Università degli Studi di Padova

Department of Civil, Environmental and Architectural engineering

Master degree in Mathematical engineering

On the interplay between disease and awareness spreading in multiplex networks

Supervisor:
Prof. Amos Maritan
External supervisor:
Prof. Yamir Moreno
(BIFI - Zaragoza)

Candidate:
Francesca Priante
Student number:
1129915

Academic year 2017/2018

Abstract

Here it's presented a model with the aim of integrating, using the multiplex networks environment, the awareness and the behavior of individuals in case of an emergence of an infectious disease within a population. Aware people are assumed to do as lower contacts between each others as possible, so that they decrease their probability of getting infected. The position of the critical point is investigated both theoretically and numerically using a methodology based on mean-field approximation and Monte Carlo simulations.

Contents

Introduction

The mathematical formulation of epidemiology is a fundamental tool in order to model, understand and then predict the behavior of a infectious disease and it has became necessary for a developed world. Hence many researchers have been working in this subject for centuries, starting from the first studies in 1663 made by John Graunt, leading to the creation of methods for containment and prevention. However for centuries the population has been treated as an homogeneous medium behaving at the same way and passively reacting to the spreading process of the infection. Only in the few past decades researchers have realized the importance of the interaction between human behavior and the spreading of the infection [14, 19]: individuals actions are even more difficult to model than the infection itself since many assumptions need to be done so that it can be possible to generalize the same reaction to the whole population. In this framework it will be presented a model based on the reduction of contacts made by individuals when they become aware of the disease. From Chapter 1 to Chapter 3 there will be some general introductions to make the reader comfortable to topics like complex networks and epidemiology. Chapter 1 introduces the classical epidemic models giving a glimpse to concepts like *basic reproducing number*, the threshold and their connections to statistical mechanics. In Chapter 2 some definitions for complex (and multilayers) networks are given, followed by some example of networks that have been used in this research. In Chapter 3 there is the connection between the aforesaid chapters, that is the modeling of epidemics in the networks environment, with particular interest to the contact based SIS model. With Chapter 4 it begins the actual research, where the problem of how to model the awareness and the information spreading is addressed.

While the disease is spreading in a specific network the information spreading process takes place in a different one (that can be seen as a virtual network such as Twitter or Facebook) and the analytical formulation of the model is here formulated. The research of the threshold between absorbing phase and epidemic phase is carried forward and an explicit expression is found in the simplest case of homogeneous networks. Considering that the complexity of the problem brings the analytical solutions to be restricted only to the simplest cases, numerical simulations are necessary to tackle the problem, whose methods, solutions and discussions are shown in Chapter 5. At the end of Chapter 5 some numerical methods are then discussed to define the critical point always using heavy Monte Carlo simulations and scaling theory from statistical mechanics; and finally there are some final considerations and perspectives.

Chapter 1

Classical epidemic models

1.1 Brief history of mathematical epidemiology

The very first statistical study of infective disease [5] is due to John Graunt's book "Natural and Political Observations made upon the Bills of Mortality" (1663). The bills consisted in weekly records of number and causes of death in the city of London and gave an estimate of the probability of dying of a certain disease. One century later Daniel Bernoulli made what is known as the first mathematical model for infectious disease when he was trying to analyze the mortality from smallpox. In 1766 he published a paper where he suggested that the inoculation of a live virus from a mild smallpox case would have decreased the mortality in the population although with a little possibility of death by the inoculation itself ([1],[2]). The father of modern epidemiology is Ronald Ross who won the Nobel prize in 1902 for his studies on malaria: he discovered that malaria was carried by mosquitos and that if they would have been reduced under a critical value it would be sufficient to contain the disease [24] introducing the concept of *basic reproducing number*, which is the average number of secondary infections that an infected individual makes in a given time interval.

1.2 The SIR model

The modern epidemiology is based on the compartmental model, a technique consisting in dividing the population into compartments, with the assumption that every individual in the same compartment has the same characteristics. One of the simplest compartmental model, built by Kermack and McKendrick from 1927 to 1933 [25, 26, 27] assumes that in a fixed population with size N, individuals can be in three different states:

- **Infected individuals (I)**: are those that carry the infection. If the infected individual can also spread the disease then it's also called infectious.
- **Susceptible individuals (S)** are those who are not infected but can contract the infection if exposed to the disease.
- **Recovered individuals (R)** are those who were once infected but then they recovered and cannot contract the disease anymore.

If we consider a discrete time evolution of the process, in every time step a susceptible individual can be in contact with an infected individual and become infected with a certain probability. In the same time step an infected individual can recover spontaneously with another probability, without any interaction with the others. Let's call μ the probability that an infected individual recovers and β the probability that a susceptible individual, meeting an infected, contracts the disease. Both of them can be estimated by clinical data, depending on the type of the disease, μ^{-1} can also be interpreted as time needed for an infected to heal, on average, so it indicates the *infectious period.*

We can outline the transition as (see also Figure 1.1):

$$\mathrm{S} + \mathrm{I} \xrightarrow{\beta} 2\mathrm{I} \tag{1.1}$$

$$\mathrm{I} \xrightarrow{\mu} \mathrm{R} \tag{1.2}$$

For the moment let's assume homogeneous mixing, that means that individuals are totally equivalent and interact with each others in a completely random way. We call $\rho^I(t)$, $\rho^S(t)$ and $\rho^R(t)$ the densities respectively of infected, susceptible and recovered individuals:

$$\rho^I(t) = \frac{N^I(t)}{N} \tag{1.3}$$

$$\rho^S(t) = \frac{N^S(t)}{N} \tag{1.4}$$

$$\rho^R(t) = \frac{N^R(t)}{N} \tag{1.5}$$

where N^I, N^S and N^R are the numbers of infected, susceptible and recovered individuals. In the continuous time limit case the process can be described by three differential equations by applying the law of mass action:

$$\frac{d\rho^I(t)}{dt} = \beta\rho^I\rho^S - \mu\rho^I \tag{1.6}$$

$$\frac{d\rho^S(t)}{dt} = -\beta\rho^I\rho^S \tag{1.7}$$

$$\frac{d\rho^R(t)}{dt} = \mu\rho^I \tag{1.8}$$

An even simpler model is based only in two states, I and S, that means that once that an individual recovers it will pass directly to the susceptible state and can be reinfected again afterwards (SIS model). The SIS model exhibits a stationary state, called *endemic state*, in which the fraction of infected individuals remains constant.

In this case the transition are (see also Figure 1.1):

$$\mathrm{S} + \mathrm{I} \xrightarrow{\beta} 2\mathrm{I} \tag{1.9}$$

$$\mathrm{I} \xrightarrow{\mu} \mathrm{S} \tag{1.10}$$

and the dynamics is defined by:

$$\frac{d\rho^I(t)}{dt} = \beta\rho^I\rho^S - \mu\rho^I \tag{1.11}$$

$$\frac{d\rho^S(t)}{dt} = -\beta\rho^I\rho^S + \mu\rho^I \tag{1.12}$$

In order to see the dynamics at the early stage time, so when the disease starts to spread, we consider $\rho^I \approx 0$ and we plug it in the first equation

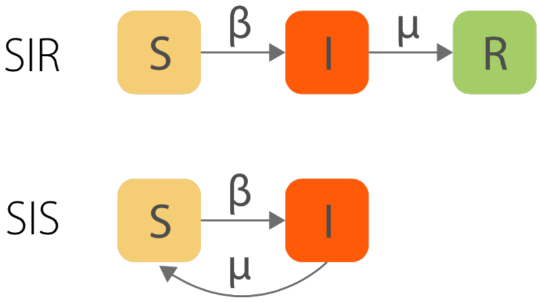


Figure 1.1: Schematic visualization of SIS and SIR processes: the boxes represent different compartmentals and the arrows the transitions between compartments, happening stochastically according to their respective rates [22].

(substituting ρ^S with $1 - \rho^I$):

$$\frac{d\rho^I(t)}{dt} \approx (\beta - \mu)\rho^I \tag{1.13}$$

where the second order term $(\rho^I)^2$ has been neglected. The solution is then:

$$\rho^I(t) = \rho^I(0)e^{(\beta-\mu)t} \tag{1.14}$$

This means that the density of infected individuals increase exponentially when $\beta - \mu > 0$. This leads to define the *basic reproducing number*

$$R_0 = \frac{\beta}{\mu} \tag{1.15}$$

that is one of the most important parameters in the framework of theoretical epidemiology since it defines the average number of secondary infections caused by one infected individual immersed in a fully susceptible population.

So if $R_0 > 1$ then the disease will lead to an outbreak, if $R_0 < 1$ it will die out and if $R_0 = 1$ it means that on average an individual infects just other one and then it loses the infection and the density remains on average constant at the initial value ρ_0. The reproducing number has that form just for this particular case in which we considered random homogeneous mixing, in general it can be a less straightforward function of the parameters; for example for an homogeneous contact network in which each individual contacts the same number of people $\langle k \rangle$ it is:

$$R_0 = \langle k \rangle \frac{\beta}{\mu} \tag{1.16}$$

The epidemic dynamics shows therefore a *threshold behavior*, see Figure 1.2, that is typical of the phase transition phenomena in non-equilibrium systems in statistical physics. A phase transition is defined as an abrupt change of state of a system that can be seen looking at an *order parameter* ρ that is zero in a phase and non zero in the other. The phase transition is due to the change of another parameter called *control parameter* λ that has a particular value λ_c in which the phase transition occurs.

$$\begin{cases} \rho = 0 \text{ if } \lambda \leq \lambda_c \\ \rho > 0 \text{ if } \lambda > \lambda_c \end{cases} \tag{1.17}$$

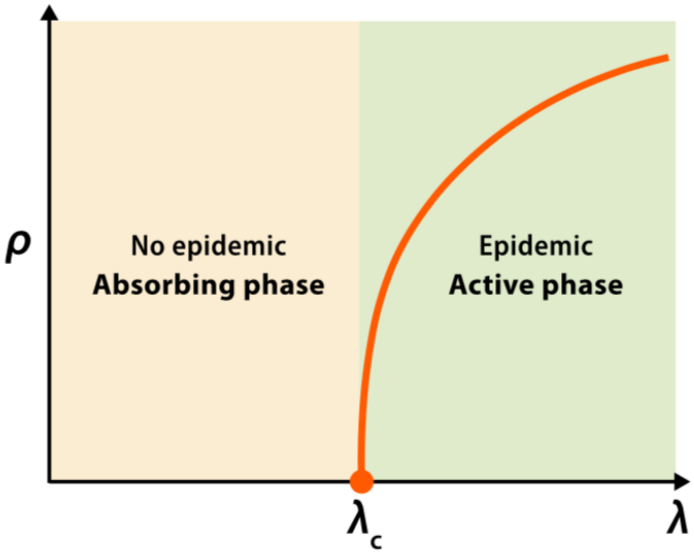


Figure 1.2: Phase transition diagram [23].

Phase transition can be continuous or discontinuous transition: the first one means that the order parameter is continuous with respect to the control parameter (but can exhibits a discontinuity in the derivative) the second one means that the order parameter is discontinuous. In the case of continuous phase transition the order parameter follows a power law behavior:

$$\rho(\lambda) \sim (\lambda - \lambda_c)^d \tag{1.18}$$

where d represents the critical exponent.

In the SIS model the order parameter is the density of infected individuals at the steady state and the control parameter is the effective infection rate, i.e. β/μ, that at the threshold (or critical point) it separates the healthy state from the infected.

Chapter 2

Complex networks

2.1 Some definitions

Networks (or graph) [3] can be described as a collection of points (or vertices) connected by edges (or link); formally, given the set N of the nodes $(n_1, n_2, ...n_N)$ and the set E of the edges $(e_1, e_2, ..., e_h)$ we can write the graph G as $G = (N, E)$ that can be also written as $G_{N,E}$. From a mathematical point of view a network can be completely written using an adjacency matrix: given N nodes, the adjacency matrix A is a $N \times N$ matrix which entries a_{ij} are one if there is an edge connecting i with j, otherwise are zero:

$$\begin{cases} a_{ij} = 1 & \text{if } (n_i, n_j) \in E \\ a_{ij} = 0 & \text{otherwise} \end{cases} \tag{2.1}$$

A graph can be either directer or undirected: directed means that the direction of the edge is important i.e., calling e_{ij} the edge from i to j, $e_{ij} \neq e_{ij}$, while the contrary holds for undirected nodes where $e_{ij} = e_{ji}$. In this framework we will be only using undirected networks.

The **degree of a node** is the number of edges coming out from it, so:

$$k_i = \sum_j a_{ij} \tag{2.2}$$

and the degree distribution $P(k)$ is the probability that, chosen a random node, it has degree equal to k. Networks can be divided into homogeneous networks, where all the nodes have all the same degree $k_i = \langle k \rangle \ \forall i$, or heterogeneous networks which follow a generic degree distribution $P(k)$.

Two vertex degree correlation can be studied looking at the probability that an edge leaving a node of degree k will reach a node of degree k', $P(k'|k)$. The correct computation of $P(k'|k)$ is although problematic in finite dimensional networks, so it's simpler to study the **average degree of the nearest neighbors** of vertices of degree k:

$$\bar{k}_{nn}(k) = \sum_{k'} k' P(k'|k)$$

Uncorrelated networks are those in which the probability that a node with degree k is connected to a node of degree k' is independent on k that means that an edge is more likely connected to a node with high connectivity. Therefore the probability that a link points to a node with k' edges is proportional to $k'P(k')$; adding the normalization it comes:

$$P(k'|k) = \frac{k'P(k')}{\sum_{k'} k'P(k')} = \frac{k'P(k')}{\langle k \rangle}$$

So for uncorrelated networks:

$$\bar{k}_{nn}^{un}(k) = \frac{\langle k^2 \rangle}{\langle k \rangle}$$

that doesn't depend on k.

Moreover, for homogeneous uncorrelated networks, since $P(k) = 1$ when $k = \langle k \rangle$:

$$\bar{k}_{nn}^{ho}(k) = \langle k \rangle$$

Another parameter useful to describe networks is the clustering coefficient which is the tendency of two nodes of being connected given that they already share a common neighbor. The clustering coefficient C is defined as the ratio between the number of loops of length three in the network, and the number of connected triples (three nodes connected by two edges).

2.2 Most used types of networks

2.2.1 Erdös-Rényi graphs

The first random graph models were proposed by Paul Erdos and Alfréd Renyi in 1959, who suggested two different ways to generate random networks. The first one was a model to generate graphs with N nodes and M edges, $G^{ER}_{N,M}$, so, starting from N disconnected nodes, edges are added to random couples of non-connected nodes (therefore avoiding multiple connections) until the number of edges equals M. The average degree in this case is:

$$\langle k \rangle = 2M/N$$

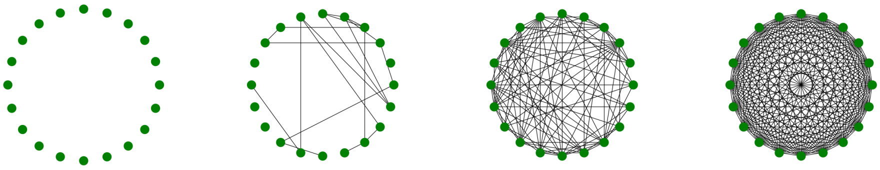
The second one was a model to generate graphs with N nodes which are linked together with given probability $0 < p < 1$, $G^{ER}_{N,p}$ (see Figure 2.1). The average degree in this case is:

$$\langle k \rangle = p(N-1)$$

The two families coincide in the limit of large N: $N \to \infty$. In this condition and for fixed $\langle k \rangle$ the degree distribution is Poissonian:

$$P(k) = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}$$

Erdos-Renyi graphs are uncorrelated by definition, since the connection between nodes is not depending on the degree and it happens with the same probability.



(a) $p = 0$ *(b)* $p = 0.1$ *(c)* $p = 0.5$ *(d)* $p = 1$

Figure 2.1: Erdös-Rényi graphs changing the probability of creating a link

2.2.2 Power-law graphs

The recent study of real networks showed that the Erdos-Renyi models are not adequate to describe real graphs topology. Indeed, real networks, are characterized by short path length, high cluster coefficient, degree correlation and heavy tail degree distribution. Most of the real networks show a power law degree distribution:

$$P(k) \sim ck^{-\gamma}$$

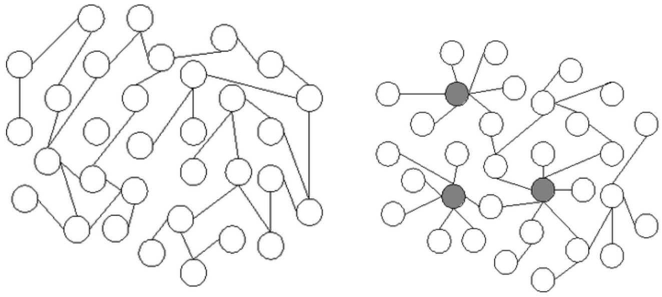
This means that these networks are characterized by having few high degree nodes (*hubs*) and the most of the nodes with low degree. Power-law (or similarly "scale-free") networks are for example: internet, social networks, airlines networks, protein-protein interaction networks.

It's useful to know which is the maximum eigenvalue of the adjacency matrix of a graph since, as we will see soon, it's strictly connected to the threshold of the epidemic. For a power-law graph it has been proved [7] that

$$\Lambda_1 \sim max(\sqrt{k_{max}}, \left\langle k^2 \right\rangle / \langle k \rangle) \tag{2.3}$$

where k_{max} is the maximum degree of the network.

In the thermodynamic limit, in the case in which $2 < \gamma < 3$ the second order moment diverges because of the precence of big degree oscillations.



(a) Erdös-Rényi *(b) Scale free*

Figure 2.2: Comparison between Erdös-Rényi and Scale free networks [6]

2.2.3 Multilayer networks

Formally a multilayer network [4, 15] is a pair $\mathsf{M} = (\mathsf{G},\mathsf{C})$ where $\mathsf{G} = \{G_\alpha : \alpha \in \{1, ..., M\}\}$ is a set of graphs that can be thought as layers and C represents the connections between the layers (*interconnection*). If we have a network characterized by having different aspects, the latters can be splitted into different layers of networks, becoming a *multilayer network*. So let's suppose to have d aspects, we can define a sequence $\mathbf{L} = \{L_\alpha\}_{\alpha=1}^{d}$ of layers, one for each aspect a.

In this framework we will be talking about *multiplex networks*, see Figure 2.3 that are those networks characterized by having the same set of nodes for each layer and for which the only possible type of interlayer connections are those in which a given node is only connected to its counterpart nodes in the rest of layers.

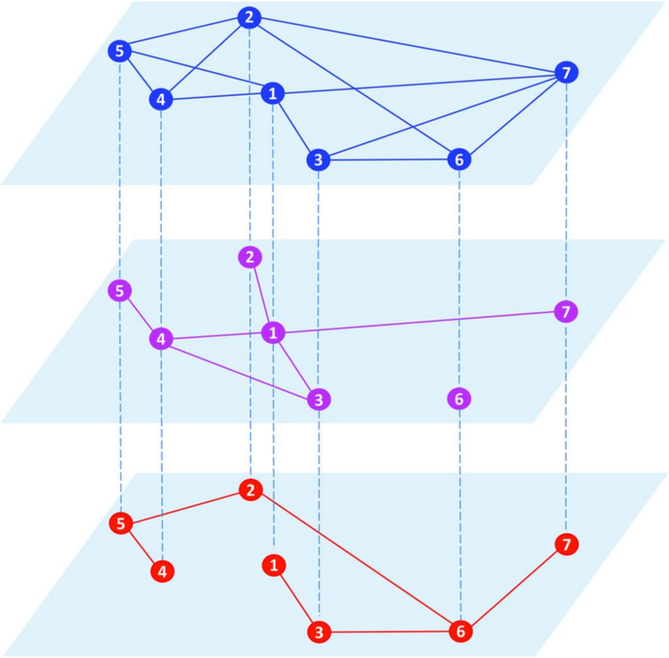


Figure 2.3: Example of multiplex network [21]

Chapter 3

Epidemics in networks

3.1 Motivation and methods

Epidemics in networks can be approached using the Markov theory, in which each node i at time t belongs to a state that is described by the values of a random variable $X_i(t) = \{0, 1, ...s - 1\}$ where s is the number of possible states. This means that for an SIS model $X_i(t)$ will take values $\{0, 1\}$. Therefore if the nodes are n, there are s^n possible configurations of the system that can be encoded in the infinitesimal Markov chain generator matrix, which will be a $s^n \times s^n$ matrix. Once the initial configuration and the infinitesimal generator are known it's possible to find the states probability at time t with matrix operations. Although this method is exact, it appears to be unfeasible for many reasons [22]: the set of s^n equations limits the solution to small system sizes and in general the structure of the infinitesimal generator Q is rather complex, which prevents one from obtaining general insights. Therefore many other simpler methods are built from this one, such as the mean field approximation, that will be discussed in the following section.

3.2 SIS contact based model

Consider a network made of n nodes with adjacency matrix A_{ij}. Let's examine an SIS discrete time contact-based model where $X_i(t)$ fully represents the state of node i at time t: we say that $X_i(t) = 1$ when it's infected and

$X_i(t) = 0$ when it's susceptible. Each node can represent either an individual, a city or an airport and the edges between nodes are any kind of channels where the disease can spread along (physical contact between individuals, streets or flight lines). Then we consider the following spreading process: at each time step a susceptible node i can visit its neighbors making λ trials and it can be infected by the infectious ones with probability β. The infected individuals can also recover with rate μ every time step. This creates a Markov chain where the configuration of the systems at time t depends only of the configuration at the previous time step. We define as r_{ij} the probability that the node i is in contact with node j given that the node i makes λ trials; thus we can define a *contact matrix*:

$$R_{ij} = 1 - \left(1 - \frac{A_{ij}}{k_i}\right)^{\lambda} \tag{3.1}$$

where k_i is the degree of node i. If λ is 1 then the node i will do just one contact per time step, R_{ij} will be equal to $\frac{A_{ij}}{k_i}$ and the process is called *contact process.* While if the number of contacts is really big and $\lambda \to \infty$, then R_{ij} will be equal to A_{ij} and the process is called *fully reactive process*, that means that the probability that two neighbors are in contact between each other is equal to 1.

We define then a Bernoulli random variable $Q_i(t)$ that is 1 if node i doesn't get infected by any neighbors and 0 if it gets infected by at least 1 neighbor, and it can be written in function of the aforesaid terms:

$$Q_i(t) = \prod_{j=1}^{n} (1 - \mathcal{B}(\beta)\mathcal{B}(R_{ij})X_j(t))$$

where $\mathcal{B}(b)$ defines a Bernoulli random variable with rate b. So now it's possible to write the discrete evolution of the process as:

$$X_i(t+1) = X_i(t)(1 - \mathcal{B}(\mu)) + (1 - X_i(t))(1 - Q_i(t)) \tag{3.2}$$

which says that node i becomes infected only if it was already infected at the previous time step and it doesn't recover (first right hand side term), or it was not infected but it becomes at this time step (second right hand side term). Now we take the expectation of the equation 3.2 and we apply a mean field approximation, that means that the expected values of tuple of random variables factorize. All the variables are Bernoulli random variables therefore taking the expectation is the same as considering the probability of the successful event.

$$\mathbb{E}(X_i(t)) = \mathbb{P}(X_i(t) = 1) = x_i(t) \tag{3.3}$$

will finally end up with:

$$x_i(t+1) = x_i(t)[1-\mu] + [1 - x_i(t)][1 - q_i(t)] \tag{3.4}$$

with

$$q_i(t) = \prod_{j=1}^{n} (1 - \beta R_{ij} x_j(t)) \tag{3.5}$$

The first right hand side term is the probability of already being infected at time t and don't recover and the second right hand side term is the probability of becoming infected at this time step. While $q_i(t)$ represents the probability of avoiding the infection at time t, so that $(1 - q_i(t))$ is the probability of getting infected at time t.

Waiting for $t \rightarrow \infty$ we get the steady state equation:

$$p_i = (1 - q_i) + (1 - \mu) p_i q_i \tag{3.6}$$

3.3 Epidemic threshold

Let's define the *density of infected* as the average probability of infection among the individuals:

$$\rho = \frac{\sum_{i=1}^{n} x_i}{n}$$

We want to look for which values of β we have the onset of the equation, so for which β_c, fixed μ and λ, we get $\rho = 0$ if $\beta \leq \beta_c$ and $\rho > 0$ if $\beta > \beta_c$.

This critical point is found by considering that as long as $\beta \to \beta_c$ [13], the probability of being infected is really low, $x_i = \epsilon_i << 1$. Substituting in equation 3.5 and neglecting the second order terms:

$$q_i(t) \approx 1 - \beta \sum_{j=1}^{n} R_{ij}\epsilon_j$$

Plugging into 3.6:

$$\sum_{j=1}^{n} \left(R_{ij} - \frac{\mu}{\beta}\delta_{ij} \right) \epsilon_j = 0 \qquad \forall i = 1, ..., n \tag{3.7}$$

whose solution is non trivial only if μ/β is the eigenvalue of the matrix R. Since we are looking for the onset of the epidemic, we want to know the lowest value of β that satisfies the latter condition, that is:

$$\beta_c = \frac{\mu}{\Lambda_{max}} \tag{3.8}$$

where Λ_{max} is the maximum eigenvalue of the contact matrix R. The Perron-Frobenius theorem ensures us that the maximum eigenvalue is positive, see Appendix A

In the case of a contact process (when $\lambda = 1$) the only solution corresponds to $\beta = \mu$ since R becomes the Markov-chain transition matrix whose maximum eigenvalue is always 1. While for a fully reactive process (when $\lambda \to \infty$) it depends on the network, for example, for uncorrelated scale-free networks it's:

$$\beta_c = \mu \frac{\langle k \rangle}{\langle k^2 \rangle} \tag{3.9}$$

In Figure 3.1 it's visible how the threshold changes with the number of

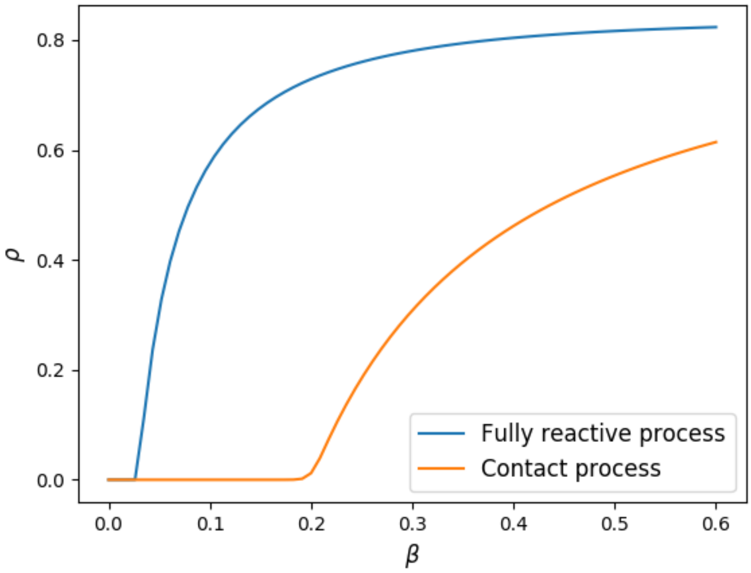


Figure 3.1: Phase diagram of the SIS contact based process in the case $\lambda = 1$ (Contact process), $\lambda \to \infty$ (Fully reactive process). Erdös-Rényi graph with $\langle k \rangle = 6$ and recovering rate $\mu = 0.2$

contacts λ; in this case the eigenvalue of R is 7.11 so that the critical point when in the reactive process when $\mu = 0.2$ is at $\beta_c = 0.028$. On the other hand if it's a contact process the critical point is at $\beta_c = 0.2$.

Since the second order moment diverges with n, the critical point for Scale-free graphs will approach zero as the size of the system increases.

Chapter 4

Modelling the awareness

4.1 Formulation of the model

Let's suppose now that, besides this epidemic spreading process, also information spreading happens, creating awareness within the population. So here it rises a natural question: how to model the awareness?

The most intuitive way of doing this is to use a multilayer network with 2 layers, one is related to the disease spreading and the other to the information spreading. Then we consider that the only interconnections between layers are made by each node and its counterpart in the other layer. So there is a one to one correspondence between the two layers (multiplex network). Let's note with index 1 the elements (networks, degrees, adjacency matrix) of the information layer and 2 the elements of the disease layer. The first layer can be though as a "virtual" general social network where the information is broadcast and the second one is a real network where each node corresponds to a single individual. This problem has already been modeled in some ways before, for example C. Granell, S. Gomez and A. Arenas [14] made the infection rate change with the awareness that can be interpreted as aware people taking preventatively medicines in order to avoid the infection. However the idea now is to act on the activity of the individual and decrease its number of contacts in the disease layer according to its awareness state. We say that an individual can be in 3 different states: infected (I), susceptible and aware (A), susceptible and unaware (U). As it has been done for the normal SIS model in 3.1, these states can be associated to Bernoulli random variables $X_i(t)$,$Y_i(t)$, $Z_i(t)$, so that the state is fully described by the tuple $(X_i(t), Y_i(t), Z_i(t))$:

$$\begin{cases} (1,0,0) & \text{if node } i \text{ is infected} \\ (0,1,0) & \text{if node } i \text{ is susceptible and aware} \\ (0,0,1) & \text{if node } i \text{ is susceptible and unaware} \end{cases}$$

Let's say that the system has the following possible transitions:

$$\mathrm{A} + \mathrm{I} \xrightarrow{\beta_2} 2\mathrm{I}$$

$$\mathrm{U} + \mathrm{I} \xrightarrow{\beta_2} 2\mathrm{I}$$

$$\mathrm{I} + \mathrm{U} \xrightarrow{\beta_1} \mathrm{I} + \mathrm{A}$$

$$\mathrm{I} \xrightarrow{\mu_2} \mathrm{U}$$

$$\mathrm{A} \xrightarrow{\mu_1} \mathrm{U}$$

the symbols above the arrows represent the probabilities to pass in the right hand side state and they are named as:

- β_1 = information rate
- β_2 = infection rate
- μ_1 = forget rate
- μ_2 = recovery rate

In these transitions some assumptions have been done:

1. An aware individual can also forget the about the disease spreading, passing then to the unaware state with rate μ_1
2. When an individual recovers it goes directly to the unaware (or ignorant) state.
3. An aware and an unaware individual, *once in contact* with an infected individual, contracts the disease with the same rate β_2.

4. Infected individuals are the core of both the spreading processes: they pass both the infection and the information that they are infected (not necessarly to the same individual, since this happens in two different layers).

From the third point it can be seen that the infection rate is not suppose to change with the awareness; one could think that an aware node could try to prevent the disease taking some kind of vaccines or other medicines in order to contrast the infection. Nevertheless, as already said, in this framework the aim is to model the behavior of the individuals with their *activity*, which directly manifests itself in the number of contacts (in the disease layer). The hypothesis is then that as long as an individual is aware, it will do as few contacts within its neighbors as possible in order to avoid the disease, so let's say 1 contact, $\lambda^A = 1$. It's preferable not to choose $\lambda^A = 0$ otherwise it would mean a completely isolation from the rest of the world which is not realistic. On the other hand an unaware individual keeps doing the normal number of contact $\lambda^U = \lambda$ as usual. This concept can be written using the contact matrix that has been shown in the previous chapter:

$$(R_2^A)_{ij} = 1 - \left(1 - \frac{(A_2)_{ij}}{(k_2)_i}\right)^{\lambda^A} \overset{\lambda^A=1}{=} \frac{(A_2)_{ij}}{(k_2)_i} \tag{4.1}$$

$$(R_2^U)_{ij} = 1 - \left(1 - \frac{(A_2)_{ij}}{(k_2)_i}\right)^{\lambda} \tag{4.2}$$

The contact matrix for the information layer is:

$$(R_1)_{ij} = 1 - \left(1 - \frac{(A_1)_{ij}}{(k_1)_i}\right)^{\lambda} \tag{4.3}$$

We assumed here that the number of contacts made in order to spread the information is the same amount as the one made by an unaware individual in the second layer.

In the case in which the number of contacts is really high, R^U has only 1 or 0 entries and the infection process will be a mix of a contact process and a fully reactive process.

Now it's possible to start building the equations, call Q_i^A, Q_i^U and V_i

the Bernoulli random variables that respectively are equal to 1: whether the individual avoids the infection while it belongs to the aware state (Q_i^A), if the individual avoids the infection while it belongs to the unaware state (Q_i^U) and if the individual doesn't get informed (V_i).

$$Q_i^A = \prod_{j=1}^{n}[1 - \mathcal{B}(\beta_2)\mathcal{B}((R_2^A)_{ij})X_j] \tag{4.4}$$

$$Q_i^U = \prod_{j=1}^{n}[1 - \mathcal{B}(\beta_2)\mathcal{B}((R_2^U)_{ij})X_j] \tag{4.5}$$

$$V_i = \prod_{j=1}^{n}[1 - \mathcal{B}(\beta_1)\mathcal{B}((R_1)_{ij})X_j] \tag{4.6}$$

As it has been said before only infected individuals spread the information of the infection, otherwise it would be necessarily to add another productory in equation 4.6 (this is the reason why there is X_j inside the equation) $\prod_j[1 - \mathcal{B}(\beta_1)\mathcal{B}((R_1)_{ij})Y_j]$, making the equation less handy.

$$\begin{cases} X_i(t+1) = X_i(t)[1 - \mathcal{B}(\mu_2)] + W_i(t)[1 - Q_i^A(t)] + Y_i(t)[1 - Q_i^U(t)] \\ Y_i(t+1) = Y_i(t)[1 - \mathcal{B}(\mu_1)]Q_i^A(t) + Z_i(t)Q_i^U(t)(1 - V_i(t)) \\ Z_i(t+1) = Z(t)Q_i^U V_i(t) + X(t)\mathcal{B}(\mu_2) + Y_i(t)\mathcal{B}(\mu_1)Q_i^A(t) \end{cases} \tag{4.7}$$

As done for the single layer SIS model, applying the expectation, the mean field approximation and exploiting the property of the Bernoulli random variables, it's possible to write everything in terms of the single probabilities (similarly of what it has been done for rumor spreading by [9]):

$$\mathbb{E}(X_i(t)) = \mathbb{P}(X_i(t) = 1) = x_i(t)$$

$$\mathbb{E}(Y_i(t)) = \mathbb{P}(Y_i(t) = 1) = y_i(t)$$

$$\mathbb{E}(Z_i(t)) = \mathbb{P}(Z_i(t) = 1) = z_i(t)$$

$$\begin{cases} x_i(t+1) = x_i(t)[1-\mu_2] + y_i(t)[1-q_i^A(t)] + z_i(t)[1-q_i^U(t)] \\ y_i(t+1) = y_i(t)(1-\mu_1)q_i^A(t) + z_i(t)[1-v_i(t)]q_i^U(t) \\ z_i(t+1) = z_i(t)v_i(t)q_i^U(t) + y_i(t)q_i^A(t)\mu_1 + x_i\mu_2 \end{cases} \tag{4.8}$$

with:

$$q_i^A = \prod_{j=1}^{n}[1-\beta_2(R_2^A)_{ij}x_j] \tag{4.9}$$

$$q_i^U = \prod_{j=1}^{n}[1-\beta_2(R_2^U)_{ij}x_j] \tag{4.10}$$

$$v_i = \prod_{j=1}^{n}[1-\beta_1(R_1)_{ij}x_j] \tag{4.11}$$

The initial point is given by a little percentage (usually 5%) of infected population and the rest is formed by unaware. In this way it's possible to recover the classical SIS contact-based model imposing $\beta_1 = 0$.

It's important to notice that using the mean-field approximation all the expectations coming out from the productories in equations 4.4, 4.5 and 4.6 factorize: this means that we have assumed all the neighbors of i independents. This is true only if their are not each other's neighbors as well, so for example for tree-graphs, that don't exhibit self-loops.

4.2 Threshold

Proceeding as done in the previous chapter, it's possible to study the position of the critical point in the $\rho - \beta_2$ diagram taking $x_i \approx 0 \quad \forall i$ at the steady state and approximating at the productories 4.9, 4.10, 4.11 at first order, getting:

$$q_i^A \approx 1 - \sum_{j=1}^{n}\beta_2(R_2^A)_{ij}x_j \tag{4.12}$$

$$q_i^U \approx 1 - \sum_{j=1}^{n} \beta_2 (R_2^U)_{ij} x_j \tag{4.13}$$

$$v_i \approx 1 - \sum_{j=1}^{n} \beta_1 (R_1)_{ij} x_j \tag{4.14}$$

Plugging them in the first equation of 4.8:

$$0 = -x_i \mu_2 + y_i \beta_2 \sum_{j=1}^{n} (R_2^A)_{ij} x_j + z_i \beta_2 \sum_{j=1}^{n} (R_2^U)_{ij} x_j \tag{4.15}$$

$$\sum_{j=1}^{n} \left(y_i (R_2^A)_{ij} + z_i (R_2^U)_{ij} - \frac{\mu_2}{\beta_2} \delta_{ij} \right) x_j \tag{4.16}$$

Therefore, in order to have a non-trivial solution, the ratio μ_2/β_2 has to be the eigenvalue of the matrix:

$$H_{ij} = y_i (R_2^A)_{ij} + z_i (R_2^U)_{ij} \tag{4.17}$$

$$(\beta_2)_c = \frac{\mu_2}{\Lambda_{max}(H)} \tag{4.18}$$

that depends strongly on the steady state values of the probabilities of being aware and unaware y_i and z_i.

4.2.1 Case $\mu_1 = 0$: individuals don't forget

Before reaching the steady state the system passes through a transient in which the infected individuals spread the information, creating a portion of aware individuals that remain as such even if the disease dies out. Since $x_i \approx 0$ it's possible to write z_i as $1 - y_i$ and if the second order term given by $x_i x_j$ in 4.16 are neglected the matrix H can be written like:

$$H_{ij} = (R_2^U)_{ij} - y_i((R_2^U)_{ij} - (R_2^A)_{ij}) \tag{4.19}$$

which lies always within $(R_2^A)_{ij} \leq H_{ij} \leq (R_2^U)_{ij}$, since $0 < y_i < 1$. Then, given an integer n:

$$||H^n||_\infty \leq ||(R_2^U)^n||_\infty$$

$$||H^n||_\infty^{1/n} \leq ||(R_2^U)^n||_\infty^{1/n}$$

$$\lim_{n\to\infty} ||H^n||_\infty^{1/n} \leq \lim_{n\to\infty} ||(R_2^U)^n||_\infty^{1/n}$$

And the Gelfand's formula for spectral radius of a matrix M states that:

$$\Lambda_{max}(M) = \lim_{n\to\infty} ||M^n||_\infty^{1/n}$$

and then we can say:

$$\frac{\mu_2}{\Lambda_{max}(R_2^U)} \leq (\beta_2)_c = \frac{\mu_2}{\Lambda_{max}(H)} \leq \frac{\mu_2}{\Lambda_{max}(R_2^A)} \tag{4.20}$$

and since $\Lambda_{max}(R_2^A) = 1$

$$\frac{\mu_2}{\Lambda_{max}(R_2^U)} \leq (\beta_2)_c = \frac{\mu_2}{\Lambda_{max}(H)} \leq \mu_2 \tag{4.21}$$

This means that the threshold is different from the classical SIS case and it's shifted to the right in the phase diagram, depending on the probabilities of awareness y_i: the greater they are, the lower will be H_{ij} and the higher will be the threshold, up to the case $y_i = 1\ \forall i$ where $H_{ij} = R_{ij}^A$ producing a contact process and a critical value at $\beta_c = \mu$. However it's important to remark that, as long as $\beta_1 \neq 0$, a fully contact process is impossible since having all the $x_i = 1$ is forbitten because there is always at least 1 unaware node coming from the recovery of the last infected node.

Homogeneous networks

In the case of homogeneous networks $k_i = \langle k\rangle\ \forall i$, and assuming for semplicity $\langle k_1\rangle = \langle k_2\rangle = \langle k\rangle$ making:

$$(R_2^A)_{ij} = \frac{(A_2)_{ij}}{\langle k\rangle} \tag{4.22}$$

$$(R_2^U)_{ij} = 1 - \left(1 - \frac{(A_2)_{ij}}{\langle k\rangle}\right)^\lambda := (A_2)_{ij} R_\lambda(\langle k\rangle) \tag{4.23}$$

with $R_\lambda(x) := 1 - \left(1 - \frac{1}{x}\right)^\lambda$.

Since for homogeneous network the solution doesn't depend on the node, then $x_i = \rho^I \approx 0$, $y_i = \rho^A$ and $z_i = \rho^U$ and equations 4.12, 4.13 and 4.14 become:

$$q^A = 1 - \beta_2 \rho^I \tag{4.24}$$

$$q^U = 1 - \beta_2 \langle k \rangle R_\lambda(\langle k \rangle) \rho^I \tag{4.25}$$

$$v = 1 - \beta_1 \langle k \rangle R_\lambda(\langle k \rangle) \rho^I \tag{4.26}$$

plugging into 4.16:

$$\rho^I \left(-\frac{\mu_2}{\beta_2} + \rho^A + \rho^U \langle k \rangle R_\lambda(\langle k \rangle) \right) \tag{4.27}$$

Using the second equation of 4.8 with $\mu_1 = 0$:

$$\left(-\rho^A \beta_2 + \rho^U \langle k \rangle R_\lambda(\langle k \rangle) \beta_1 \right) \rho^I = 0 \tag{4.28}$$

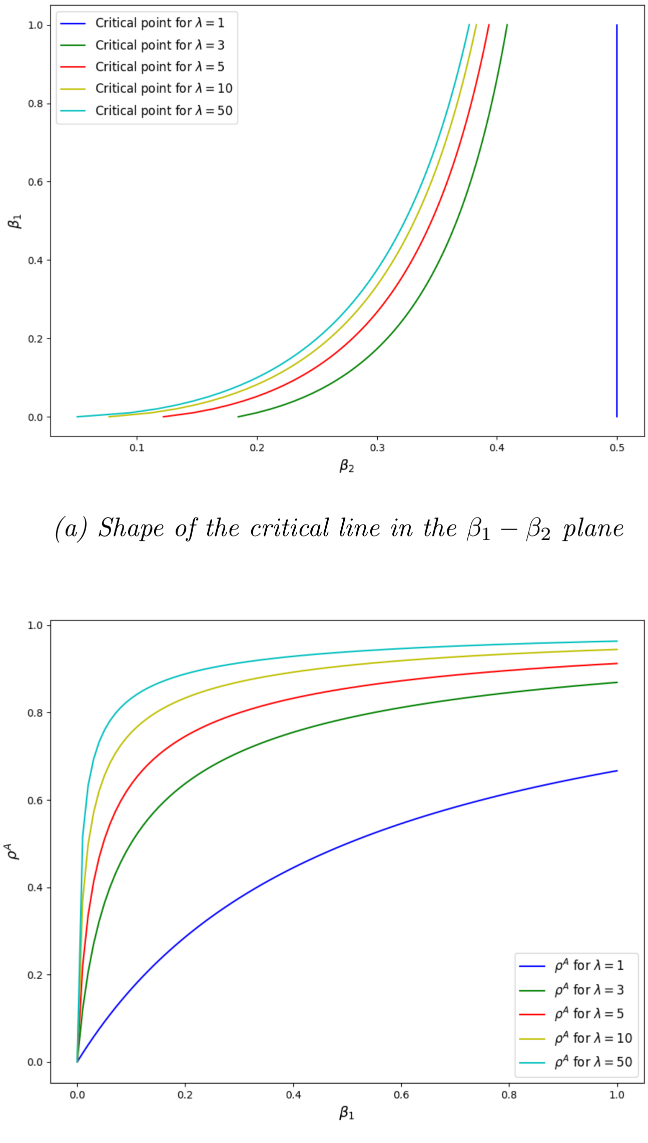
and writing $\rho^U = 1 - \rho^A - \rho^I$ we reach a second order equation whose solution with respect to β_2 is:

$$\beta_2 = \frac{\langle k \rangle R_\lambda(\langle k \rangle)\beta_1 + \mu_2 + \sqrt{(\langle k \rangle R_\lambda(\langle k \rangle)\beta_1 - \mu_2)^2 + 4(\langle k \rangle R_\lambda(\langle k \rangle))^2 \mu_2 \beta_1}}{2\langle k \rangle R_\lambda(\langle k \rangle)} \tag{4.29}$$

In this case there are also explicit expressions for ρ^A and ρ^U at the threshold:

$$\rho^A = \frac{\langle k \rangle R_\lambda(\langle k \rangle)}{\beta_2 + \beta_1 \langle k \rangle R_\lambda(\langle k \rangle)} \tag{4.30}$$

$$\rho^U = \frac{\beta_2}{\beta_2 + \beta_1 \langle k \rangle R_\lambda(\langle k \rangle)} \tag{4.31}$$



(a) Shape of the critical line in the $\beta_1 - \beta_2$ plane

(b) Value of ρ^A at the critical point in function of β_1

Figure 4.1: Critical line and critical solution for an homogeneous network changing with the initial number of contacts λ. $\langle k \rangle = 10$

4.2.2 Case $\mu_1 \neq 0$: individuals can forget

In the case in which aware individuals can forget the information, that is when $\mu_1 \neq 0$, at the threshold also y_i is approximately zero $\forall i$ since the aware individuals eventually forget about the disease and they don't get informed anymore as long as also $x_i \approx 0$. This means that the population is mostly unaware ($z_i \approx 1 \qquad \forall i$). Then:

$$H_{ij} = z_i (R^U)_{ij} \approx (R^U)_{ij} \tag{4.32}$$

$$(\beta_2)_c = \frac{\mu_2}{\Lambda_{max}(R^U)} \tag{4.33}$$

This says that the threshold is the same as in the classical SIS contact-based model (i.e. when $\beta_1 = 0$). So the spreading of the information doesn't give any advantage in term of the critical point, in other words: as hard the individuals try to avoid the infection decreasing their contacts, the onset of the disease is given by the same parameters β_2 and μ_2 that were found in the case $\beta_1 = 0$. The advantage will be to decrease the entity of the infection, so ρ is decreasing with β_1 given the same parameters, but this will be seen in section 5.2.

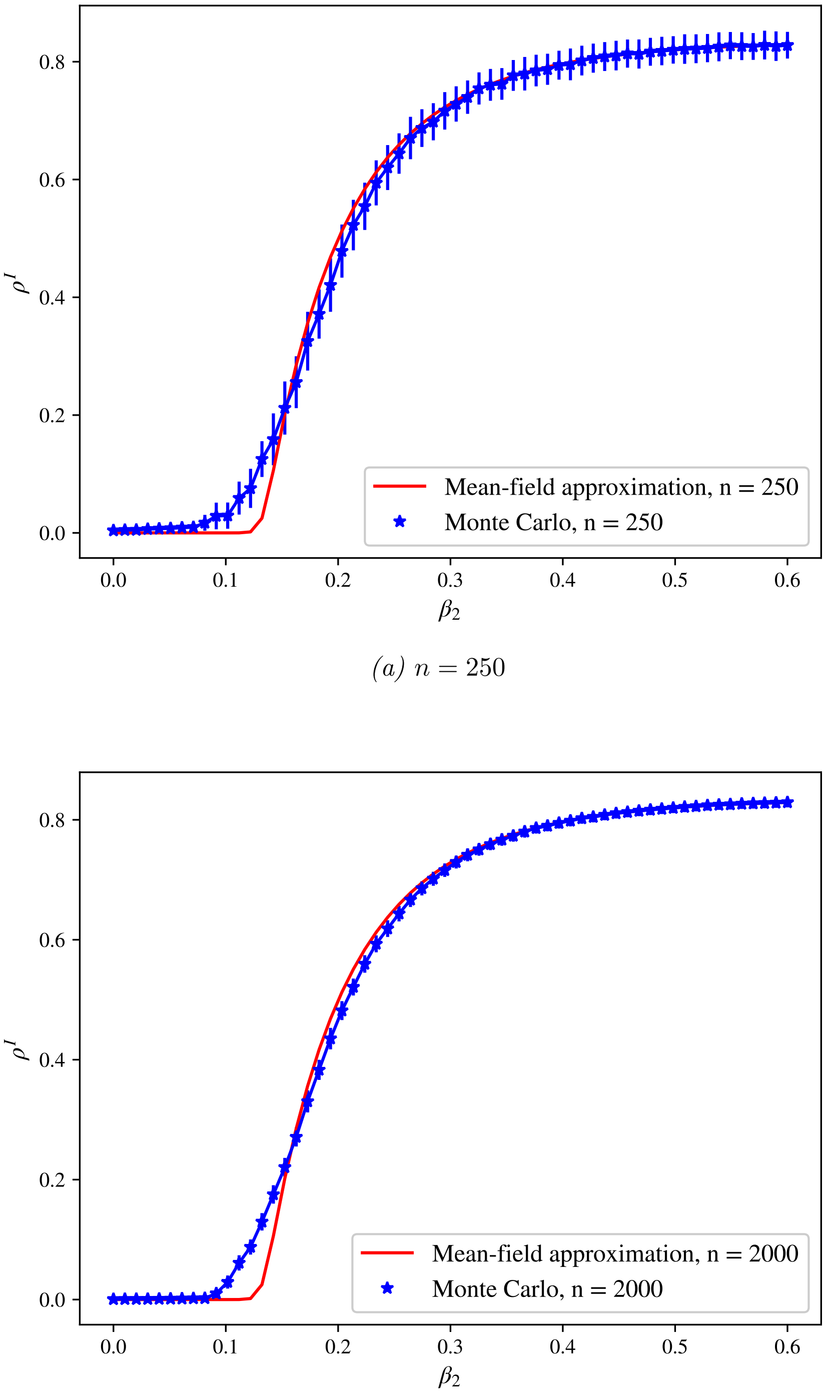
Chapter 5

Numerical results

5.1 Monte Carlo simulation

In order to validate the model, some pure stochastic simulations have been done and then compared to the Mean-field solution of equations 4.8. First an initial percentage of infected nodes is randomly selected, then the stochastic simulation runs, following the same logic of equations 4.7. The stochastic simulation can have random oscillations and reach the absorbing state (where there aren't infected individuals anymore) even if the parameters are above the threshold; in order to avoid this it's better to average the density of infected individuals over many successful simulations. However this method is really wasteful in terms of computational cost, since the survival runs are rare close to the critical point. So it has been used the *quasistationary method* (QS) [10] based on the idea of constraining the system in an active state. In order to do so a list of M active configurations is stored during the simulation and as soon as the system tries to visit the absorbing state, where the number of infected is equal to zero, then the present configuration is substituted with a random one chosen from these M configurations. The density of infected is then computed within an averaging time t_a after a relaxation time t_r.

In Figure 5.1 it can be seen that the agreement between steady state solutions given by Monte Carlo method and Mean-field equations gets better increasing the system size and also the dispersion decreases. But the mismatch between the two solutions is still present for two reasons: the first is due to correlation dynamics that are neglected in the mean-field approximation, the second is because of the small system size used in the simulations. Let's try to explain these problems better: Monte Carlo solution matches with the Mean-field solution only if the recovery rate is not to close to 1. This problem rises because having a recovery rate equal to one $\mu = 1$ means that at each time step all the infected individuals, after infecting others, recover with probability 1. So the systems follows really fast time scales and the mean field approximation is less correct, because inside a time step each node depends strongly on the connections between its neighbors (given by custering coefficient), breaking the condition $\mathbb{E}[X_i(t)X_j(t)] = \mathbb{E}[X_{j_1}(t)]\mathbb{E}[X_{j_2}(t)]$, where j_1, j_2 are neighbors of i. If the time step represents 1 day and the disease is influenza, imposing $\mu = 1$ is a really strong condition since it means that an infected person can recover in just one day. So taking lower values for μ, in a range $[0, 0.5]$ makes the mean-field model more accurate and also reasonable. In order to have a better approximation it's necessary to change the deterministic model and use another approximation; for example it has been shown [10] that a pair-wise approximation [12] gives more precise solutions, since it takes into account the correlations between neighbors. The second reason is that for bigger system sizes the shape of the curve becomes sharper, as it has been shown in [17, 8], but for practical reasons in this framework it's impossible to reach such dimensions.



(a) $n = 250$

(b) $n = 2000$

Figure 5.1: Comparison between Monte Carlo QS method and Mean-field equation of the steady state density of infected. Vertical bars define the dispersion. Random regular graphs $\langle k \rangle = 6$ *and parameters* $\beta_1 = 0.2$, $\mu_2 = 0.2$, $\mu_1 = 0$, $\lambda = 50$

5.2 Phase diagrams

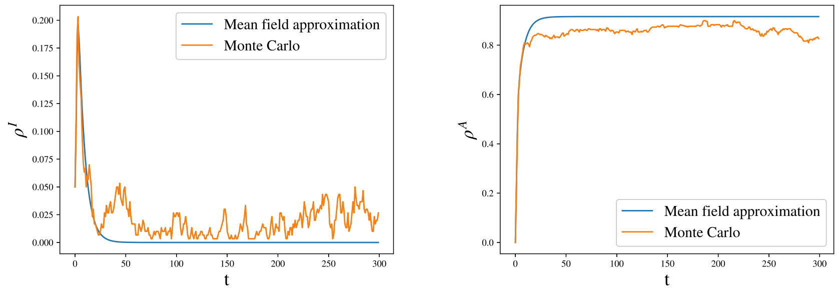
In order to see the effects of the the intensity of the information rate on the system, first the dynamic equations 4.8 are iterated and then the steady state solutions of the densities (so after the relaxation time is passed) are taken and plotted in the phase diagram, varying the value of β_1. First let's define the densities of infected, aware (or *awareness*) and unaware (or *ignorance*):

$$\rho^I = \frac{\sum_j^N x_j}{N} \tag{5.1}$$

$$\rho^A = \frac{\sum_j^N y_j}{N} \tag{5.2}$$

$$\rho^U = \frac{\sum_j^N z_j}{N} \tag{5.3}$$

In Figure 5.2 is shown the dynamics of the density of infected and aware individuals when the information rate is high ($\beta_1 = 0.7$) and it's enough to make the infection die out after few time steps. On the other hand in Figure 5.3 β_1 is lower and the infection is still present at the steady state. The blue and orange curve just define the difference between deterministic and stochastic equation along the dynamics.

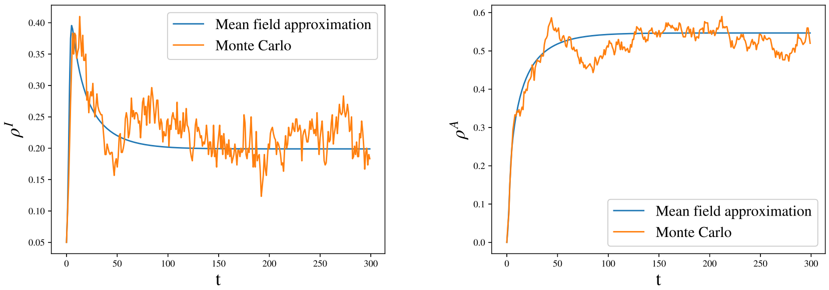


(a) ρ^I

(b) ρ^A

Figure 5.2: Dynamics of the densities in the absorbing phase for Erdös-Rényi graphs (randomly correlated) with $\langle k \rangle = 8$ *and parameters* $\beta_1 = 0.7$, $\beta_2 = 0.2$, $\mu_2 = 0.5$, $\mu_1 = 0$, $n = 300$, $\lambda = 50$

From Figure 5.4a it can be seen as (in the case of $\mu_1 = 0$) the information rate shifts the threshold: this means that in order to have an onset of the



(a) ρ^I *(b)* ρ^A

Figure 5.3: Dynamics of the densities in the active phase for Erdös-Rényi graphs (randomly correlated) with $\langle k \rangle = 8$ *and parameters* $\beta_1 = 0.1$, $\beta_2 = 0.2$, $\mu_2 = 0.5$, $\mu_1 = 0$, $n = 300$, $\lambda = 50$

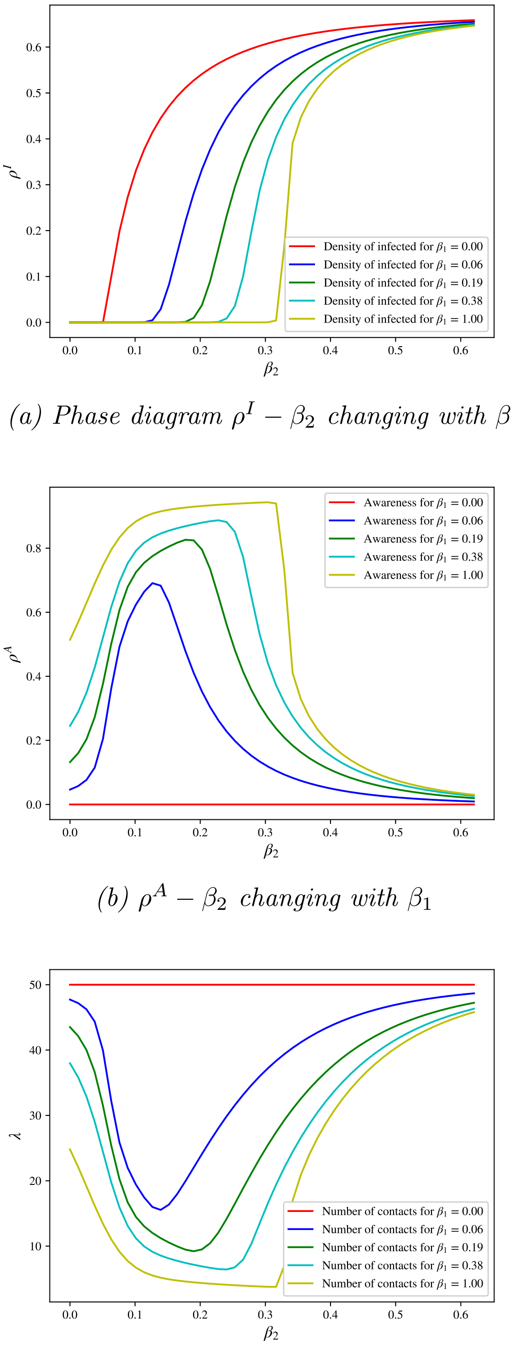
epidemic, it's necessary to have a more "aggressive" disease (larger β_2) as β_1 increases. While for large values of β_2 the densities converge at the same value, meaning that if the disease is strong enough, the spreading of the information has no effects. Figure 5.4b shows that the maximum of awareness corresponds to the critical point after which it starts to decrease up to zero. From 5.4c it can be seen as the threshold corresponds to the lower number of contacts. The average number of contacts is just the weighted average of the contacts made by susceptible individuals:

$$\langle \lambda \rangle = \frac{\lambda \rho^U + \rho^A}{\rho^U + \rho^A} \tag{5.4}$$

As long as the information rate is increased, the average number of contacts at the threshold decreases. And it's even more reduced when the infection rate approaches the critical point, since the two process are correlated and when the number of infected grows, these infected nodes also pass the information to some of their neighbors making the awareness grow as well. When β_1 approaches 1, every contacted node in the information layer receives successfully the information and then changes its contacts to 1, making the average number of contacts very low. At the threshold the average number of contacts $\langle \lambda \rangle$ gets closer and closer to 1, but it will never become 1 since the recovered nodes will still have λ contacts. Therefore after a while is no more possible to decrease the average number of contacts further, since the most of the nodes will be aware, and the growth of the infection rate can only have the effect to increase the spreading of the infection, but not the information, since the latter is at its maximum value.

In Figure 5.5 are plotted the heat maps of the phase diagrams of ρ^I, ρ^A and ρ^U in function of the 2 spreading rates: the blue area in 5.5a is the safe region in which the disease is not present. In these plots is even more evident how the information has effect mostly on lower values of the infection rate.

In the case in which $\mu_1 \neq 0$ the behavior is completely different, as shown in Figures 5.6 and 5.7: the critical point is no more shifted, as it has been said in the previous section, and the only effect that the information has is to decrease the entity of the infection, but not to nullify it. From now on the research will be focused only on the case where $\mu_1 = 0$, which is more interesting since the threshold is not trivial and it's also more reasonable since in the case of a real epidemic, people don't drop the information after a while, but just in the case when they are told that the infection is over.

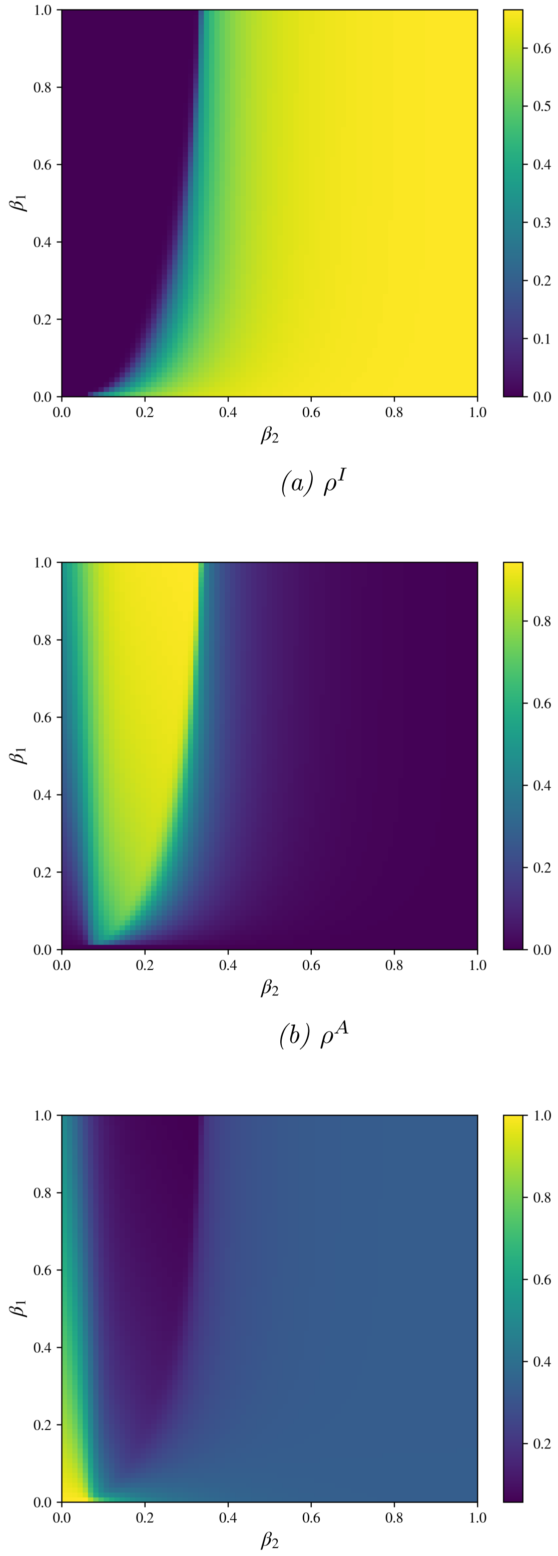


(a) Phase diagram $\rho^I - \beta_2$ changing with β_1

(b) $\rho^A - \beta_2$ changing with β_1

(c) Average number of contacts changing with β_1

Figure 5.4: Effects of changing the information rate on the system for Erdös-Rényi graphs (randomly correlated) with $\langle k \rangle = 8$ and parameters $\mu_2 = 0.5$, $\mu_1 = 0$, $n = 300$, $\lambda = 50$

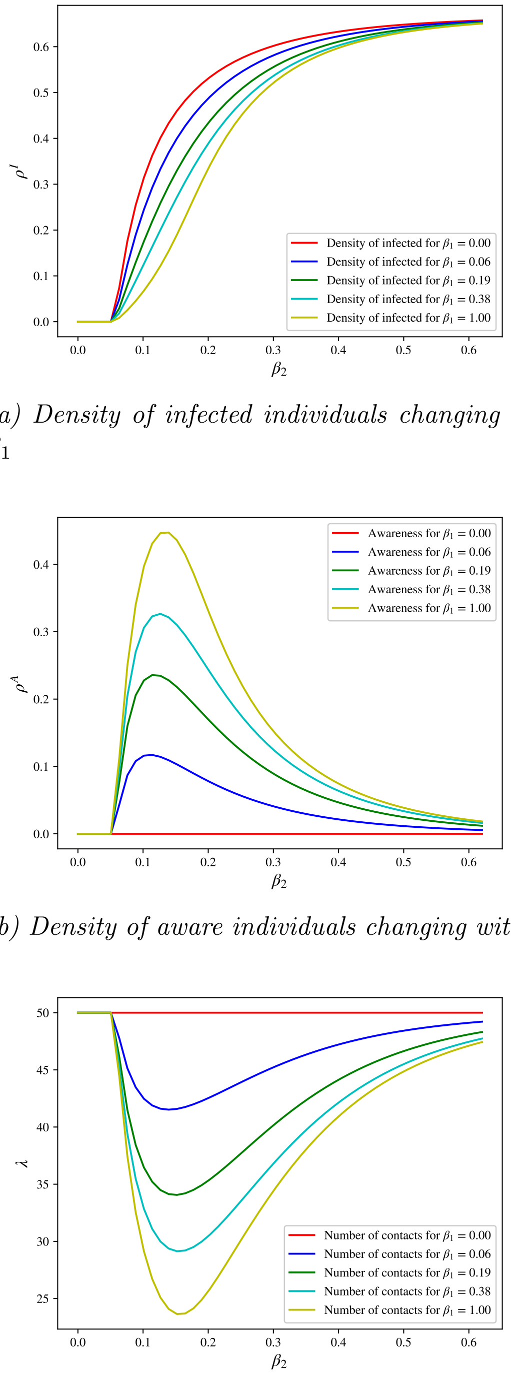


(a) ρ^I

(b) ρ^A

(c) ρ^U

Figure 5.5: Effects of changing the two spreading rates on the densities for an Erdös-Rényi graph with $\langle k \rangle = 8$ and parameters $\mu_2 = 0.5$, $\mu_1 = 0$, $n = 300$, $\lambda = 50$

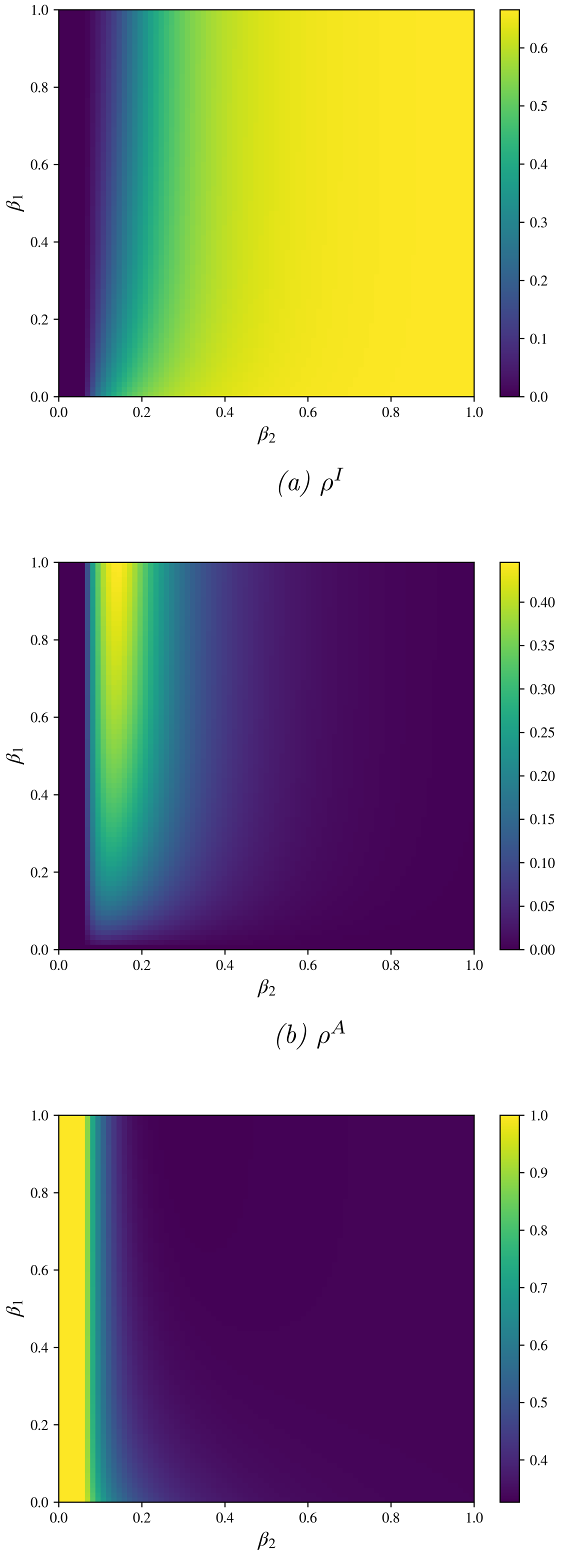


(a) Density of infected individuals changing with β_1

(b) Density of aware individuals changing with β_1

(c) Average number of contacts changing with β_1

Figure 5.6: Effects of changing the information rate on the system for Erdös-Rényi graphs (randomly correlated) with $\langle k \rangle = 8$ and parameters $\mu_2 = 0.5$, $\mu_1 = 0.5$, $n = 300$, $\lambda = 50$



(a) ρ^I

(b) ρ^A

(c) ρ^U

Figure 5.7: Effects of changing the two spreading rates on the densities for an Erdös-Rényi graph with $\langle k \rangle = 8$ *and parameters* $\mu_2 = 0.5$*,* $\mu_1 = 0.5$*,* $n = 300$*,* $\lambda = 50$

5.3 Degree densities

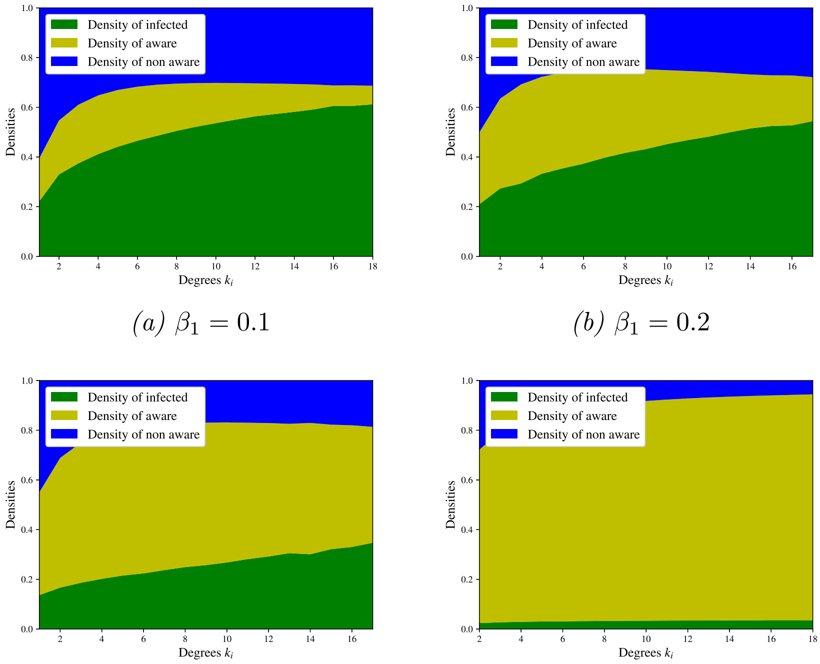
So far only the global properties of the systems have been studied, such as the global densities, but it can be interesting to see where the disease and the awareness concentrate more inside the network. Let's take then two identical networks for both layers and let's see whether the hubs are mostly occupied by the disease or by the information at the steady state and with conditions not too far from the critical point. The degree densities are defined as follows:

$$\rho_k^I = \frac{\sum_{i \in V_k} x_i}{N_k} \tag{5.5}$$

$$\rho_k^A = \frac{\sum_{i \in V_k} y_i}{N_k} \tag{5.6}$$

$$\rho_k^U = \frac{\sum_{i \in V_k} z_i}{N_k} \tag{5.7}$$

where V_k is the set of nodes with degree equal to k and N_k is its size. From Figure 5.8 it can be seen how the partition of the degree-densities changes with the degree, varying the information rate. Looking at the green and yellow area it can be noticed that the the hubs show always a "disease prevalence", in the sense that, despite the increasing value of β_1, most of the disease will concentrate on the highest degree nodes, while the awareness is more distributed in lower degree nodes, until β_1 is high enough to make the information prevail. This is even more evident for a Barabasi-Albert network, Figure 5.9, that is a particular power-law network with exponent equal to 3. This happens because it was supposed at the beginning that only the infected nodes spread the information, therefore when the infection starts, it's more likely for the spreading process to concentrate on the hubs; so in the case in which $\beta_1 < \beta_2$, the hubs are more likely to be infected and only at the next time step they can communicate their status to the remained nodes, that are lower degree nodes. Since we assumed for now that individuals don't forget ($\mu_1 = 0$), the aware nodes will keep just 1 contact until they get infected (but it's really unlikely for them to contract the infection), this means that the infection will keep bouncing between the higher degree nodes, followed by the information, until the steady state. It's better to emphasize that this qualitative explanation only works when β_2 is close to the critical point and such that there exist a β_1 such that it makes the infection die out. Because it has already been said that for high values of β_2 the information spreading has no effects on the systems.



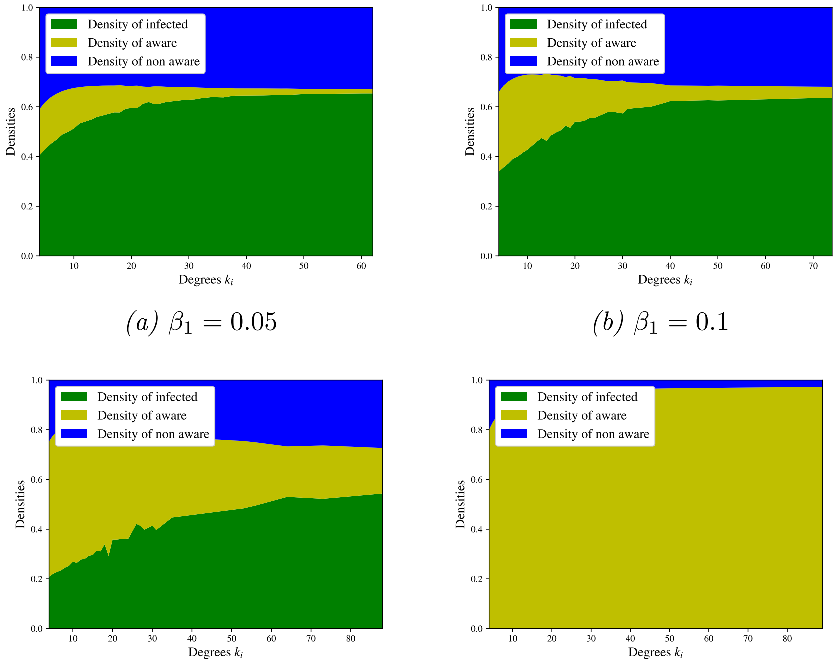
(a) $\beta_1 = 0.1$

(b) $\beta_1 = 0.2$

(c) $\beta_1 = 0.3$

(d) $\beta_1 = 0.4$

Figure 5.8: Degree densities for an Erdös-Rényi graph with $\langle k \rangle = 8$ *and parameters* $\mu_2 = 0.5$*,* $\mu_1 = 0$*,* $\beta_2 = 0.3$ $n = 500$*,* $\lambda = 50$



(a) $\beta_1 = 0.05$

(b) $\beta_1 = 0.1$

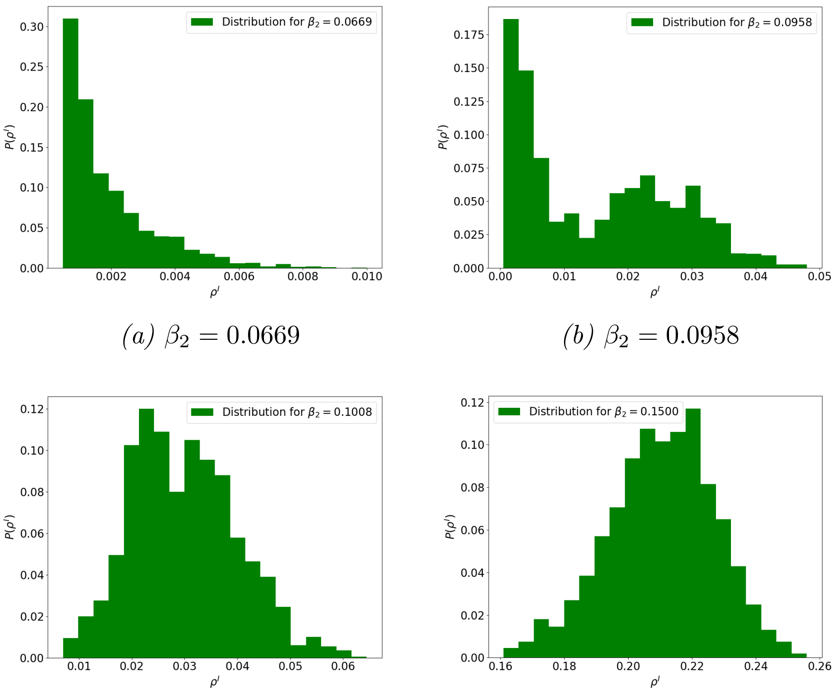
(c) $\beta_1 = 0.2$

(d) $\beta_1 = 0.3$

Figure 5.9: Degree densities for an Barabasi-Albert graph with $\langle k \rangle = 8$ *and parameters* $\mu_2 = 0.5$*,* $\mu_1 = 0$*,* $\beta_2 = 0.25$ $n = 500$*,* $\lambda = 50$

5.4 Numerical localization of the critical point

The position of the critical point is one of the most important feature when one studies the epidemic spreading processes since it clearly divides the endemic phase of the process with the absorbing phase, where the disease dies out after a while. There is then a plethora of studies about this that involves knowledge and techniques used mainly in statistical mechanics, thanks to the similarity that comes upon in these kind of processes. All the following methods are based on the statistics of pure stochastic simulations, that are performed with quasi-stationary Monte Carlo methods that have been described in 5.1. Moreover, from now on only pure homogeneous networks (Random Regular graphs) are used, since they are the only networks for which an analytic expression of the critical point has been found (see 4.29). To fix a reference we're going to use random regular graphs with $\langle k \rangle = 6$ for both layers (but different) and parameters $\beta_1 = 0.2$, $\mu_2 = 0.2$, $\mu_1 = 0$,



(a) $\beta_2 = 0.0669$

(b) $\beta_2 = 0.0958$

(c) $\beta_2 = 0.1008$

(d) $\beta_2 = 0.1500$

Figure 5.10: Distribution of the density of infected before and after the critical point

$\lambda = 50$, which critical point in the mean field approximation is $\beta_2 = 0.13$. However it's better to stress again that the analytic solution will differ a bit from the pure stochastic solution, as it has been explained in 5.1, but the methods that are used here can still find the effective critical point since they disregards approximations and they are only affected by the number of simulations where to average on.

In Figure 5.10 it can be seen how the distribution of the density of infected changes from the absorbing phase to the active phase: before the critical point the density has a peak in zero and decays up to little value of $\rho^I \approx 0.008$; on the other hand over the critical point it distributes like a gaussian whose standard deviation decreases with n. In this case the critical point is around the value $\beta_2 = 0.0958$ since from the picture it's evident how the distribution (and therefore the system) is passing through a transition.

5.4.1 Finite size scaling

Statistical mechanics shows how continuous (or second order) phase transitions are characterized by power-law behaviors of the thermodynamics variables with respect to the temperature, such as magnetization, susceptibility, latent heat etc. This means that while approaching the critical temperature T_c every thermodynamic function F behaves like $F(t) \sim |t|^{\alpha}$, with $|t| = (T - T_c)/T_c$. For example, taking the phase transition from ferromagnetic to paramagnetic, without external magnetic field ($H = 0$), it comes out that the magnetization m goes to zero with a power-law behavior (therefore the exponent must be positive) while the susceptibility χ and the correlation length ξ diverge at the critical temperature (then the exponents are negative) as:

$$m \sim |t|^{\alpha} \ , \ \alpha > 0 \tag{5.8}$$

$$\chi \sim |t|^{-\gamma} \ , \ \gamma > 0 \tag{5.9}$$

$$\xi \sim |t|^{-\nu} \ , \ \nu > 0 \tag{5.10}$$

This phenomena and the phase transition itself make sense only at the thermodynamic limit so when the size of the system approach infinity ($n \to \infty$), which is unfeasible for obvious reasons, but the study for finite dimensions in this case comes out to be useful to investigate the threshold.

The finite size scaling method [10, 16] allows to estimate numerically the critical point and the critical exponents studying the density for different system sizes. The main concept of this method is based on the hypothesis of Fisher and Barber [11, 16] which said that there must be only a characteristic length of the system describing the change of the thermodynamic singularities from the infinite case to the finite one. This means that the correlation length in finite system is assumed to be cut off to L when it reaches the system size, while in the case in which $\xi << L$ it's equal to the infinite system one, calling with L the system size in the general case. Plugging 5.10 into 5.8 we get:

$$m \sim L^{-\alpha/\nu} \tag{5.11}$$

which tells us that m decays as a power law also with respect to the system size at the critical point.

Therefore, in our case where the system size is defined by the network size n, the order parameter by ρ^I, the control parameter as β_2, this reasoning still holds and we have:

$$\rho^I \sim (\beta_2 - \beta_c)^\alpha \tag{5.12}$$

$$\xi \sim (\beta_2 - \beta_c)^{-\nu} \tag{5.13}$$

Putting them together using $\xi \sim n$

$$\rho^I \sim n^{-\alpha/\nu} \tag{5.14}$$

if the control parameter β_2 stays under the critical point β_c then $\rho^I \sim \frac{1}{n}$ since it's limited to only one infected node because of the QS method. While when $\beta_2 > \beta_c$ the density is constant with respect to the system size. The critical point is characterized by a power-law decay $\rho^I \sim n^x$, $x > 0$ [17]. Plotting in logarithmic scale ρ^I in function of the dimensions and for different control parameters β_2 it can be estimate where the critical point is: for example in Figure 5.11 we have that approximately until $\beta_2 = 0.08$ the system is in the absorbing phase, since the lines have slope close to 1, when $\beta_2 = 0.09$ instead the slope increases until it reaches zero (so when the system is in the ordered phase). Therefore we can conclude that the critical point is in between $0.08 < \beta_c < 0.10$. This critical point doesn't match with the analytic one from equation 4.29 for the reasons said in section 5.1.

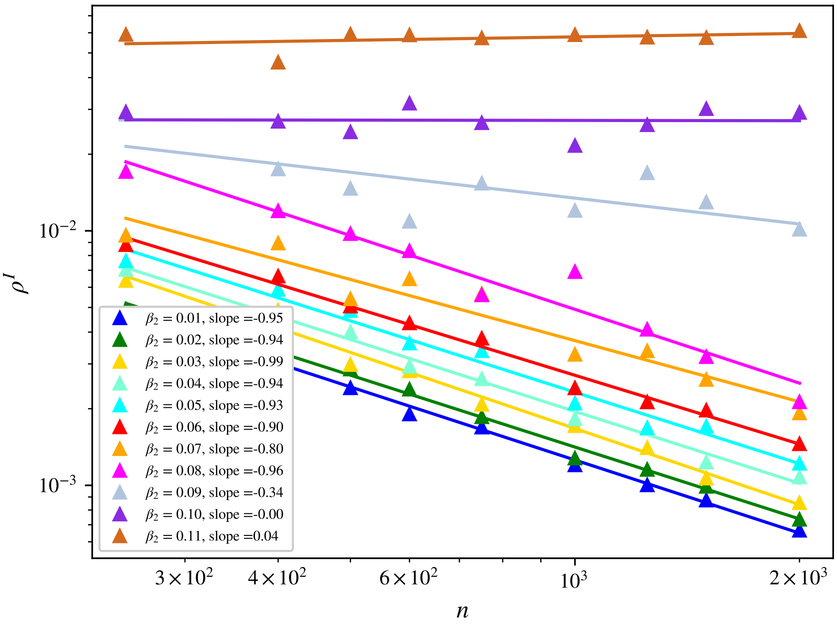


Figure 5.11: Random regular graphs with $\langle k \rangle = 6$ *and parameters* $\beta_1 = 0.2$, $\mu_2 = 0.2$, $\mu_1 = 0$, $\lambda = 50$

5.4.2 Susceptibility peak

The susceptibility is a measure of the oscillation of the density which in the finite size case it presents a peak at the critical point, since it's the point in which the biggest oscillations are found. The modified susceptibility is defined as follows [10]:

$$\chi = N\frac{\langle(\rho^I)^2\rangle - \langle\rho^I\rangle^2}{\langle\rho^I\rangle} = \frac{\langle n_I^2\rangle - \langle n_I\rangle^2}{\langle n_I\rangle} \tag{5.15}$$

with n_I the number of infected individuals. The average is taken among the last t_a values of the density of infected individuals coming from the QS Monte Carlo simulaton. It's been used equation 5.15 and not the classical expression for susceptibility $\chi = N\langle(\rho^I)^2\rangle - \langle\rho^I\rangle^2$ since it leads to clearer numerical results keeping the same scaling properties [10]. It can be verified in Figure 5.12 since the peak seems to coincide with the critical point. Moreover, if we plot the susceptibilities altogether we can see that the maximum of

χ increases with the system size, as it can be seen in Figure 5.14 and in order to see it clearer the susceptibility is zoomed in near the critical point (Figure 5.13b). Since the closer to the threshold the system is, the bigger the oscillations are, it has been necessary to pass it through an moving average filter, as it has been shown in Figure 5.13b. When we plot the values of these maximum in a logarithmic scale plot in function of the system size, they concentrate along a straight line, as predicted in the finite size scaling theory. In fact taking equation 5.10 and plugging it into 5.9 it comes out that

$$\chi \sim n^{\gamma/\nu} \tag{5.16}$$

The slope of the line is the value of the critical exponent. In this case then $\gamma/\nu = 0.45$

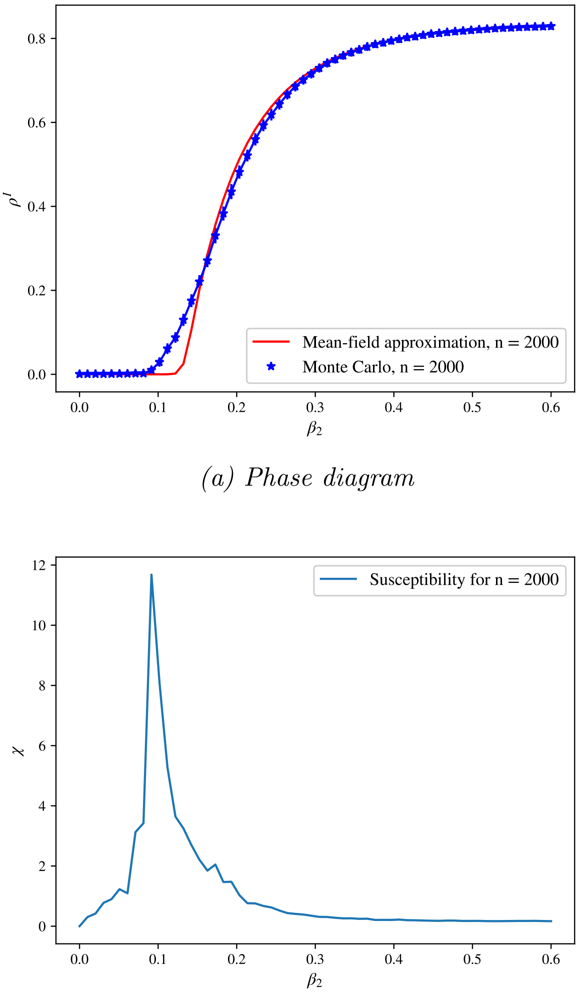
The position of the maximum of the susceptibility is actually shifted from the real critical point (for infinite systems $\beta_c(\infty)$) and it places itself in a *pseudo critical point* $\beta_c(n)$. It can be expected that [16], for sufficiently large system ($L \to \infty$):

$$(\beta_c(n) - \beta_c(\infty)) \sim L^{-\lambda'} \tag{5.17}$$

where λ' is the so called *shift exponent*. Moreover, the more the dimension is small, the broader the curves are, and this can be described defining a *rounding parameter* $\beta_2^\star$ such that if $|\beta_2 - \beta_c| \geq |\beta_2^\star - \beta_c|$ than the finite size susceptibility is equal to the one from the infinite case $\chi(n) = \chi(\infty)$. Equivalentely to the shift exponent one may define the *rounding exponent*, for sufficiently large system ($L \to \infty$), as:

$$(\beta_2^\star - \beta_c(\infty)) \sim L^{-\theta} \tag{5.18}$$

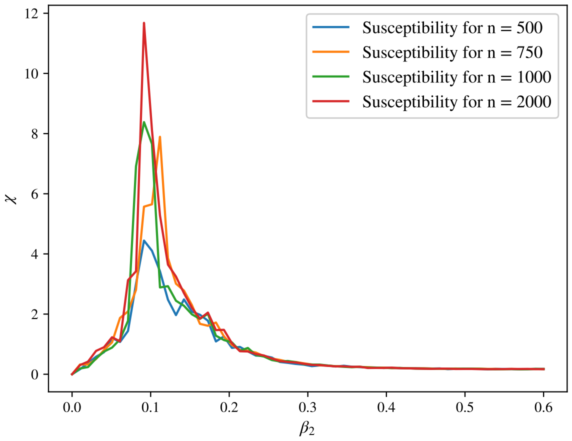
If we use the assumption proposed by Fisher and Barber [11] described in 5.4.1 ($\xi = n$) it comes out that $\lambda' = \theta = 1/\nu$ since $\xi \sim (\beta_2 - \beta_c)^{-\nu}$ close to the critical point.



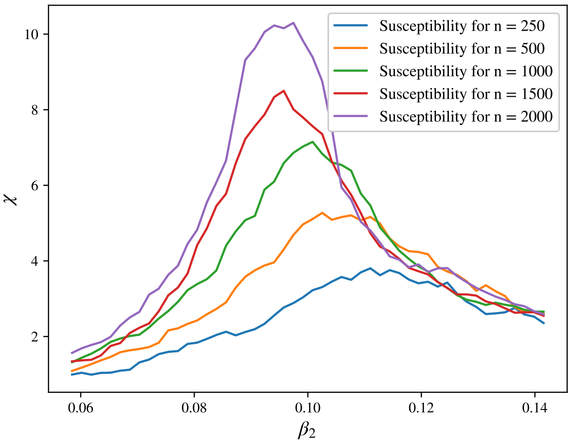
(a) Phase diagram

(b) Susceptibility

Figure 5.12: Showing the corrispondency between the critical point and the peak of susceptibility



(a) Susceptibility for different system sizes in function of β_2



(b) Susceptibility close to the critical point for different system sizes in function of β_2

Figure 5.13: Random regular graphs with $\langle k \rangle = 6$ *and parameters* $\beta_1 = 0.2$, $\mu_2 = 0.2$, $\mu_1 = 0$, $\lambda = 50$

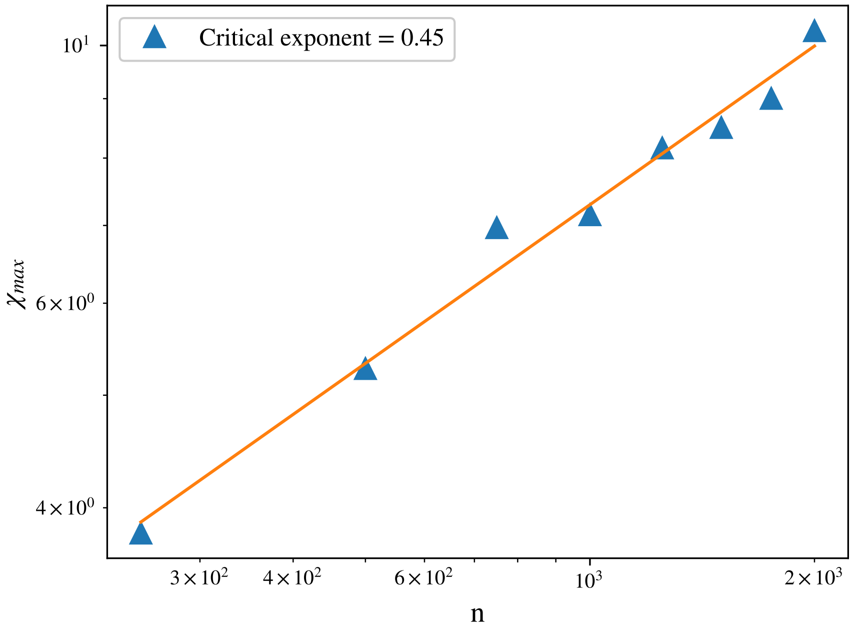


Figure 5.14: Maximum of the susceptibility χ changing the system size n. Random regular graphs with $\langle k\rangle = 6$ and parameters $\beta_1 = 0.2$, $\mu_2 = 0.2$, $\mu_1 = 0$, $\lambda = 50$

5.4.3 Lifespan method

Another way to distinguish the absorbing and the epidemic phase relies on the average lifetime of the infection over the simulations, in other words, the lifespan of the infection [18]. Theoretically speaking, just above the epidemic threshold, the lifespan of the infection is infinite while below it's finite and then it's possible to compute it averaging the time it takes to reach zero along the simulations. With the quasi-stationary Monte Carlo method it's defined in a slightly different. Calling $P(n_I)$ the probability of having n_I infected nodes, computed along a given time t_a in a single simulation, the lifespan will be then:

$$\langle T\rangle = \frac{1}{P(1)} \tag{5.19}$$

that is the rate at which configurations with a single infected nodes appear.

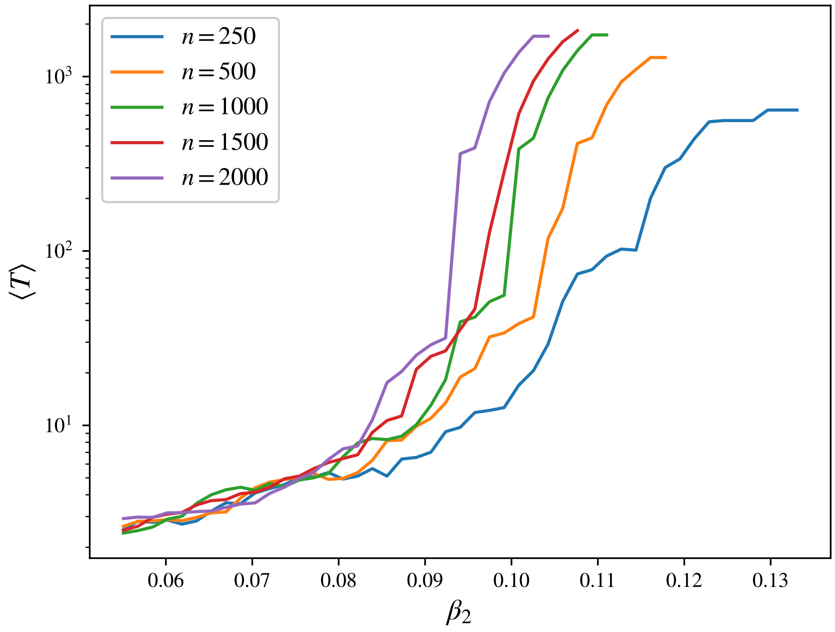


Figure 5.15: Random regular graphs with $\langle k \rangle = 6$ *and parameters* $\beta_1 = 0.2$, $\mu_2 = 0.2$, $\mu_1 = 0$, $\lambda = 50$

In the practical cases the endemic phase won't be characterized by infinite lifespan, but it's cut-off a the final simulation time, which is going to be larger for bigger network size as can be seen in Figure 5.15; when the ending time of the simulation is achieved without reaching $n^I = 1$, the lifespan is not considered. In Figure 5.15 then the critical point is given by the ending point of the curves. Since close to the critical point the oscillations are huge, firstly the lifepan is passed through an average filter to clarify better the results. The critical point is estimated better for larger running time for the simulations, which is here limited to $t_{max} = 3000$ with a relaxation time of 1000 for the biggest sizes, that leads to $t_a = 2000$.

Conclusions

This thesis proposed a new idea to face the human reaction to the infection, based on the activity of the individuals: it took several months of work, because from intuition to modeling there is a really big step. The main result is that the spreading of the information have a clear impact on the infection when people don't forget the information ($\mu_1 = 0$) since it shifts the critical point to an higher value and the density of infected at the steady state is always lower or equal to the ordinary case in which information is not present. While, when individuals lose the information, the threshold stays at the same point such as in a fully reactive process and the only effect of being aware is to decrease the value of the density of infected. The analytical equations built from the mean-field approximation have been always compared to the Monte Carlo simulation so that it was possible to check the consistency of the solutions. The main problem in this case is the imperfection of the mean-field equations close to the critical point where the approximation leads to a non exact solution. Moreover, also Monte Carlo would need to be taken to heavier simulations since here, for numerical limitations of a simple laptop, it's restricted to the order of 10^3 time steps and up to a maximum of 2000 nodes per layer. In order to fix this it could be necessary to do stronger Monte Carlo simulations and to use finer models with other approximations instead of the mean-field one, but this would mean to rewrite everything from the beginning. For future perspectives it could be also interesting to couple this model to similar other ones, for example the one made by C. Granell et al. [14] which assumes that the individuals react trying to prevent the infection taking medicines (reducing β_2), which is another kind of hypothesis. Furthermore here the approach is only probability-modeling driven, while one could see the problem from the data and statistical point of view, that would mean to first study data from both real networks and social network and find a pattern on how the people aware of the infection pass the information and react to it. This needs to take into consideration to be able to have access to data related to health and social networks, which is not so straightforward. However this thesis could be a really good launchpad where to begin if someone wanted to get involved in this transverse and complex world of the epidemic and social dynamics.

Acknowledgments

This thesis has been possible firstly thanks to the people who directly helped me with this work: prof. Yamir Moreno who welcomed me in his lab and who has been a wise advisor at any time, prof. Amos Maritan who has always been a reference and an inspiration during the whole master period, and finally Dr. Sandro Meloni who really gave me practical suggestions and help to go on when I was stuck with some problems.

I would like to express my deep gratitude to my family who have always had faith in me and without which I couldn't have all the possibilities that I have right now. I think I am one of the luckiest people on earth because of you.

I would like to thank the friends from Piovene and neighborhoods: once I was asked what I like most of my friends and I replied "the ability of being smart and stupid at the same time", I think you all are like this: all of you are really witty but at the same time you are able to live in lightness. I love this.

Thanks to all the people I've met in Padova, from my engineers friends who always made the Wednesday nights fun as never, to my flatmates in Arcella that I consider my second home by now, with siblings coming and going.

Thanks also to the beautiful people I met in my experiences away, from Tu/e mates, Cederlaan community and fellow travelers around Netherlands, to my colleagues at BIFI in Zaragoza.

A big thanks goes to my boyfriend Giucci, whose patience has been really effective in many periods to counterbalance my stress. You are the best

listener I have ever met, the best critical thinker, but mostly, you are a wonderful person. Thank you darling.

Even if my master degree choice departed a bit from its activities, a huge thanks goes also to LEDS, which is the best thing that happened in my university life and where, for the first time in my life, I had a "feeling of belonging". I strongly believe this association has the will and the possibility to change this world for the better.

Thanks finally to Python and its open source libraries and to LaTeX which made the work easier.

Bibliography

[1] Daniel Bernoulli. Réflexions sur les avantages de l'inoculation. *Mém. Paris*, pages 439–482, 1758.

[2] Daniel Bernoulli. Essai d'une nouvelle analyse de la mortalité causée par la petite vérole, et des avantages de l'inoculation pour la prévenir. *Histoire de l'Acad., Roy. Sci.(Paris) avec Mem*, pages 1–45, 1760.

[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4):175 – 308, 2006.

[4] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

[5] Fred Brauer. Mathematical epidemiology: Past, present, and future. *Infectious Disease Modelling*, 2(2):113 – 127, 2017.

[6] Carlos Castillo. *Effective Web Crawling*. PhD thesis, University of Chile, 2004.

[7] Fan Chung, Linyuan Lu, and Van Vu. Spectra of random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 100(11):6313–6318, 2003.

[8] Guilherme Ferraz de Arruda, Francisco A Rodrigues, and Yamir Moreno. Fundamentals of spreading processes in single and multilayer complex networks. *arXiv preprint arXiv:1804.08777*, 2018.

[9] Guilherme Ferraz de Arruda, Francisco Aparecido Rodrigues, Pablo Martín Rodríguez, Emanuele Cozzo, and Yamir Moreno. A general markov chain approach for disease and rumour spreading in complex networks. *Journal of Complex Networks*, 2017.

[10] Silvio C Ferreira, Claudio Castellano, and Romualdo Pastor-Satorras. Epidemic thresholds of the susceptible-infected-susceptible model on networks: A comparison of numerical and theoretical results. *Physical Review E*, 86(4):041125, 2012.

[11] Michael E. Fisher and Michael N. Barber. Scaling theory for finite-size effects in the critical region. *Phys. Rev. Lett.*, 28:1516–1519, Jun 1972.

[12] James P Gleeson. High-accuracy approximation of binary-state dynamics on networks. *Physical Review Letters*, 107(6):068701, 2011.

[13] Sergio Gómez, Alexandre Arenas, J Borge-Holthoefer, Sandro Meloni, and Yamir Moreno. Discrete-time markov chain approach to contact-based disease spreading in complex networks. *EPL (Europhysics Letters)*, 89(3):38009, 2010.

[14] Clara Granell, Sergio Gómez, and Alex Arenas. Dynamical interplay between awareness and epidemic spreading in multiplex networks. *Physical review letters*, 111(12):128701, 2013.

[15] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of complex networks*, 2(3):203–271, 2014.

[16] S. Lubeck M. Henkel, H. Hinrichsen. *Non-Equilibrium Phase Transitions, Volume 1: Absorbing Phase Transitions.* Springer, 2008.

[17] J. Marro and R. Dickman. *Nonequilibrium phase transitions in lattice models.* Cambridge University Press, Cambridge, 1999.

[18] Angélica S Mata and Silvio C Ferreira. Multiple transitions of the susceptible-infected-susceptible epidemic model on complex networks. *Physical Review E*, 91(1):012816, 2015.

[19] Sandro Meloni, Nicola Perra, Alex Arenas, Sergio Gómez, Yamir Moreno, and Alessandro Vespignani. Modeling human mobility responses to the large-scale spreading of infectious diseases. *Scientific reports*, 1:62, 2011.

[20] Carl D Meyer. *Matrix analysis and applied linear algebra*, volume 71. Siam, 2000.

[21] John Nopsa, Gregory Daglish, David W. Hagstrum, John Leslie, Thomas Phillips, Caterina Scoglio, Sara Thomas-Sharma, Gimme H. Walter, and Karen A. Garrett. Ecological networks in stored grain: Key postharvest nodes for emerging pests, pathogens, and mycotoxins. 65, 10 2015.

[22] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *Reviews of modern physics*, 87(3):925, 2015.

[23] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *Reviews of modern physics*, 87(3):925, 2015.

[24] Ronald Ross. The prevention of malaria, with addendum on the theory of happenings. *Murray, London*, 1911.

[25] A. G. McKendrick W. O. Kermack. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 115(772):700–721, 1927.

[26] A. G. McKendrick W. O. Kermack. Contributions to the mathematical theory of epidemics. ii. —the problem of endemicity. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 138(834):55–83, 1932.

[27] A. G. McKendrick W. O. Kermack. Contributions to the mathematical theory of epidemics. iii.—further studies of the problem of endemicity.

Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 141(843):94–122, 1933.

Appendices

Appendix A

Perron-Frobenius theorem

$\mathbf{A} \in \mathbb{R}^{m\times n}$ is said to be a nonnegative matrix whenever each $a_{ij} \geq 0$ and this i denoted as $\mathbf{A} \geq 0$

Perron-Frobenius theorem for nonnegative and irreducible matrices states that [20]:
If $\mathbf{A}_{n\times n}$ is irreducible the following statements are true.

- $r = \rho(\mathbf{A}) \in \sigma(\mathbf{A})$ and $r > 0$
- r has algebraic multiplicity equal to 1
- There exists an eigenvector $\mathbf{x} > 0$ such that $\mathbf{A}\mathbf{x} = r\mathbf{x}$
- The unique vector defined by

  $$\mathbf{A}\mathbf{p} = r\mathbf{p}, \qquad \mathbf{p} > 0, \qquad \text{and} \qquad ||p||_1 = 1$$

  is called the *Perron vector*.

Appendix B

Main codes

ML_functions_2L.py

```
import numpy as np
import networkx as nx

# Useful functions used in mean-field equations

def contact_mat(A, n, l, k):
    #---------
    # Creates the contact matrix of 1 layer given:
    # A = adjacency matrix
    # n = number of nodes
    # l = number of contacts that an infected does
    # k = vector of the node degrees (number of neighbors)
    #---------
    k = k.astype(float)
    RR = A/k[:,None]
    R = np.empty([n, n])
    R = 1-np.power((np.ones((n,n)) - RR),l)
    return R
```

```
def iter_dynamic_eq_lambda(p, w, x, beta2, beta1, R1,
    A2,l1,l2, k1, k2, mu2, mu1, n):
     # Iteration of the dynamic equations
     # --- INPUTS ---
     # - p,w,x = probability of respectevely being
          infected, aware and unaware at time t. n-dim
          vectors
     # - beta2, beta1 = infection rate, information rate
          .
     # - R1 = contact matrix of information layer
     # - A2 = adjacency matrix of infection layer
     # - l1, l2 = lambda, so number of contacts in layer
          1 and 2 (for unaware people).
     # - mu2, mu1 = recovery rate, forget rate
     # - n = number of nodes
     # --- OUTPUTS ---
     # p_new, w_new, x_new = probability of respectevely
          being infected, aware and unaware at time t+1
     q_a = np.ones(n)
     q_u = np.ones(n)
     v = np.ones(n)
     R2_u = contact_mat(A2, n, l2, k2)
     R2_a = A2/k2[:,None]

     for j in range(n):    # perform productory
          patrick2_a = np.array(R2_a[:,j]).flatten()
          patrick2_u = np.array(R2_u[:,j]).flatten()
          patrick1 = np.array(R1[:,j]).flatten()
          q_a = q_a*(1-beta2*patrick2_a*p[j])
          q_u = q_u*(1-beta2*patrick2_u*p[j])
          v = v*(1-beta1*patrick1*p[j])

     p_new = p*(1.-mu2)+w*(1.-q_a)+x*(1.-q_u)
    w_new = w*q_a*(1.-mu1) + x*(1.-v)*q_u
    x_new = x*v*q_u + w*q_a*mu1 + p*mu2
    return p_new, w_new, x_new
```

---

funct_MC_contacts3.py

```
import numpy as np
import matplotlib.pyplot as plt
import networkx as nx
import scipy as sp
import scipy.optimize
import math
from Create_RR_2L import create_RR_2L
import random
from ML_functions_2L import contact_mat

# Monte Carlo dynamics functions using the contact
    matrix R instead of doing a pure MC

def MC_time_step3(infected,aware,ignorants,G1,G2,R1,R2,
    A2,k1,k2,beta1,beta2,mu2,mu1,l1,l2, n):
    # Dynamics of the Monte Carlo simulation in 1 time
        step

    # Indices of infected nodes
    inf = np.where(infected==1)
    inf = np.array(inf).flatten()
    # Indices of aware nodes
    aw = np.where(aware == 1)
    aw = np.array(aw).flatten()
    # indices of ignorant nodes
    ign = np.where(ignorants == 1)
    ign = np.array(ign).flatten()
```

```
# New states for t+1
infected_new = np.zeros(n)
ignorants_new =  np.zeros(n)
aware_new =  np.zeros(n)

# ------------------------------------------
# Dynamics of infected individuals
# The infected nodes can recover and become unaware
Ber_recover = np.random.binomial(n=1,p=mu2, size=np
    .size(inf))
infected_new[inf] = 1-Ber_recover #  x*(1-mu2)
ignorants_new[inf] = Ber_recover # x*mu2

# ------------------------------------------
# Dynamics of aware individuals
# Aware individuals can forget (or not) and then
    can be infected (or not)
for a in aw:
    Ber_forget = np.random.binomial(n=1,p=mu1) # =1
         if the node forgets, = 0 if it's still
        aware

    neighbors2_a = G2.neighbors(a)
    nnb2 = np.size(neighbors2_a)

    neighbors2_a = np.array(neighbors2_a)
    R2_a = 1./float(nnb2)
    contaminated = 0
    c = 0
    for i in neighbors2_a:
        r1 = np.random.uniform(0,1)
        # r2 = np.random.uniform(0,1)
        # if r1 < beta2 and r2 < R2_a and infected[i]==1:
        if r1 < beta2*R2_a*infected[i]:
            contaminated=1
        #       c = 1
        # contaminated = np.max([contaminated,c])

    infected_new[a] = contaminated          # w*(1-q_a)
    aware_new[a] = (1-contaminated)*(1-Ber_forget)  # w*q_a*(1-mu1)
    ignorants_new[a]=(1-contaminated)*(Ber_forget)  # w*q_a*mu1

#-------------------------------------
# Dynamics of ignorant individuals
for m in ign:
    neighbors1_m = G1.neighbors(m)
    neighbors2_m = G2.neighbors(m)

    # Hypothesis: only infected nodes comunicate their own status
    informed = 0
    d=0
    for j in neighbors1_m: # node m can get informed by the neighbors
        r1 = np.random.uniform(0,1)
        if r1 < beta1*R1[m,j]*infected[j]:
            informed = 1
        #       d = 1
        # informed = np.max([informed,d])

    contaminated = 0
```

```
        c=0
        for i in neighbors2_m:
            r2 = np.random.uniform(0,1)
            if r2 < beta2*R2[m,i]*infected[i]:
                contaminated=1
            #     c = 1
            # contaminated = np.max([contaminated,c])

        infected_new[m] = contaminated
        aware_new[m] = (1-contaminated)*informed
        ignorants_new[m] = (1-contaminated)*(1-informed)
    return infected_new, aware_new, ignorants_new
# ----------------------------------------
# ----------------------------------------
def QS_MC3(G1,G2,R1,R2,A2,k1,k2,l1,l2,beta1,beta2,mu2,mu1,n,tmax,inf0, M):
    # Quasi-stationary Monte Carlo
    inf0 = np.random.randint(0,n,np.int(n*inf0)) # select randomly which nodes are infected
    infected = np.zeros(n)
    infected[inf0] = 1 # Mark with one the infected nodes
    aware = np.zeros(n)
    ignorants = np.ones(n)-infected

    rho_MC = np.zeros(tmax)
    awareness_MC = np.zeros(tmax)
    ignorance_MC = np.zeros(tmax)
    awww = np.zeros(tmax)
    rho_MC[0] = sum(infected)/float(n)
    awareness_MC[0] = sum(aware)/float(n)
    ignorance_MC[0] = sum(ignorants)/float(n)
```

```
awww[0] =rho_MC[0]
history_inf = np.zeros((n,tmax))
history_inf[:,0] = infected
history_aw = np.zeros((n,tmax))
history_aw[:,0] = aware
for t in range(1,tmax):
    infected_new, aware_new, ignorants_new =
       MC_time_step3(infected,aware,ignorants,G1,G2
       ,R1,R2,A2,k1,k2,beta1,beta2,mu2,mu1,l1,l2, n
       )
    rho_MC[t] = np.sum(infected_new)/float(n)
    awareness_MC[t] = np.sum(aware_new)/float(n)
    if rho_MC[t] == 0:
        # Take an active state
        r = random.randint(1,min(t,M))
        infected_new = np.copy(history_inf[:,t-r])
        aware_new = np.copy(history_aw[:,t-r])
        ignorants_new = np.ones(n)-infected_new-
            aware_new
        #print t, 'subtituting with ', t-r

    history_inf[:,t] = np.copy(infected_new)
    history_aw[:,t] = np.copy(aware_new)

    infected = np.copy(infected_new)
    ignorants = np.copy(ignorants_new)
    aware = np.copy(aware_new)

    rho_MC[t] = np.sum(infected_new)/float(n)
    awareness_MC[t] = np.sum(aware_new)/float(n)
    ignorance_MC[t] = np.sum(ignorants_new)/float(n
       )
    return rho_MC, awareness_MC, ignorance_MC

def QS_steady3(G1,G2,R1,R2,A2,k1,k2,l1,l2,beta1,beta2,
   mu2,mu1,n,tmax,inf0, M, av_last):
    # Take the average of the last M values of the
        quasi-stationary MC
    rho_MC, awareness_MC, ignorance_MC = QS_MC3(G1,G2,
        R1,R2,A2,k1,k2,l1,l2,beta1,beta2,mu2,mu1,n,tmax,
        inf0, M)

    rho_mean = np.mean(rho_MC[(-av_last):-1])
    aware_mean = np.mean(awareness_MC[(-av_last):-1])
    ignorance_mean = np.mean(ignorance_MC[(-av_last)
        :-1])

    return rho_mean, aware_mean, ignorance_mean, rho_MC
        [-av_last:], awareness_MC[-av_last:]
```

MC_vs_det_steady.py

```
import numpy as np
import matplotlib.pyplot as plt
import networkx as nx
import scipy as sp
from ML_functions_2L import contact_mat,
   Create_SF_Configuration, built_adj,
   iter_dynamic_eq_lambda, solve_until_steady_lambda
import scipy.optimize
import math
from Create_RR_2L import create_RR_2L
from funct_MC_contacts3 import MC_contacts_steady3,
   QS_steady3
import pickle
```

```
# ==== DESCRIPTION ====
# Phase diagram of rho-beta2 comparing both the Monte
    Carlo solution and the Markov solution.
# Markov solution is made iterating the equation up to
    a time tmax

# Index 1 is refered to the information layer
# Index 2 is refered to the disease layer

# ------------ PARAMETERS ------------
n = 300 # Number of nodes per layer
beta1 = 0.2
mu = 0.2
nu = 0
iterations = 30
change2_min = 0.
change2_max = 0.6
tmax = 400
inf0 = 0.05 # initial number of infected
M = 30 # choose from active configuration of MC
tmax_MC = n*2
av_last = n

# ------------ INITIALIZATION ------------
change2 = np.zeros(iterations, float)
step2 = (change2_max-change2_min)/float(iterations-1)
rho = np.zeros(iterations)
awareness = np.zeros(iterations)
rho_MC2 = np.zeros(iterations)
rho_MC = np.zeros(iterations)
awareness_MC = np.zeros(iterations)
ignorance_MC = np.zeros(iterations)
susceptibility = np.zeros(iterations)
```

```
STD = np.zeros(iterations)
num_inf = np.zeros((av_last))
colors = ['b', 'g', 'r', 'c', 'm', 'y', 'k']

# ---------- CREATE THE GRAPH ----------
# Number of contacts
l1 = 50
l2 = 50

# pickle_in1 = open('ER1_n500_k10.pkl', 'rb')
# G1 = pickle.load(pickle_in1)
# pickle_in2 = open('ER2_n500_k10.pkl', 'rb')
# G2 = pickle.load(pickle_in2)

k_mean1 = 6
k_mean2 = 6
p1 =float(k_mean1)/float(n-1)
p2 =float(k_mean2)/float(n-1)

G1 = nx.random_regular_graph(k_mean1, n, seed=None)
k1 = nx.degree(G1).values()
A1 = nx.to_numpy_matrix(G1)
k1 = np.array(k1)
k1 = k1.astype(float)
G2 = nx.random_regular_graph(k_mean2, n, seed=None)
k2 = nx.degree(G2).values()
A2 = nx.to_numpy_matrix(G2)
k2 = np.array(k2)
k2 = k2.astype(float)

R1 = contact_mat(A1, n, l1, k1)
R2 = contact_mat(A2, n, l2, k2)
```

```
b1 = beta1
Rlam = 1.-(1.-1./float(k_mean1))**l1
b2 = (-k_mean1*Rlam*b1+mu + np.sqrt(  (k_mean1*Rlam*b1-
   mu)**2 + 4*(k_mean1*Rlam)**2*mu*b1  )  )   /float(2*
   k_mean1*Rlam)
print b2 # Analytical critical point

# ---------------- COMPUTATIONS ----------------
for i in range(iterations):
     p_old = inf0*np.ones(n)
     w_old = 0.0*np.ones(n)
     x_old = (1-p_old)*np.ones(n)
     change2[i] = change2_min + i*step2
     beta2  = change2[i]
     #rho[i], awareness[i], c =
         solve_until_steady_lambda(p_old, w_old,x_old,
         change2[i],beta1, R1, A2,l1,l2, k1, k2, mu, nu,
         n, tmax)

     rho_MC[i], awareness_MC[i], ignorance_MC[i],
         rho_last = QS_steady3(G1,G2,R1,R2,A2,k1,k2,l1,l2
         ,beta1,change2[i],mu,nu,n,tmax_MC,inf0,M,
         av_last)
     print i, change2[i], rho[i], rho_MC[i]


# -------- PLOT --------
plt.figure(1)
plt.plot(change2,rho, label = 'Markov␣chain')
label = r'Awareness␣for␣$\beta1$␣=␣%1.1f'%change1)
plt.hold
plt.plot(change2,rho_MC,'*', label = 'Monte␣Carlo')
plt.legend()
plt.title(r'How␣the␣steady␣state␣density␣changes␣with␣$
    \beta_D$''␣\n'r'$\mu_2$␣=%1.1f,␣$\mu_1$␣=␣%1.1f,␣n␣
    =%i␣'%(mu,nu, n))
plt.xlabel(r'$\beta_D$')
plt.ylabel(r'$\rho$')
plt.show()
```

heat_map.py

```
import numpy as np
import matplotlib.pyplot as plt
import networkx as nx
import scipy as sp
from ML_functions_2L import contact_mat,
    iter_dynamic_eq_lambda, solve_until_steady_lambda
import scipy.optimize
import math
from Create_RR_2L import create_RR_2L
import random
from matplotlib.colors import ListedColormap


# Number of nodes per layer. Later I take them randomly
n = 300
# Number of contacts
l1 = 50
l2 = 50
# Intralayer adjacency matrices
k_mean1 = 8
k_mean2 = 8
p1 =float(k_mean1)/float(n-1)
p2 =float(k_mean2)/float(n-1)
m=0
while m == 0:
    G1 = nx.erdos_renyi_graph(n,p1)
    m = min(G1.degree().values())
print 'G1␣done', min(G1.degree().values())
m2=0
while m2 == 0:
    G2 = nx.erdos_renyi_graph(n,p2)
    m2 = min(G2.degree().values())
print 'G2␣done', min(G2.degree().values())

A1 = nx.to_numpy_matrix(G1)
k1 = nx.degree(G1).values()
k1 = np.array(k1)
k1 = k1.astype(float)

k2 = nx.degree(G2).values()
A2 = nx.to_numpy_matrix(G2)
k2 = np.array(k2)
k2 = k2.astype(float)

mu1 = 0.
mu2 = 0.5
#beta1= 0.5
#beta2 = 0.1
R1 = contact_mat(A1, n, l1, k1)
R2 = contact_mat(A2, n, l2, k2)

iterations = 80
change_min = 0.
change_max = 1.0
change = np.empty(iterations, float)
step = (change_max-change_min)/float(iterations -1)
change1 = np.zeros(iterations)
change2 = np.zeros(iterations)
```

```
rho = np.zeros((iterations, iterations))
awareness = np.zeros((iterations, iterations))
inf0 = 0.05 # initial number of infected
tmax = 200

p = inf0*np.ones(n)
w = np.zeros(n)
x = np.ones(n)*(1-inf0)

for i in range(iterations):
    change1[i] = change_min+step*i
    for j in range(iterations):
        change2[j] = change_min+step*j
        rho[i,j], awareness[i,j], c =
            solve_until_steady_lambda(p, w, x, change2[j
            ], change1[i], R1, A2,l1,l2, k1, k2, mu2,
            mu1, n, tmax)
        print i,j

# SAVE RESULTS
np.save('heatmap_ER_n300_k8_mu1_0_mu2_05_rho', rho)
np.save('heatmap_ER_n300_k8_mu1_0_mu2_05_awareness',
    awareness)
np.save('heatmap_ER_n300_k8_mu1_0_mu2_05_betas',
    change2)

plt.pcolormesh(change1,change2,rho)
plt.colorbar()
plt.xlabel(r'$\beta_1$')
plt.ylabel(r'$\beta_2$')
#plt.title(r'$\mu_2$ =%1.1f, $\mu_1$ = %1.1f, n =%i '%(
    mu2,mu1, n))
```

```
plt.show()
```

---

change_n.py

```
import numpy as np
import matplotlib.pyplot as plt
import networkx as nx
import scipy as sp
from ML_functions_2L import contact_mat,
    Create_SF_Configuration, built_adj,
    iter_dynamic_eq_lambda, solve_until_steady_lambda
import scipy.optimize
import math
from Create_RR_2L import create_RR_2L
from funct_MC_contacts2 import MonteCarlo_contacts2,
    MC_contacts_steady
from funct_MC_contacts3 import QS_steady3
import time

# Change the size of the networks and perform Monte
    Carlo
# in order to see the effect of the scaling on the
    susceptibility

# Parameters
beta1 = 0.2
mu2 = 0.2
mu1 = 0.
l1 = 50
l2 = 50
M = 30
inf0 = 0.05
iterations = 60
change2_min = 0.05
```

```
change2_max = 0.15
change2 = np.zeros(iterations, float)
step2 = (change2_max-change2_min)/float(iterations-1)
nn = np.array([250,500,750,1000,1250,1500,1750,2000])
size = np.size(nn)
tmax_markov = 300

# Initialize variables
rho = np.zeros((size,iterations))
awareness = np.zeros((size,iterations))
ignorance = np.zeros((size,iterations))
contacts = np.zeros((size,iterations))
rho_MC = np.zeros((size,iterations))
awareness_MC = np.zeros((size,iterations))
ignorance_MC = np.zeros((size,iterations))
susceptibility = np.zeros((size,iterations))
STD = np.zeros((size,iterations))

colors = ['b', 'g', 'r', 'c', 'm', 'y', 'k']

# Computations
for j in range(size):
    n = nn[j]
    tmax_MC = np.min([n*4,3000])
    av_last = np.min([n*3,2000])

    k_mean1 = 6
    k_mean2 = 6

    G1 = nx.random_regular_graph(k_mean1, n, seed=None)
    k1 = nx.degree(G1).values()
    A1 = nx.to_numpy_matrix(G1)
    k1 = np.array(k1)
```

```
k1 = k1.astype(float)
G2 = nx.random_regular_graph(k_mean2, n, seed=None)
k2 = nx.degree(G2).values()
A2 = nx.to_numpy_matrix(G2)
k2 = np.array(k2)
k2 = k2.astype(float)

R1 = contact_mat(A1, n, l1, k1)
R2 = contact_mat(A2, n, l2, k2)

rho_last = np.zeros((av_last,iterations))
aw_last = np.zeros((av_last,iterations))
for i in range(iterations):
    p_old = inf0*np.ones(n)
    w_old = 0.00*np.ones(n)
    x_old = (1-inf0)*np.ones(n)
    change2[i] = change2_min + i*step2
    #rho[j,i], awareness[j,i], ignorance[j,i] =
        solve_until_steady_lambda(p_old, w_old,
        x_old, change2[i],beta1, R1, A2,l1,l2, k1,
        k2, mu2, mu1, n, tmax_markov)

    num_inf = np.zeros(av_last) # last av_last
        number of infected in the simulations

    rho_s, awareness_s, ignorance_s, rho_last[:,i],
        aw_last[:,i] = QS_steady3(G1,G2,R1,R2,A2,k1,
        k2,l1,l2,beta1,change2[i],mu2,mu1,n,tmax_MC,
        inf0,M, av_last)
    num_inf = rho_last[:,i]*n
    num_aw = aw_last[:,i]*n
    rho_MC[j,i] = np.mean(num_inf)/float(n)
    awareness_MC[j,i] = np.mean(num_aw)/float(n)
```

```
            susceptibility[j,i] = np.var(num_inf)/np.mean(
                num_inf)
            STD[j,i] = np.std(num_inf/n)
            print n,i # STD[j,i], np.sqrt(susceptibility[j,
                i]*rho_MC[j,i]/float(n))
            time.sleep(3)

    # SAVE RESULTS

    # np.save('change_n%i_ER_k%i_%iter_rho'%(n,k_mean1,
        iterations), rho)
    # np.save('change_n%i_ER_k%i_%iter_awareness'%(n,
        k_mean1,iterations), awareness)
    np.save('change_n%i_RR_k%i_%iter_rhoMC'%(n,k_mean1,
        iterations), rho_MC[j,:])
    np.save('change_n%i_RR_k%i_%iter_awarenessMC'%(n,
        k_mean1,iterations), awareness_MC[j,:])
    np.save('change_n%i_RR_k%i_%iter_aw_lastMC'%(n,
        k_mean1,iterations), aw_last)
    np.save('change_n%i_RR_k%i_%iter_susceptibility'%(n
        ,k_mean1,iterations), susceptibility[j,:])
    np.save('change_n%i_RR_k%i_%iter_std'%(n,k_mean1,
        iterations), STD[j,:])
    np.save('change_n%i_RR_k%i_%iter_rho_last'%(n,
        k_mean1,iterations), rho_last)

for j in range(size):
    n = nn[j]
    col = colors[j]
    plt.figure(1)
    plt.plot(change2,rho[j,:],'%s'%col, label = r'
        Density␣of␣infected␣for␣n␣=␣%i'%n)
    plt.hold
    plt.plot(change2,rho_MC[j,:],'*%s'%col, label = r'
       Density␣of␣infected␣MC␣for␣n␣=␣%i'%n)
    plt.hold
    plt.errorbar(change2, rho_MC[j,:], STD[j,:], marker
       ='*', color = '%s'%col)
    plt.title('$\mu_2$␣=%1.1f,␣$\mu1$␣=␣%1.1f,␣n␣=%i␣'
       %(mu2,mu1, n))
    plt.xlabel(r'$\beta_2$')
    plt.ylabel(r'$\rho$')
    plt.legend()

    plt.figure(2)
    plt.plot(change2,susceptibility[j,:],'%s'%col,
       label = r'Susceptibility␣for␣n␣=␣%i'%n)
    plt.title('$\mu_2$␣=%1.1f,␣$\mu1$␣=␣%1.1f,␣n␣=%i␣'
       %(mu2,mu1, n))
    plt.xlabel(r'$\beta_2$')
    plt.ylabel('Susceptibility')
    plt.legend()
# axes = plt.gca()
# axes.set_ylim([0,1])
plt.show()
```