantage of the simulation speed. Indeed, from Fig.6 notice that the error bars are smaller than in the previous case, and the global behaviour is smoother.

$$\lambda^1_{c,Gill} = 0.098; \quad I = 0.30^{+0.59}_{-0.09}$$
$$\lambda^2_{c,Gill} = 0.093; \quad I = 0.36^{+0.64}_{-0.16}$$

In spite of that, these values are still reasonable and our self-made implementation provides results that are notably similar to the affimed library. Even so, for $\lambda >> \lambda_c$ we clearly see the difference between the two stochastic processes. Indeed, in the first implementation we have a discrete time in which we attempt 1000 modifications at once. This means that we are neglecting the updating of the status of the nodes during this step and the relative variation of the propensity rate. For example, to make this problem more clear, if we consider a node which is infected at time t, it can heal at time $t + 1$, but it cannot get immediately reinfected at same time $t+1$. This fact limits our dynamics preventing too high stationary $< I >$ differently from the second algorithm.

Furthermore the speed of the Gillespie algorithm allowed us to zoom very well around the critical point. In this way we can clearly note the finite side-effects which makes the transition between the two phases "smoother".

Despite these differences, these two algorithms give similar results for λ_c and the slight different shapes between the phase diagrams could be a hint that could indicate the different structures of the networks.

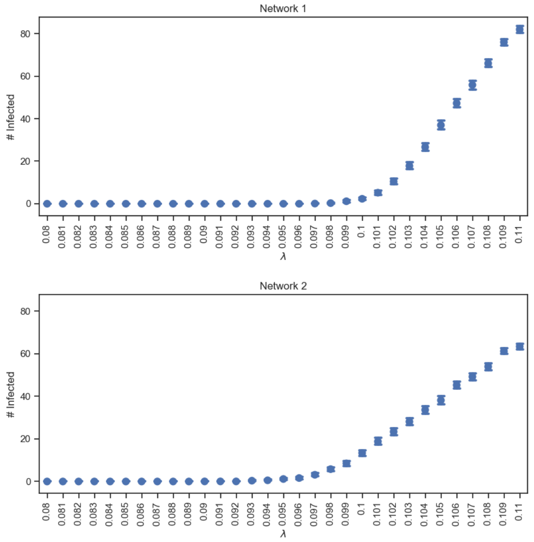

Figure 6: Phase transition close to the critical point obtained with our implementation of the Gillespie Algorithm.

Afterwards, we decided to analyse a variant of the Gillespie method, using the approximation $a_1 \approx pIS$, where p is the probability of a given node to have an edges $\frac{<k>}{N-1}$. We expect that this approximation could work very well for poissonian networks and it's not accurate for real systems, since knowing which nodes are infected could be important to evaluate the evolution of the propagation.

As you can see from Fig.7 we get quite similar results for the first network, while different values for the second one. This could underline once more the different structures of the two networks.

3 Networks Characterisation

In this section we analyse some structural proprieties of the two given networks.

They both have exactly 1000 nodes, Network 1 has 2573 edges with average degree $< k >_1 = 5.146$, whereas Network 2 has 1796 edges with $< k >_2 = 3.592$. They are both made of one connected component, but their *connectivity* is 1 and 2 respectively. This means that just this amount of people connects two groups

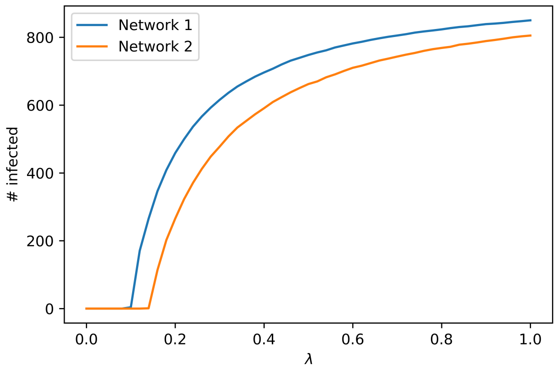

Figure 7: Phase transtion of the two network with Gillespie algorithm using an approximated propensity rate.

that would be otherwise unrelated, thus they play a crucial role if the epidemic spread in one group at first and they would get sick after some time. In the original file, these node are indexed as 848 for Network 1 and as 876, 933 for Network 2.

The *degrees* distributions is shown in Fig.8. Network 1 has a bell-shaped distribution, similar to a Gaussian or Poissonian, meaning that all the nodes are roughly the same number of edges. The two most connected nodes (labelled as 131 and 917) have 14 edges.

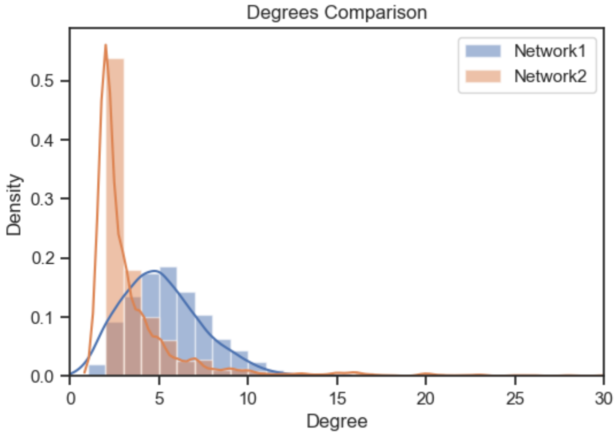

Figure 8: Degrees distributions of the two networks with overlapping KDE.

On the contrary, the edges of Network 2 follow a exponential (or power law) heavily-tailed distribution with 12 people that have 20 or more links, so they are more exposed to infections and they will infect more people in turn. Besides, it turns out that the most connected person, with $k = 30$, is also one of the two that if removed would disconnect the graph (node 876). Notice that this is not valid for Network 1, that seems to follow a more random structure.

The *PageRank* distributions (Fig.9) display similar shapes. Even though pageranking would be best suited for directed graphs, once again we found that the nodes with the highest pageranking are also the ones with highest number of links.

We computed the average *clustering coefficients* that are $< c >_1 = 0.0047$ and $< c >_2 = 0.0158$ and they are both close to zero. Indeed, the value of the clustering coefficient is exactly 0 for 950 nodes in Network 1 and for 926 in Network 2. This indicate that for most of the nodes there is not a triangular connection and this should slow down the epidemic diffusion. Just for aesthetic reasons, we discarded these values and plot the KDE for the remaining ones in Fig.10. Observe that there are nodes with $c = 1$, just one for Network 1 and eleven for Network 2. These nodes are fully connected among their neighbours and therefore they could be a sort of hot spots for the epidemic, but a deeper analysis teaches us that they all have $k = 2$, thus being rather isolated and not that significant.

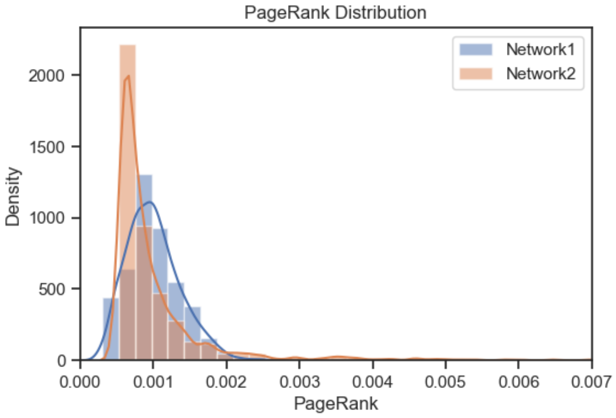

Figure 9: PageRank distributions of the two networks with overlapping KDE.

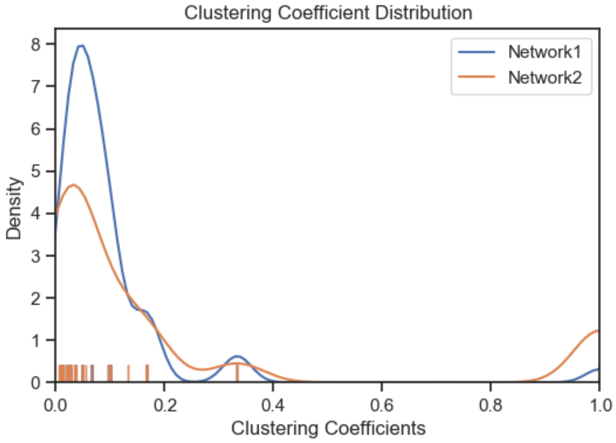

Figure 10: Clustering Coefficients KDE and rugplot for the values different than zero. They are too few and too skewed for a good looking histogram plot.

4 Node Segregation

Given the reasoning made so far, if we were to remove 1% (i.e. 10) of the individuals with the aim to prevent as much as possible the epidemic outbreak, one possible and rational criterion would be by picking the nodes with the highest degree plus those that, if removed, disconnect the networks. We followed this modus operandi and simulated once more the SIS model as in section 2 with the `NDlib` tools and for the same values of λ close to the critical point, as shown in Fig.11. As one would expect, these Networks with segregated nodes have a higher λ_c, that is to say that the infection must be more viral since we removed the most contagious individuals. We have that:

$$\lambda^1_{c,seg} = 0.097; \quad I = 0.0007^{+0.0013}_{-0.0001}$$
$$\lambda^2_{c,seg} = 0.117; \quad I = 0.0071^{+0.0100}_{-0.0044}$$

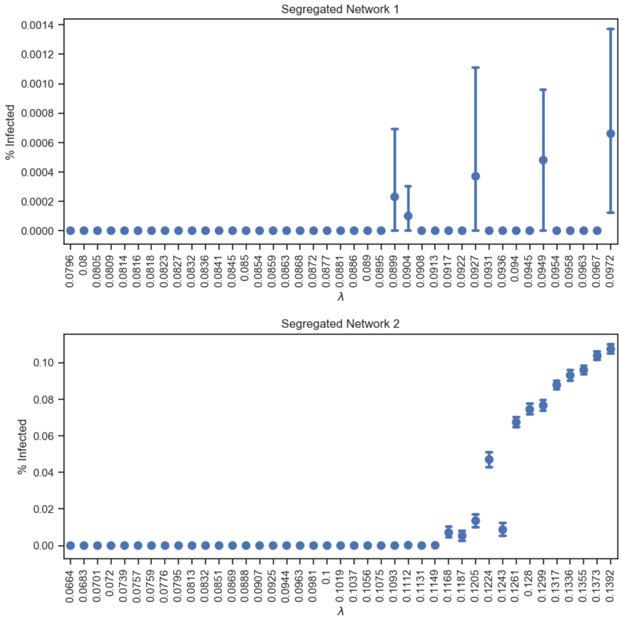

Figure 11: Phase transition close to the critical point for the segregated versions of the two Networks.

In addition, the 90% confidence intervals and the mean amount of infected are smaller and closer to zero than in the previous case. For these reasons we could expand our investigation to higher λ to obtain results with even better confidence levels, even though in epidemiology it is tolerated to underestimate the critical infection rates when dealing with real-world policy making.

5 Study of the early stages of a chicken pox outbreak

In the last part of this work, we analysed data relative to the initial phase of a possible propagation of chicken pox in 100 different localities. The model we had recourse to was the so-called SIR. In this way, we considered three different states: susceptible, infected and removed. The difference between the SIS model is that once an individual recovers, it cannot be infected again. This implies that the disease cannot live for an infinite period of time among the population, as in the SIS, but once there is no more susceptible people it will

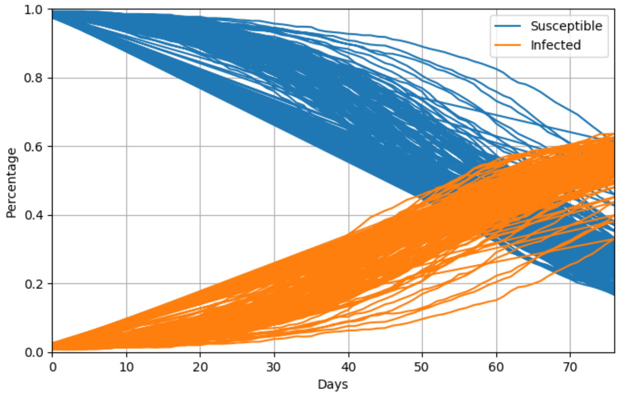

Figure 12: First 77 days of propagation of chicken pox in 100 different localities.

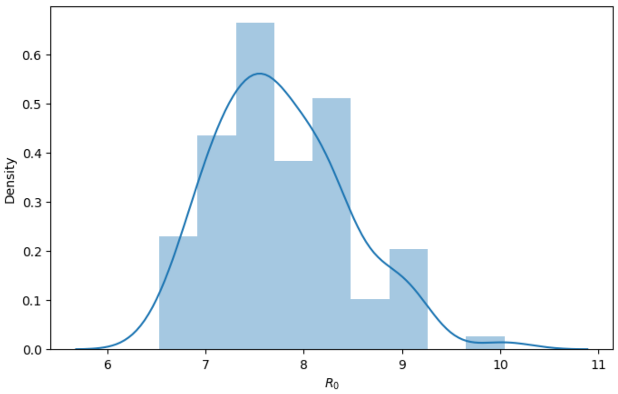

Figure 13: Histogram and KDE of the parameter R_0 estimated from the data.

die out. If we write the equations which rule the dynamics in a mean-field approach we find the following expressions:

$$\begin{cases} \dot{s}(t) = -\beta i(t)s(t) \\ \dot{i}(t) = \beta i(t)s(t) - \gamma i(t) \\ \dot{r}(t) = \gamma i(t) \end{cases} \tag{4}$$

where $s(t)$, $i(t)$ and $r(t)$ are the concentrations of the three states, while the parameters β and γ regulate the velocity of the transition from $S+I \longrightarrow I+I$ and $I \longrightarrow R$. If $R_0 = \beta/\gamma > 1$ the disease propagates, otherwise the time of diffusion is so slow that the recovering process prevents the spread. Therefore, if we want to understand whether the chicken pox outbreak will become viral or not, we have to estimate the parameter R_0. In order to do that, if we look at equations 4, we notice that we can express β and α as

$$\begin{cases} \beta = \frac{s(0)-s(t)}{\int_0^t i(\tau)s(\tau)d\tau} \\ \gamma = \frac{r(t)-r(0)}{\int_0^t i(\tau)d\tau} \end{cases} \tag{5}$$

If our data provides us the concentration of the three states day by day, we can estimate our parameters by using

$$\begin{cases} \beta = \frac{s_1-s_D}{\sum_{k=1}^{D} s_k i_k} \\ \gamma = \frac{r_D-r_1}{\sum_{k=1}^{D} i_k} \end{cases} \tag{6}$$

where the index k refers to the day from 1 to $D = 77$. So we could easily estimate the parameters for each location and the corresponding R_0. We obtained the following results:

$<\beta>$	$<\gamma>$	R_0
0.0750 ± 0.0026	0.00971 ± 0.00076	7.77 ± 0.67

Table 1: The values, obtained averaging over the all 100 locations, are reported with their respective standard deviation.

On the whole we can say that $R_0 >> 1$ so the chicken pox will spread eventually. Nevertheless, it's very hard to estimate R_0 precisely with this data, since we have such a large variance. One could try neglecting the R_0 of given locations which are too distant from the mean. To be precise we eliminated the location for which $|R_0 - <R_0>| \geq 3\sigma$. In this case we obtain a new mean with the data remained $R_0 = 7.75$.