Cascades on Temporal Networks

Georgios Kapros-Anastasiadis
St Anne's College
University of Oxford

A thesis submitted for the degree of
M.Sc. in Mathematical Modelling and Scientific Computing
September 2012

Acknowledgements

I would like to thank my supervisor Mason Porter for his guidance, help, insight, and support that made the writing of this thesis an incredibly interesting, enjoyable and rewarding experience. I am also grateful to him for introducing me to the exciting field of networks and to the community that enthusiastically undertakes research in this area. I would also like to thank James Gleeson for his help, and Kevin O'Sullivan for his invaluable help and guidance in producing part of the code used in this thesis. I am grateful to Sergey Melnik for his advice and guidance regarding the numerical modelling of random networks, and I would also like to thank Till Hoffmann and Renaud Lambiotte for their assistance and insights during the early stages of my introduction to network science. I also thank Sang Hoon Lee for his helpful comments and for introducing me to previous work similar to the topic of this thesis.

Additionally, I am grateful to Kathryn Gillow and everyone involved in the M.Sc. for making this course an overall amazing experience. A big thank you to all my fellow students in the M.Sc. as well. I am grateful for studying this course with all of you.

Lastly, to my friends and family, who I love and am forever grateful to.

Abstract

In this thesis, we investigate threshold models for information cascades in time-dependent networks. This entails the modelling of temporal networks, the numerical implementation of social contagion dynamics coupled with networks dynamically evolving in time due to edge activation and deactivation rules, the basic implementation of mean-field and pair-approximation theories, and the interdependence of these concepts.

We first introduce networks in general, and then discuss some of their structural properties, dynamical processes on networks, and synthetic networks given by random graphs. We then turn to various threshold models, describe them, and connect them with real phenomena, before focusing on the Watts threshold model and its generalisations. We introduce the emergent field of temporal networks, and we extend the Watts threshold model to account for some temporal activity and run simulations on random graphs. We present our numerical results while trying to get some insight into how edge activation patterns affect information spreading and cascades. Finally, we take a step back to introduce analytical approximations techniques that could prove useful in analytically investigating cascades in temporal networks.

Contents

Chapter 1

Introduction

1.1 An Introduction to Networks

Networks represent collections of entities and the interactions between them [23]. They consist of a set of items (vertices) called *nodes*, and a set of connections (lines) called *edges* [20]. A multitude of systems–both natural and man-made–can be represented as networks, and the great interest in studying them has given rise to a new discipline that is often called network science. Examples of systems that can be represented by networks include technological and information networks–the Internet, the World Wide Web, or telephone networks–, social networks of connections between individuals, networks of business relations, biological networks–neural networks, metabolic networks, delivery networks of blood vessels, protein networks, networks of interactions between species–, and other networks such as networks of citations between papers or networks of words in a language [20, 21, 28].

A network can be written mathematically as $\mathcal{G} = (\mathcal{S}, \mathcal{E})$, where $\mathcal{S} = \{s_1, \ldots, s_N\}$ is a set of N nodes, and $\mathcal{E} = \{\varepsilon_1, \ldots, \varepsilon_M\}$ is a set of M edges connecting these nodes. Such objects are also called *graphs*, and they have been studied primarily in graph theory, which is an area of discrete mathematics [18]. Real systems, however, are large and complex–often consisting of millions or even billions of nodes–and studying very large networks requires additional tools (*e.g.* summary statistics) to determine their structure and behaviour, because analysing the role of each node and edge individually is prohibitive. Therefore, networks are often characterised using methods from statistics to complement analytical methods from graph theory and other subjects, as well as using numerical computation.

To begin characterising networks, we define the *adjacency matrix*. The adjacency

matrix is a matrix $A \in \mathbb{R}^{N\times N}$ where N is the number of nodes

$$A_{ij} = \begin{cases} 1, & \text{if } i \text{ is connected to } j, \\ 0, & \text{otherwise .} \end{cases} \tag{1.1}$$

This matrix encodes the information of the nodes and edges in the simplest type of network, where there is no direction in the connections between nodes and there are no weights assigned to the edges. Such a network is called *undirected* and *unweighted*. On the contrary, a *directed* network differentiates between edges going into nodes, and edges coming out of nodes. We represent an edge going from i to j by $i \to j$. In undirected networks, $i \to j$ is equivalent to $j \to i$, while directed networks can have $i \to j$, $j \to i$, neither of the two, or both edges between node i and node j. Additionally, in a weighted network, each edge is assigned a number, typically positive, representing the strength of connection or some other property, and $A_{ij} \in \mathbb{R}^+$ accordingly. The matrix A in Equation 1.1 corresponds to an unweighted, undirected network. As can be easily seen, it is symmetric, which need not be the case for directed networks. We illustrate these ideas with two examples in Figure 1.1.

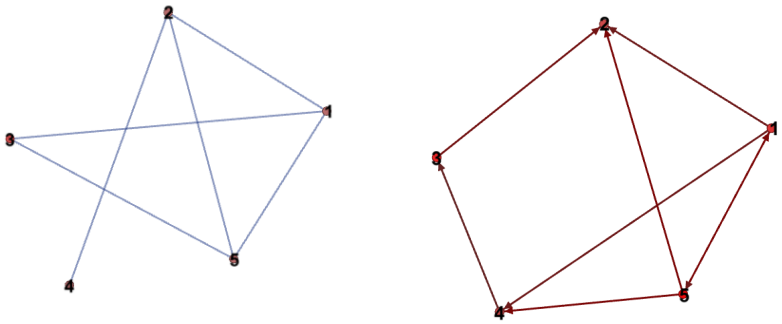

Figure 1.1: An undirected (left) and a directed (right) network with 5 nodes each.

On the left we have an undirected unweighted network with adjacency matrix

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

and on the right a directed unweighted network with adjacency matrix

$$B = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

Note that A is symmetric, whereas B is not. We can use these examples to illustrate two more important ideas. These are the *degree* of a node, and the *degree distribution* of a network. The degree of node i in an undirected network is the number of its neighbours (*i.e.*, the number of edges connected to it). In the case of directed networks, the out-degree is the number of edges emanating from node i, and the in-degree is the number of edges directed towards node i. We will denote the degree in the case of an undirected network by k_i and it can be simply computed by $k_i = \sum_{j=1}^{N} A_{ij}$. As we can see in Figure 1.1, in the undirected network on the left, node number 2 is connected to nodes 1, 5 and 4. Therefore, its degree is 3. We can also add the second row or column of matrix A (because it is symmetric), to deduce the same result.

Collecting the degrees of all N nodes leads to the concept of a degree distribution. The degree distribution is the probability distribution of the degrees of the nodes over the whole network. Let K be a random variable denoting the degree and $P(K)$ be the degree distribution of an undirected network. Then for a specific degree k^*, $P(K = k^*)$ can be thought of either as the fraction of nodes with degree k^* [23] or as the probability that a randomly chosen node has degree k^* [5]. We usually just write p_k with k denoting the degree, although this is an abuse of notation.

There are many other properties that are important when studying networks, but the aforementioned concepts are enough to proceed. While we can use properties such as the degree distribution to understand the structure of networks, in order to capture and predict the behaviour of complex systems, we have to consider the processes and dynamics taking place on networks which may assign rules dictating the behaviour of nodes and edges, and which depend on the underlying structure.

1.2 Dynamical Processes on Networks

The aim of much of the recent research done in network science is to understand systems built upon networks [20] and the interplay between the network structure and the evolution of the system. These systems can be described using dynamical processes taking place on networks, and we would like to capture the dynamics with a model using deterministic or stochastic local dynamic rules. For example, we might want to study surfing on the World Wide Web, voting behaviour in a population, the spread of a disease or (as in our case) the spread of information in a social network. Relevant examples could include almost anything, but only in some cases it has been possible to derive some analytical or otherwise predictive descriptions of such systems.

In one possible type of description, a dynamical process on a network is specified using a differential equation of some kind for each node that represents the time evolution of this node in terms of some quantity of interest. If we assume that there is only 1 variable per node and that each node depends only on its neighbours (which need not be the case) [23], we can write a general formula for the dynamics of node i with respect to a variable x

$$\dot{x}_i = f_i(x_i) + \sum_j A_{ij} g_{ij}(x_i, x_j), \tag{1.2}$$

where f_i represents the intrinsic dynamics of node i, and the second term represents the dynamics resulting from the interaction of i with its neighbours. Because the dynamics are usually very complex, one typically needs to derive approximate equations using various techniques to obtain analytical results. Relevant techniques include mean-field theory and pair-approximation theory, which we will explore in Chapter 5.

It is also very useful to numerically investigate these dynamical systems. This can be done, for example, by analysing empirical data from real networks and then computing various quantities of interest and how they behave as a system evolves. A serious complication, however, is that real networks change in time as well, and a dynamical system on a network and the dynamics of a network itself affect one another in nontrivial ways. Additionally, it is useful to construct artificial networks and then perform numerical simulations following some set of rules that define the dynamics. With this approach, one hopes that the studied networks and processes are sufficiently related to the (much more complicated) real networks and processes on them, in order to be able to derive useful insights about the system of interest.

1.3 Random Networks

Random networks or are graphs constructed by a certain stochastic rule. More formally, they are families or *ensembles* of graphs with this rule or with some probabilistic property. We will use random graphs to investigate the behaviour of dynamical processes and how the topology (*i.e.*, connectivity) of a specific ensemble influences the dynamics in question. Random graphs are also sometimes used as "null models" to real networks [23], aiming to capture a property encountered in various real networks, even if the corresponding real network is significantly different in other ways. For example, certain random graphs display the small-world effect, which means that one could transverse a small number of edges to get from any node to any other node, and that this number of edges scales sufficiently slowly as the size of the network

increases (*e.g.*, logarithmically) [16]. Some real networks also display the small-world effect, as demonstrated by Milgram in his famous "six degrees of separation" experiment [17]. Here we will present the two random graph ensembles which are used in this thesis.

A simple random graph is the Erdős-Rényi or Poisson random graph. This actually refers to two closely related ensembles, which are denoted $\mathcal{G}(N, M)$ and $\mathcal{G}(N, p)$. A $\mathcal{G}(N, p)$ graph can be constructed by the following simple rule. Take N nodes and connect each pair of nodes independently with probability p. Another characterisation of the ensemble is that each graph with exactly M edges appears with probability $p^M(1-p)^{\mu-M}$ over the whole ensemble, where $\mu = \frac{1}{2}N(N-1)$ is the maximum possible number of edges [20]. We can construct a $\mathcal{G}(N, M)$ graph in the following way. Consider N nodes and M edges and choose uniformly at random among all networks with such N, M fixed. If we hold the mean degree z constant and take the limit as $N \to \infty$ in the $\mathcal{G}(N, p)$ ensemble, we can derive that the degree distribution is Poisson [20]

$$p_k = \frac{z^k e^{-z}}{k!}, \tag{1.3}$$

which is why such a graph is sometimes called a Poisson random graph.

Various generalisations have been developed that can account for many different degree distributions. A certain generalisation that we are going to use in our numerical simulations in Chapter 4 is the *configuration model*. We use the following algorithm to construct such an unweighted, undirected network [23]. Suppose there are N nodes. We specify a given degree sequence $\{k_i\}$. This fixes the number of edges $M = \frac{1}{2}\sum_{i=1}^{N} k_i$, because each node has k_i edges emanating from it and we have to divide by 2 to eliminate double counting. The half-ends of edges attached to nodes are called *stubs*, and there are $2M$ stubs in the network. Choose two stubs uniformly at random and connect them with an unweighted, undirected edge. Then choose a pair among the remaining stubs and repeat the same process. The ensemble of graphs produced in this way is called the configuration model. Note that self-edges are allowed, but we are going to artificially remove any self-edges when constructing random networks with the configuration model in the following chapters.

Random graphs can be characterised in ways other than the degree distribution. This role can be performed by the probability generating function $g(w)$, defined by

$$g(w) = \sum_{k=0}^{\infty} p_k w^k. \tag{1.4}$$

This function encodes all the moments of the degree distribution. These are computed

by evaluating the derivatives of $g(w)$ in the following way [23]

$$g(1) = \sum_{k=0}^{\infty} p_k = 1,$$

$$g'(w) = \sum_{k=0}^{\infty} k p_k w^{k-1} \Rightarrow g'(1) = \sum_{k=0}^{\infty} k p_k = \langle k \rangle = z,$$

$$\left. \frac{d^m g}{d(\ln w)^m} \right|_{w=1} = \langle k^m \rangle.$$

In this and other ways, generating functions are a powerful analytical tool. Many of the techniques that use generating functions, however, rely on the assumption that the network is locally tree-like, *i.e.* that there are no, or almost no triangles of edges. Moreover, most of these techniques consider fixed degree distributions.

Chapter 2

Threshold Models and Cascades

2.1 Introduction to Threshold Models

Many dynamical systems consist of some process that spreads through a network, where nodes can affect each other via the transfer of some quantity or by altering each other's state. When local interactions among a small set of nodes cause propagation to the whole network or at least to a large part of it, we say that there has been a *cascade* [8]. The term originates from the study of social networks, but a similar effect is observed and described with different terminology in other networks as well. In biological networks, when considering the spread of a disease, there is the possibility of an epidemic which quickly leads to the infection of a large part of the whole network [1]. Similar events could occur in the spread of computer viruses through e-mail networks or the Internet [10], in the critical failure of a technological network such as the power failure of a power distribution network [30], or when considering the crash of a financial network [35].

We are often interested in the spread of information, ideas, or opinions on social networks. In the context of modelling the spread of social influence, we can model the spread of rumours, the collective adoption of ideas, and the adoption of fads as cascades in a social network. Inspired by biological epidemic models, the spread of social influence is sometimes called *social contagion* [26]. By considering the nodes of the networks as having states and letting these states change according to the states of each node's neighbours' states, we can study these problems using *threshold models*. Threshold models, in the most common case, state that the rule for changing the states of nodes is that a state change occurs when the fraction or the number of a node's neighbours with a certain state is above a given threshold. Consider, for example, a situation in which the nodes of a network have one of two states: state 0 or state 1. State 1 might correspond to "adopted" and state 0 to "unadopted",

which could model the adoption of ideas, the purchase of a product or service, or the spreading of a rumour, where an "adopted" node adopts the role of gossiper and spreads the rumour, and an "unadopted" node ignores or rejects it. Because nodes can only be in two possible states, these are examples of binary-state threshold models. If a node of state 0 switches to state 1 when the fraction of its neighbours with state 1 exceeds a certain threshold, the model is a fractional threshold model. Accordingly, when we consider the number of neighbours, we refer to an absolute threshold model instead. Another distinction is between monotonic and non-monotonic dynamics. The former refers to models where any node with state 1 can never switch to state 0 and thus the dynamics always move from state 0 to state 1 (or not at all). In the latter case we consider both directions, having nodes which can switch from state 1 to state 0.

A convenient way to formulate a threshold model is by defining a *response function*. A response function allows one to write the possible transitions in a cumulative way with respect to some variables. Let $\sigma(i)$ denote the state of node i, so σ is either 1 or 0. Also let $\Gamma(i)$ be the set of neighbours of node i. The number of neighbours of node i with state 1 is $l = \sum_{j \in \Gamma(i)} \sigma(j)$. Consider a monotonic fractional threshold model in which all of the nodes have a common threshold R. We can then write the response function $F(l, k)$, where k is the degree, as

$$F(l,k) = \begin{cases} 0, & l \leq Rk, \\ 1, & l > Rk. \end{cases} \tag{2.1}$$

This can be thought of as the probability that a node of degree k with l state 1 neighbours adopts state 1 in time dt (if we consider the network to be updated in continuous time) or in time t (if we consider discrete updates–*e.g.*, one at time t and the next at $t + 1$). Note that a state-1 node cannot change states because the dynamics are monotonic. We can use this formulation to distinguish between different social influence models and even consider models of biological epidemics, as we show in Section 2.2.

2.2 Examples of Threshold Models

First, let us briefly consider a family of epidemiological models, which constitute some of the most widely studied dynamical systems in network science. Many results that apply elsewhere have been formulated in terms of these models, and although they are typically presented in another fashion, some of them can be written using the language of threshold models. Two of these are the SI and the SIS models of disease dynamics. Consider a network comprised of a population of individuals

and the connections between them. There is a disease that divides the population in two camps: the infected (*i.e.*, individuals that have contracted the disease) and the susceptible (*i.e.*, the individuals that do not have the disease but can get it in the future). More complicated models extend this by including immune and recovered individuals, by considering a changing populations with births and deaths, or by incorporating other factors. Here, S stands for the fraction of susceptible individuals among the population and I stands for the fraction of infected individuals. In the case of the SI model, the dynamics are monotonic because the immunity is permanent; in the SIS model, however, an infected individual can revert back to being susceptible.

Consider the SI model. The simplest (although naive) way to approach the problem is to consider a simple rate of infection β and assume that the population is well-mixed so that every node contributes equally and independently to the transmission of the disease [23]. This is a type of mean-field assumption, as we will see in Chapter 5. In this case, the nodes in $\mathcal{S}$ and $\mathcal{I}$–the sets of susceptible and infected nodes–are only affected by S, I and β. We can thus write the ordinary differential equations

$$\begin{aligned} \dot{S} &= -\beta SI, \\ \dot{I} &= \beta SI. \end{aligned} \tag{2.2}$$

As $S + I = 1$, the second equation becomes $\dot{I} = \beta(1 - I)I$, which can be solved analytically (it is the logistic equation) [23].

Another, more realistic, way to approach the dynamics of epidemics is to consider the interactions of neighbouring nodes. We can then still consider a uniform infection rate β, but now a susceptible node is affected only by its infected neighbours. Let the number of infected neighours be denoted by l. We see that a node will be affected by its infected neighbours independent of any thresholds, but this is the same as having a fractional threshold of 1. Therefore we can write a response function for susceptible individuals

$$F(l) = \beta l. \tag{2.3}$$

Deriving a differential equation ruling global dynamics is more difficult in this case, but it is possible (see Chapter 5).

We now turn to threshold models of social influence. Here, the response function is a step function, reflecting the assumption that nodes will change state if a threshold is reached. Various models have been developed and they have been connected to various social phenomena in different ways.

The Watts threshold model [34] was developed to model cascades in the transmission of rumours, innovations, or cultural fads. It is a binary-state model, with state 1 corresponding to "adopted", and state 0 corresponding to "unadopted". As argued

in [34], a binary-state framework covers a surprising range of social situations. A decision is often made between just two options, where although there are many complex factors involved in the decision itself, in the end an individual chooses between two options according to the decisions of other individuals. This reflects situations in which people are influenced heavily by the decisions of others and do not act and think independently. Various relevant social phenomena have been proposed, although this binary behaviour might not fully describe the phenomena sociologically [6]. For example, someone who hears a rumour might not try to independently verify the truth, and instead rely on the person telling him about it. One can think of more examples, such as choosing a restaurant or buying a popular product. In the Watts model, each node is assigned a threshold R_i taken from a probability distribution of thresholds. All nodes have state 0. A single node then adopts state 1. This constitutes the initial condition of the model, where one node has state 1 and all other nodes have state 0. Each node looks at its neighbours, and if the fraction $\frac{l_i}{k_i}$ of its neighbours in state 1 exceeds its threshold R_i then it adopts. A node with state 1 keeps its state forever, thus the dynamics are monotonic. If we consider a uniform threshold R, the response function is given by Equation 2.1.

We can also formulate an absolute threshold variant of the above model, as has been studied in [2] and [8] and discussed in [6]. In this case, a node adopts when a *number* of its state-1 neighbours (instead of a fraction) exceeds a threshold. This reflects a situation in which the "unadopted" neighbours exert no influence on an individual, contrary to the Watts model, in which a larger number of state-0 neighbours lowers the fraction $\frac{l_i}{k_i}$. One can imagine various situations in which this would be appropriate. For example, in spontaneous street riots, individuals might be encouraged by others taking part, but they might not be discouraged if others do not. Other examples could be urban legends or conspiracy theories, where, sometimes, it is the number of people that mention them that matters, and not the people who do not [6]. Assuming that there is a uniform threshold of number of state 1 neighbours Θ, and denoting the number of state 1 neighbours by l, the response function for this model is

$$F(l) = \begin{cases} 0, & l \leq \Theta, \\ 1, & l > \Theta. \end{cases} \tag{2.4}$$

Another threshold model is the multi-stage complex contagion model developed in [26]. Simple contagion refers to a process where the effect of a single neighbouring node is enough to change the state of another node, while complex contagion refers to systems where a node has to interact with multiple neighbours to evolve. Our previous two examples constitute cases of such complex contagion models, while disease epidemics correspond to simple contagion, because a single infected node infects a susceptible neighbour. In the case of multi-stage complex contagion, we consider

three possible influence levels for each node of the network (although the model can be extended to any finite number of levels). The nodes are divided to "inactive" (state S_0), "active" (state S_1) and "hyperactive" (state S_2) nodes. This is a natural extension to the binary-state framework we have examined thus far, and it makes the model more realistic, as it accounts for varying levels of influence, interest, and commitment that is often present in social influence in real networks. For example, a dedicated proponent of a social movement plays a different role in that movement than someone who has merely adopted its ideas, and regular users of a product spread a product more enthusiastically than casual users (or we could say that advertisers promote the product to users more than users do to other users), and so on [26]. We can then consider individual response functions for each of the N nodes of a network as the probability that a degree-k node switches to state S_i, $i = 1, 2$, given that the number of its neighbours in state S_1 is l_1 and the number of its neighbours in state S_2 is l_2. We then define $P = (l_1 + \alpha l_2)/k$ as the "peer pressure" corresponding to the total amount of influence received by a node of degree k by its neighbours in S_1 and S_2. The parameter α measures the bonus influence exerted by "hyperactive" nodes and increases the probability of adoption. For a uniform threshold R, we have the response function

$$F(l_1, l_2, k) = \begin{cases} 1, & \text{if } \ (l_1 + \alpha l_2)/k \geq R, \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

As investigated in [26], the presence of different levels of social influence can have a significant effect on network cascades.

2.3 Watts Model

In this section, we focus on the Watts threshold model to hopefully gain some general insight on cascades. In the original Watts paper [34], the model is applied on random graphs and this is the approach that we will follow as well, although one could apply the model on real networks drawn from empirical data. We follow this approach in order to be able to control a dynamically changing random network in later investigations of time-dependent networks.

To apply the Watts model on random networks, we have to express it as a numerical algorithm. We present pseudo-code in Algorithm 1, which we will explain. The network we use is the Poisson random graph. Additionally, we consider only uniform thresholds.

Algorithm 1 Watts Threshold Model

procedure WATTSMODEL(N, z, R) % N: number of nodes. z: mean degree. R: uniform threshold
    Construct Poisson random graph
    Compute k_i for $i = 1, \ldots, N$ % compute the degrees of all the nodes
    Set all node states σ to $\sigma(i) = 0$ for $i = 1, \ldots, N$ % set all initial states to 0
    Choose a node j at random and set $\sigma(j) = 1$ % set a random node's state to 1
    Change state to 1 if $\sigma(i) = 0$ and if $\sum_{j \in \Gamma(i)} \sigma(j)/k_i > R$ for $i = 1, \ldots, N$ % update states according to threshold rule
    Save new states in a new vector σ_{new} % keeping the old states in σ
    while $\sigma \neq \sigma_{\text{new}}$ **do** % while new nodes have adopted
        Set $\sigma = \sigma_{\text{new}}$ % update the state vector
        Change state to 1 if $\sigma(i) = 0$ and if $\sum_{j \in \Gamma(i)} \sigma(j)/k_i \geq R$ for $i = 1, \ldots, N$ % update states according to threshold rule
        Save new states in a new vector σ_{new} % keeping the old states in σ
    end while
end procedure

First, to construct a Poisson random graph, we connect any pair of the N nodes with an edge with a probability $p = z/N$, where z is the mean degree. The graph can also be constructed by specifying the Poisson degree distribution $p_k = z^k e^{-z}/k!$ on the configuration model (through the degree sequence $\{k_i\}$), as described in Chapter 1. We then set all nodes to state 0, flip a random node to state 1, and update the network according to the threshold rule. The updating is done in a synchronous fashion, as all N nodes are updated simultaneously (*i.e.*, on the same update loop). The process continues until the states of nodes stop changing. Note that $\Gamma(i)$ denotes the set of neighbours of node i.

We now turn to some of the arguments and analytical results that Watts presented in his paper [34]. The following is taken directly from his paper and does not constitute our own work. Our purpose is to present it along with some explanations to help convey some interesting facts about cascades. Watts distinguished nodes in the network between *vulnerable* and *stable* nodes. These are defined in terms of the growth of the initial seed (consisting of one state-1 node). In a large, sparsely connected (with a relatively low mean degree) random graph such as the Poisson random graph, there are approximately no short cycles or triangles (*i.e.*, any edges between any three nodes never form a triangle), and thus the graph has a locally tree-like structure. Therefore, the initial seed grows (changing the state of its neighbours) if

at least one of its neighbours has a degree $k \leq 1/R$. Nodes with this property are vulnerable, and those without it are stable. In fact, the structure of the community of these vulnerable nodes is crucial to whether a cascade occurs or not. Watts conjectured that the largest connected cluster of vulnerable nodes has to occupy a finite fraction of an infinite network (*i.e.* as $N \to \infty$) in order for a cascade to occur. This means that the fraction of vulnerable nodes has to diverge to infinity as $N \to \infty$. In this case, we say that the vulnerable cluster *percolates*, so the conjecture states that the vulnerable cluster needs to percolate for a global cascade to be possible. Using this and the following calculations, one can derive a necessary condition for a cascade to occur. This condition is, of course, related to the connectivity of the network.

Watts uses a generating function approach. We know that the probability that any node has degree k is given by the degree distribution p_k. Similarly, the probability that a node with degree k is a vulnerable node is $\rho_k = P(R \leq 1/k)$. Assuming independence, the probability that some node u has degree k *and* is vulnerable is $\rho_k p_k$. Therefore, we can write the corresponding generating function

$$G_0(x) = \sum_k \rho_k p_k x^k, \tag{2.6}$$

which encodes all of the information of the degree distribution of vulnerable nodes. To derive results for vulnerable clusters, we have to determine the degree distribution of a vulnerable node v that is a random neighbour to the initially chosen node u [34]. The probability of choosing v is proportional to kp_k, where k is the degree of v. This is because the larger the degree of v, the greater the probability that v is a neighbour of u, and thus the more likely that v is chosen. The corresponding normalised generating function for a random neighbour of u is

$$G_1(x) = \frac{\sum_k \rho_k p_k x^{k-1}}{\sum_k k p_k} = \frac{G_0'(x)}{z}. \tag{2.7}$$

By introducing analogous generating functions for vulnerable clusters, denoted by $H_0(x)$ for the generating function corresponding to a randomly chosen node belonging to vulnerable cluster of size n, and by $H_1(x)$ for the one corresponding to the neighbour of that chosen node, it is stated in [34] that the average cluster size $\langle n \rangle = H_0'(1)$ is given by

$$\langle n \rangle = G_0(1) + (G_0'(1))^2/(z - G_0''(1)), \tag{2.8}$$

which diverges when $G_0''(1) = z$. The condition requires that the vulnerable cluster reaches a finite size of an infinite network, which yields a cascade condition:

$$G_0''(1) = \sum_k k(k-1)\rho_k p_k = z. \tag{2.9}$$

This equation thus distinguishes between two regimes: one in which $G_0''(1) < z$, the vulnerable clusters are small, and the "adopted" nodes become isolated; and one

in which $G_0''(1) > z$ and the size and structure of the vulnerable clusters allow the possibility of a cascade. Therefore, an important consequence of this model is the presence of a phase transition in terms of the mean degree z.

We also consider some numerical results from Watts' paper [34]. To investigate the relationship between the frequency and the size of cascades to the connectivity of the network, we consider a figure from that paper, namely Figure 2(b). This figure depicts the fraction of "adopted" nodes versus z, with $R = 0.18$. We refer to this fraction as the cascade size. The cascade size depends on the mean degree of the network, with a sudden transition occuring when z reaches a certain large value. The cascade size is constrained when the mean degree is low in the initial values of z, which is to be expected because, even if the initial seed causes its neighbours to switch states, the nodes that have adopted become isolated due to the low connectivity. More interesting is the fact that a large mean degree prohibits the occurence of cascades. A possible explanation, as discussed in [34], is that when the mean degree becomes larger, there are two phenomena that are occuring simultaneously. Firstly, some nodes in the network have many neighbours. Because only one seed node has state 1 initially, even if it spreads to some nodes, these nodes with high degree will never change their state. Thus, the spreading process dies out and the cascade size is small. Secondly, if the initial seed hits the percolating vulnerable cluster, a cascade might occur, and its size will be large (perhaps covering the whole network). This event, however, becomes rarer as the connectivity gets higher, and the first scenario becomes more probable, until at some point, global cascades become impossible. This sudden transition was actually shown to be discontinuous and was shown to happen due to a certain transcritical bifurcation [11]; we will come back to this in later chapters.

The implementation of the Watts threshold model on random graphs depends on the type of random graph, the initial conditions, the distribution of thresholds, and, of course, the update rule of the model. As was already done in [34], one could consider different random graphs and consider probability distributions of thresholds. Random graphs with degree heterogeneity seem to constitute a network more stable to cascades, while threshold heterogeneity makes the network less stable and the cascades more likely. Although both have a cascade-promoting effect on the local switching of states due to the fractional threshold condition, low mean degrees in graphs with heterogeneous degree distributions have an inhibiting effect on cascades because nodes become isolated, while low thresholds can only be positive towards cascades.

Studying the effects of different initial seed sizes [11] leads to new results, which we will replicate and discuss in Chapter 4. Another variation that has been studied [9] is that of considering random networks with community structure or degree-degree correlations. Degree-degree correlations describe the situation of nodes having positive or negative correlations with respect to the connections between each other. Such correlations in terms of degree mean that nodes connect to other nodes of similar degree in the case of positive correlations, and connect to nodes of dissimilar degree in the case of negative correlations. The former is an example of assortative mixing (or "homophily"), and the latter is an example of disassortative mixing (or "heterophily") [19]. Moreover, many networks are observed to be divided in communitites, and some network features might vary from one community to another. Communities are groups of nodes with dense connections between them but sparse connections with nodes of other groups [23]. An extension to the investigation we perform in this thesis would be to consider random graphs that exhibit some of the aforementioned properties.

Another interesting extension was done in [32], where a model of history-dependent contagion was considered. Instead of just having states of 0 and 1, they also assign a variable v_i to each node whose value increases every time node i has state 0 and interacts with a state 1 node. When the value of v_i reaches a certain threshold v_{thr}, the state 0 node i (assuming it is still a state-0 node) becomes an "adopted" (*i.e.*, state-1) node. In the absence of interactions between state 1 and state 0 nodes, however, v_i decays exponentially in time. This framework, for example, could model the interest of an individual in a certain topic, and the decay would represented the loss of interest in the absence of interactions. This bursty nature of information spreading–using periods of interactions followed by periods of inactivity–was found to facilitate spreading and the occurence of cascades.

Chapter 3

Temporal Networks

3.1 Introduction to Temporal Networks

All of the examples we have considered thus far, have been static networks. Static networks retain the same network topology and edge weights in time, but many real complex systems change in time and the network itself evolves. In static networks, the explicit time-dependence of networks is usually discarded or only partially incorporated in some way. Traditionally, when studying empirical networks, the time dimension of network structure has been accounted for by using static networks in various ways. For example, the interactions in time between nodes can be aggregated in weighted edges of a weighted static network, where, if we assume that we are looking at the time evolution of a network up to a certain time, a larger weight might correspond to more interactions (in either number or duration) during the time interval under consideration [22]. Such aggregation, however, implicitly assumes that the evolution of the edges (the activation and deactivation in time) follows a Poisson process, where an edge appears at some time chosen uniformly at random in a time interval [31]. However, numerous complex systems, especially social and economic networks, exhibit non-Poisson temporal statistics, and thus such an aggregation and depiction in a weighted static network is problematic and inaccurate [31].

Another problem with static networks with aggregated edges is that they do not account for various characteristics of temporal network structure. For example, edges do not have to be transitive. Let $i \to j$ denote an undirected, unweighted edge between node i and node j. In static networks, edges are always transitive, *i.e.* if $i \to j$ and $j \to k$, then it is true that $i \to k$. In a static network with aggregated edges we might encounter a situation in which two such edges are present and thus transitive, but in the actual corresponding time-evolving network, they are actually not tran-

sitive. If $i \to j$ and $j \to k$ are active at two non-overlapping time intervals, then i cannot be connected to k, and the static depiction ignores this lack of transitivity [22].

Temporal networks are networks in which time-dependence is build into the network itself. Such networks, therefore, have an explicit temporal structure in addition to the topological structure and edge weights of static networks. The study of temporal networks is an emerging field of network science that is being developed to model complex systems with a temporal structure which are ubiquitous in reality. There is a tradeoff between the complexity that is added when trying to analyse a network that has both topological and temporal network structure and the loss incurred when collapsing the time-dependence to construct a static network [22]. Therefore, a system should have suitable temporal structure in order for a temporal network approach to be valuable. To understand what this structure could be, let us consider some examples. As mentioned above, networks where the occurence of connections or node behaviour does not follow Poisson statistics could benefit from a temporal network approach. This is the case in e-mail and online communication networks [4,24], where bursty behaviour can occur whereby people send e-mails or communicate online for a certain relatively short period of time and then become inactive for some period of time, and so on. Another example in which it is important to consider temporal patterns is epidemic dynamics, where individuals staying at home because they have contacted a disease correspond to nodes in the network becoming inactive, and we would expect this behaviour to influence the contagion of the disease. Whenever the interactions between nodes or the activity of nodes in time are not too random or too regular, temporal network modelling could prove particularly useful. In general, however, we would like to model the temporal patterns of any system that has time-dependence, and this is the purpose of studying temporal networks.

A possible representation of a temporal network is $\mathcal{G}(\mathcal{S}, \mathcal{E}, \mathcal{T})$, where $\mathcal{T}$ is the set of all sets $\mathcal{T}_\varepsilon = \{(t_1, t_1'), \ldots, (t_n, t_n')\}$, which is the set of intervals over which an edge $\varepsilon \in \mathcal{E}$ is active, for all M edges. As before, the set $\mathcal{S}$ denotes the set of nodes and the set $\mathcal{E}$ denotes the set of edges. Most of the tools used in static networks have to be updated, for instance we would define an adjacency index

$$a(i, j, t) = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are connected at time } t \\ 0 & \text{otherwise.} \end{cases}$$

We could also define *time-respecting* paths as sequences of contacts with non-decreasing times (*i.e.* there are no paths to the past) that connect set of vertices. Using these paths, one can in principle determine which nodes can be reached from which other nodes within a time interval $[t_0, T]$. One can also define the *set of influence* of a

node i as the set of nodes that can be reached by time-respecting paths from node i. Such ideas generalise the concept of paths to account for temporal structure. We can also consider the *latency* (or *temporal distance* [25]) of two nodes, which is defined as the shortest time within which node i can reach node j through time-respecting paths. This extends the concept of distance by measuring time in addition to space. Other quantities can also be generalised [22], and they can be used in analysing the temporal structure of real networks drawn from empirical data. We expect that these ideas will prove useful in investigating social cascades in temporal networks starting from the temporal network framework and proceeding analytically, but we are not going to do this in this thesis. Instead, we are going to consider time-dependence in the sense of a dynamically changing network, as we will explain in Chapter 4.

3.2 Approach to Temporal Networks

3.2.1 General Considerations

Our aim is to investigate how the temporal structure of a network affects dynamical processes on the network and vice versa. In Chapter 4, we will do this by simulating social influence threshold models (specifically the Watts model with varying seed sizes) on random networks that evolve in time, and we will compare the spread of influence and the occurence or size of cascades with corresponding results from static networks. The feedback loop between dynamics of networks and dynamics on networks amounts to a temporal network structure which is directly dependent on the dynamical system on top of the network and not on an external activity pattern of nodes or edges. For example, someone might have a long-term connection to another person (*e.g.*, a family tie), independently of the dynamical process of social influence between him and other individuals. On the contrary, a friendship or acquaintance might be damaged or terminated because of a difference of opinion which might be caused by the spread of opinions on a social network, thus rendering an edge inactive. This would in turn affect the process of opinion formation, and so on.

3.2.2 How Should One Model Temporal Networks?

Modelling temporal networks entails expressing the temporal structure of a network. One might attempt to do this by considering various temporal properties of a network, such as the aforementioned concepts of temporal distance or time-respecting paths. Alternatively, one might attempt to replicate some temporal behaviour of

nodes and edges encountered in real networks from scratch, *i.e.*, by constructing a time-dependent random network. These approaches can then be used to explain the emergence or describe the function of network characteristics encountered in real temporal networks, or to provide the synthetic networks on which various dynamical processes take place, and investigate these systems numerically. We are interested in modelling cascade dynamics in social networks and we will thus have the latter modelling aim in mind. Although the process of modelling temporal networks is still in its infancy [22], we will briefly describe some contributions that have been made before considering the specific characteristics of social networks that can provide some insight on the mechanisms that we will propose to model this temporal network structure in Chapter 4.

One approach is to start with N nodes and build a network from scratch according to some stochastic rule that drives both network topology, edge weights, and the activation pattern of nodes and edges, which determines the temporal structure of the network. This could be done in conjuction with a dynamical process running on a network, as in a model proposed for generating temporal graphs to model sexual contact networks and sexually transmitted diseases [14], where we start with N nodes and follow the following rules [22]:

1. Create edges in the following way:
   (a) Form an edge with probability ρ between nodes chosen according to a mixing function ϕ:
      i. Choose two nodes i and j uniformly at random;
      ii. Decide whether they can form an edge according to $\phi(i, j)$;
      iii. If yes, done; else go to step i.
   (b) Repeat the step 1(a) $N/2 - P$ number of times.
2. In every edge between susceptible and infected nodes, transmit the disease with probability η.
3. Break every edge with probability σ.

The function $\phi(i, j)$ is used to generate mixing by degree in the network, accounting for assortative and disassortative mixing, which has been found to be an important property of sexual contact networks [14]. In this case, mixing by degree means that, for the assortative case for instance, individuals with many sexual contacts are more likely to have sexual contacts with other individuals who have many sexual contacts

themselves, and likewise for individuals with few sexual contacts. In the case of assortative mixing, $\phi(i,j) = 1 - \xi + \xi \frac{k_i k_j}{k_{\max}^2}$, where ξ is a parameter measuring the strength of assortativity, and $k_{\max}$ is the maximum degree. Moreover, P is the number of pairs, as the population is divided into single nodes, and nodes belonging to pairs. Another class of models for temporal networks are *randomised reference models* [22]. These models are, in a way, extensions of the configuration model in a temporal sense. Instead of only assigning a degree sequence $\{k_i\}$, we also assign an event sequence of edge or node activation times and then construct a network. Furthermore, a configuration model with a Poisson degree distribution removes the topological correlations between nodes and edges and thus the connectivity in the network is random. In the same way, we might want to remove temporal correlations between nodes or edges to create a temporal network that would serve as a null model with no correlations. To do this, we randomise either contact times, nodes, edges, or combinations of the three. For example, the *randomised edges* model (RE) is the following [22] (note that $i \to j$ denotes an unweighted undirected edge):

1. Specify a degree sequence and a contact sequence and construct the network.
2. Go over all edges one by one.
3. For every edge $i \to j$, pick another edge $n \to m$.
4. Replace $i \to j$ and $n \to m$ by $i \to m$ and $j \to n$ with probability $\frac{1}{2}$. Otherwise, replace them by $i \to n$ and $j \to m$.
5. If there are any self-edges or multi-edges, created by step 4, then destroy them and go back to step 2.

Thus, the edges are randomised in terms of the contact-time distribution, while the initial contact times (*i.e.* the distribution itself) are kept constant. Note that this accounts for edges but not for nodes being activated and de-activated in time.

Another approach, as done in [7], is to consider empirical data drawn from real temporal networks, and assign the time stamps taken from these data to the nodes and edges of a random network. This method has the advantage that it considers realistic temporal patterns, but the disadvantage that it offers no explanation of how or why the temporal network structure might arise.

In the next chapter, we will use a different method in our numerical simulations to generate temporal patterns. We will start from a random network and consider different rules for creating, breaking, or rewiring edges. These rules then give rise to

contact time patterns as the network evolves. The way that this happens depends on the dynamical process being considered, and the temporal structure arises due to this process, while the changing topological and temporal structure in turn affects the evolution of the dynamics on top of the network. As we are considering cascades in social networks, we wish to determine how the evolving structure affects the cascade dynamics. Before that, however, we have to think about a real system and see what the aforementioned rules might be.

Chapter 4

Numerics

4.1 Watts Threshold Model with Network Dynamics

As previously discussed, we want to propose a new model, based on the Watts threshold model, that generates temporal network patterns. We present it in this section, starting from the Watts threshold model and considering different rules for the activation or deactivation of edges.

Consider the Watts threshold model, where nodes have states 0 or 1. Imagine we have a social communication network, where a viral marketing or other advertising campaign for a product is taking place. State 1 corresponds to someone who has bought the product, and state 0 corresponds to someone who has not. Any individual who bought the product instantly becomes an advertiser and tries to make anyone he is connected with buy the product as well. Whether a node adopts state 1 or not depends on the local threshold dynamics. We assume, however, that a node that has not adopted after an interaction with its neighbours has a probability of breaking the edge with any of its state-1 neighbours. This might correspond to an individual who becomes annoyed at people trying to sell him/her a product and breaks the connection between them. We could additionally assume that edges between two state 0 or two state 1 nodes have a probability of being rewired after the local interactions take place. This could correspond to ephemeral connections such as "face-to-face" encounters or online activity. On the contrary, any edges between a state-0 node and a state-1 node (*i.e.* 1-0 edges) are not broken or rewired unless the breaking is initiated by the state-0 node after not adopting state 1. One could imagine that this might happen due to advertisers choosing to retain the connection in order to sell a product, or in general due to an individual continuously urging someone to buy or use a product or

service. There are many extensions and modifications that one might consider, such as creating edges or assigning some memory function to nodes, but we start with this model as a starting point. We will denote the probability of breaking edges by τ_b and the probability of rewiring edges by τ_r, and turn to numerical investigations on random networks in the next section.

4.2 Formulation and Implementation of Numerical Models

Our aim is to investigate the model proposed in the preceeding section, where we combine the Watts threshold model to a dynamically evolving network structure, in order to examine cascades on temporal networks. We run simulations of this dynamical process using MATLAB (see Appendix A for the code), on configuration-model networks. First, however, we have to test our code against known results to establish its validity. To do this, we will try to replicate a figure from [34], using the original Watts model on a Poisson random graph. The algorithm for this model is given by Algorithm 1 in Chapter 2. As we will see, there are some issues with replicating this figure, therefore we will turn to [11], where the Watts threshold model is extended by considering various seed sizes instead of a single initial seed as in [34], and replicate one of their results. After succesfully doing this, we will consider a network evolving in time according to certain rules and we will investigate the dynamics by producing various plots and comparing them with the corresponding results in static networks.

We can implement Algorithm 1 numerically in a straightforward way. The large size of the networks involved (*e.g.* the network used in [11] is comprised of $N = 10^5$ nodes), however, make the implementation computationally expensive and thus require some consideration of how to optimise the code. The most computationally expensive parts of the code are the construction of the random network and the loops required to update the network according to the dynamics. The most important measure we can take that significantly reduces the computational complexity is vectorisation. For example, the degree of a node k_i can be computed by $k_i = \sum_{j=1}^{N} A_{ij}$, where A is the adjacency matrix given by Equation 1.1. Equivalently, we can compute the degrees for all the nodes and store them in a vector $\mathbf{k} = [k_1, \ldots, k_N]^T$ by multiplying the matrix A with the vector of ones $[1, \ldots, 1]^T$. After fully vectorising our code, however, most functions that we want to perform are quite computationally expensive. In particular, the construction of the network is still computationally

expensive, whereas updating the network itself requires us to find types of edges using the adjacency matrix, and rewire or break edges by changing the adjacency matrix itself. Therefore, the size of the network and the corresponding size of the adjacency matrix makes it computationally expensive to construct and update this matrix. This is why, in producing some of the later results, we will also reduce the size of the network, considering networks of $N = 10^4$ or $N = 10^3$ nodes.

Extending the numerical algorithm to account for varying seed sizes is trivial. To include edge breaking and rewiring mechanisms, however, we have to modify Algorithm 1 at the stage where node states are updated according to the threshold rule. We present pseudo-code for this modified algorithm where both edge breaking and edge rewiring is included in Algorithm 2.

Algorithm 2 Watts Threshold Model with Network Dynamics

procedure DynamicModel(N, z, R, τ_b, τ_r) % N: number of nodes. z: mean degree. R: uniform threshold, τ_b: breaking probability. τ_r: rewiring probability.
    Construct random graph % Poisson or configuration model
    Compute k_i for $i = 1, \ldots, N$ % compute the degrees of all the nodes
    Set all node states σ to $\sigma(i) = 0$ for $i = 1, \ldots, N$ % set all initial states to 0
    Choose a set of random nodes S and set $\sigma(j) = 1 \; \forall j \in S$ % set a fraction of nodes' state to 1
    Change state to 1 if $\sigma(i) = 0$ and if $\sum_{j \in \Gamma(i)} \sigma(j)/k_i \geq R$ for $i = 1, \ldots, N$ % update states according to threshold rule
    Save new states in a new vector σ_{new} % keeping the old states in σ
    while $\sigma \neq \sigma_{\text{new}}$ **do** % while new nodes have adopted
        Set $\sigma = \sigma_{\text{new}}$ % update the state vector
        Change state to 1 if $\sigma(i) = 0$ and if $\sum_{j \in \Gamma(i)} \sigma(j)/k_i \geq R$ for $i = 1, \ldots, N$ % update states according to threshold rule
        Break any 1-0 edge with probability τ_b
        Find all 1-1 and all 0-0 edges
        Re-wire a node at a random end of these edges to a randomly chosen node with probability τ_r
        Save new states in a new vector σ_{new} % keeping the old states in σ
    end while
end procedure

Here we consider breaking edges between state-1 nodes and nodes that have not adopted state 1 after interacting with their neighbours, *i.e.* any i such that $\sum_{j \in \Gamma(i)} \sigma(j)/k_i < R$, and rewiring edges between unadopted state-0 nodes as well (also rewiring 1-1 edges). We stop the algorithm when the system reaches a steady

state with respect to the states, although a further perturbation due to rewiring and breaking edges might cause the process to continue. Note that the whole network is updated at once (*i.e.*, we do not update each node or each edge at a time). This is a case of synchronous rather than asynchronous updating, both of which we briefly describe in the following section.

4.3 Synchronous and Asynchronous Updating

Synchronous and asynchronous updating refer to the way in which nodes and edges of a network are updated in time due to a dynamical process taking place on the network. We treat time either as discrete, in the case of synchronous updating, or continuous, in the case of asynchronous updating. In the synchronous case, the whole network is updated at some discrete time step, say t_n, and the next update happens at t_{n+1}. That is, we are considering a discrete dynamical system, as the temporal evolution of a quantity is expressed as an iterative equation $x(t_{n+1}) = F(x(t_n))$. Conversely, in asynchronous updating, time is divided into infinitesimal time intervals dt, and parts of the updating happen continuously, so one can write a differential equation for some quantity x, such as Equation 1.2. Numerically, this regime can be approximated by defining a time interval Δt, say $\Delta t = 1/N$, updating at each consecutive $\Delta t + \frac{i}{N}$, $i = 0, \ldots, N-1$.

The crucial difference is that for synchronous updating, at each t_n, each node behaves according to the dynamics but independently of the other nodes' dynamics, with no knowledge of what the other nodes are going to do [13]. Therefore, all the nodes are synchronised. Conversely, in the asynhronous case, nodes are updated according to some sequential update process that takes place between two time points t_n and t_{n+1}, with the time interval $t_{n+1} - t_n$ divided into steps of size Δt. At each step, a node is updated, and at the next step the node to be updated takes into account the update that just happened for the previous node. For example, nodes could be picked at random from the N nodes and updated, then placed back into the group of N nodes that the next node is picked from, and the process repeated N times [13]. There are other ways for asynchronous updating to happen, though. For instance, at each time step Δt, we can choose a node at random without replacement, *i.e.* the next node is chosen among the remaining $N-1$ nodes. In this way, all nodes are updated before the next time point t_{n+1} [15]. In another version of asynchronous updating, a fixed random order is assigned to the N nodes of the network. This is the order that the nodes are going to be chosen with, for every time point between

t_n and t_{n+1} [15]. These different asynchronous updatings have different effects on the evolution of the dynamical process at hand, although sometimes these effects are not closely examined. These and additional types of updating are more extensively studied in a paper regarding cellular automata [33] and in the modelling of stochastic processes, for example in the Kinetic Monte Carlo (KMC) method (often used in statistical physics), which is a type of asynchronous updating [3].

In our system of a social network with information or social influence spreading according to threshold dynamics, it is not intuitively clear which of the two updating schemes is more appropriate. It can depend on the relation between the speed of propagation of information about the states of neighbouring nodes and the speed of propagation of the dynamical process. If nodes are instantly or almost instantly aware of the states of other nodes, then it would be appropriate to model this by using asynchronous updating, where the update of a node affects the decision of another node an infinitesimal amount of time later. Otherwise, all nodes would be updated synchronously. In any case, it can be shown that, when considering the steady state of the system in the Watts threshold model and its extension using various seed sizes, both synchronous and asynchronous updating produce the same numerical results [11].

4.4 Numerical Results for Threshold Models on Static Networks

We start by attempting to replicate "Figure 2(b)" [34] using the Watts threshold model (Algorithm 1). The random network used is a Poisson (*i.e.*, Erdős Rényi) random graph (defined in Chapter 1). In Watts' original paper [34], the network is defined by considering N nodes and assigning an edge to any pair of nodes with probability $p = z/N$, where the mean degree z is given, and while different than the way we defined an Erdős Rényi random graph, they both have a resulting Poisson degree distribution, *i.e.* $p_k = \frac{z^k e^{-z}}{k!}$, therefore they are, in effect, equivalent. Thus, we use Algorithm 1 on a Poisson random graph of $N = 10^4$ nodes, and depict the phase diagram of mean degree z versus thresholds R in Figure 4.1. The term phase diagram, in this case, refers to two distinct phases of the system. One in which a cascade occurs, and one in which it does not.

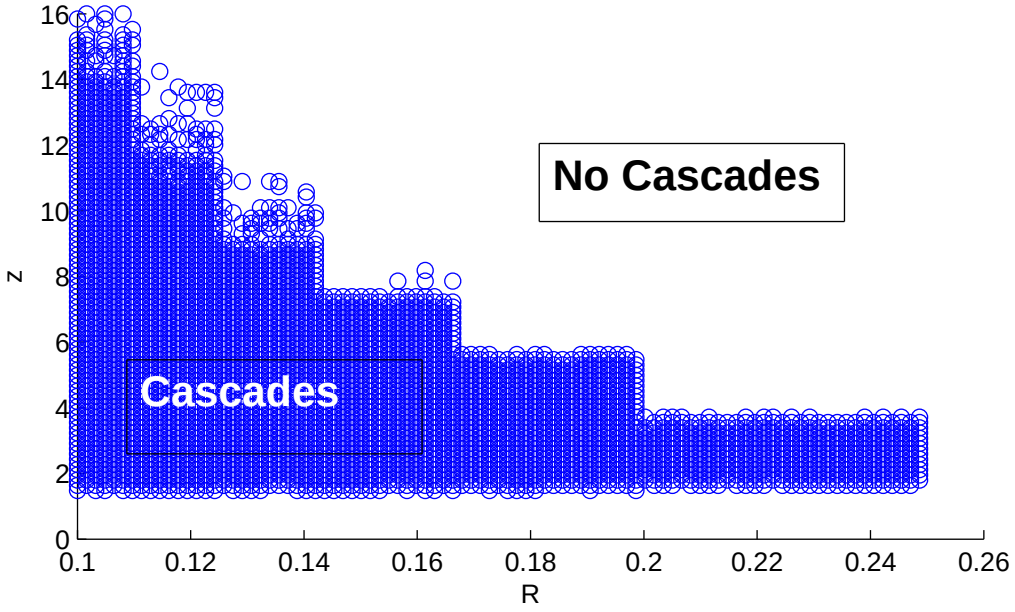

Figure 4.1: Phase diagram of mean degrees z versus thresholds R. The coloured area represents the range of values for which cascades occur, and the blank area the range for which they do not. The results are averaged over 100 realisations (different initial conditions, same network) and cascades are defined as events for which the ratio of state 1 nodes over all nodes is greater or equal to 0.3.

Although the results are very similar to those in "Figure 2(b)" in [34], the phase diagram is not exactly the same, so we have not replicated the results accurately. An issue we encountered in producing this figure has been the question of how exactly to define cascades in this context. Although global cascades can be intuitively defined as situations where the whole network has adopted, or in other words that the ratio of state 1 nodes over all network nodes is 1, a cascade in general means that a large enough fraction of the network has adopted due to a small initial perturbation. The problem is that an actual size was not provided in [34], nor was some other cascade condition specified. We tried different ratios for the cascade condition in order to replicate that particular figure, but none of the ones we used produced results that exactly matched those in [34]. There is the possibility that we did not try enough ratios, but to try all of them would be very computationally expensive. The natural question is whether there is something wrong with our code, so we turn to [11] and attempt to replicate a result therein using the same code, although now accounting for various initial seed sizes as well.

In [11], the authors extended the Watts threshold model to account for varying

seed sizes, where ρ_0 denotes the fracton of N nodes that are initially activated. They also denote the fraction of state-1 nodes over the whole network when the steady state has been reached by ρ. In this context, they derive analytical approximations for ρ in the infinite network limit $N \to \infty$. The Watts model and the results in [34] are then a limiting case of $\rho_0 \to 0$, as a single node that initially adopts corresponds to $\rho_0 = 1/N$, and $N \to \infty$. It is clear, therefore, that by replicating the simulation results of [11], we have replicated an extended version of the Watts threshold model, and thus our code succesfully applies the Watts model in general. We use the Poisson random graph once more, and we use the configuration model with a Poisson degree distribution to generate it. We place seed nodes at random, as mentioned in Algorithm 2 (ignoring edge breaking and rewiring processes in that algorithm). In Figure 4.2, we replicate the figure from [11] labeled "FIG. 1", which depicts the cascade size ρ versus mean degree z, for a uniform threshold $R = 0.18$ and for a Poisson random graph with $N = 10^5$ nodes.

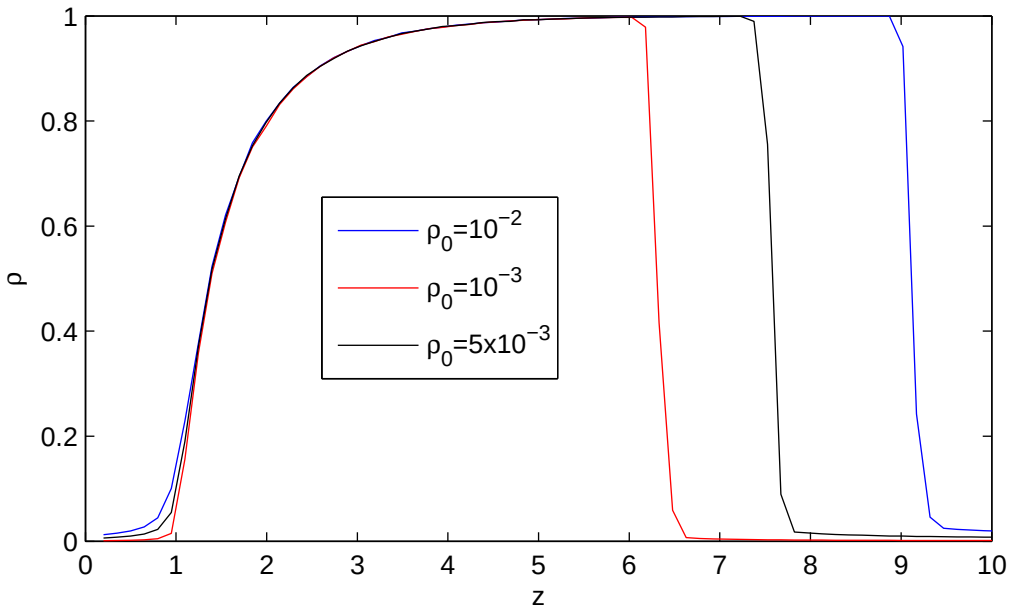

Figure 4.2: Mean density of adopted nodes (*i.e.* cascade size) ρ versus mean degree z. We depict three different seed sizes: $\rho_0 = 10^{-3}$ (red), $\rho_0 = 5 \cdot 10^{-3}$ (black), and $\rho_0 = 10^{-2}$ (blue). The uniform threshold is $R = 0.18$. The results are averaged over 100 realisations of the dynamics (different initial seed) on a Poisson random graph with $N = 10^5$ nodes.

The results completely agree with those in [11]. Note that ρ is very sensitive to the initial seed size ρ_0 and that as z increases, there is a sudden drop in ρ which

appears to be discontinuous. Conversely, there appears to be a continuous transition between a regime in which there are no cascades when the mean degree is low and ρ reaching the value of 1, meaning that global cascades occur and the cascade size is 1, as z increases. We see that the larger the number of seed nodes, the higher the value of z at which the discontinuous transition occurs. We can use the reasoning provided in Chapter 2 (and by Watts in [34]) that offered a possible explanation to a figure in [34], in order to gain some intuitive understanding of why this happens. As the mean degree gets higher, some nodes with many neighbours become isolated from the spreading process as they never overcome their threshold. As this happens to more and more nodes, the process has a tendency to die out. There are some events, however, in which the seed nodes are well-placed enough that the process spreads; due to the high mean degree, it spreads to the whole network causing a global cascade. As the seed size gets larger, the probability that the seed nodes are well-placed gets larger, and the probability of these events occuring gets larger as well. At some point, however, for some higher z, they are so rare that the spreading process dies out completely, and this is the transition point.

To compare the speed of the cascades, let us consider Figure 4.3.

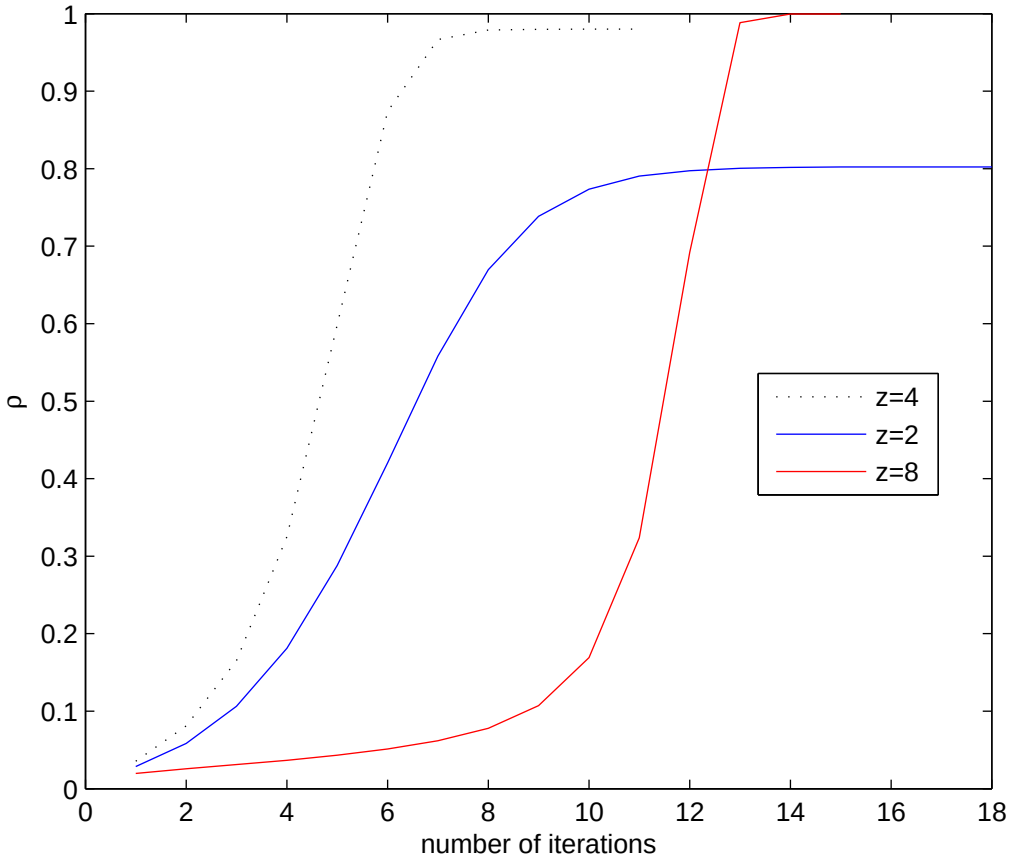

Figure 4.3: Fraction of adopted nodes ρ versus the number of iterations until steady-state (measuring time to steady-state). The seed size used is $\rho_0 = 10^{-2}$, *i.e.* 1,000 initial seed nodes in a network of $N = 10^5$ nodes. Plots are given for $z = 2$ (blue line), $z = 4$ (black dotted line), and $z = 8$ (red line). We note that the transition for this seed size occurs at about $z = 9$.

We can distinguish the speed of influence until steady-state for the three different mean degrees z we consider in Figure 4.3. The number of iterations that are needed is small for all three, so the process reaches steady state very quickly in all cases. However, we can see that for $z = 4$, which is right after the continuous transition to global cascades, the speed of contagion is the highest, while when $z = 2$, *i.e.* on the continuous transition itself, the speed of contagion is slower (if only slightly). Therefore, we see that the triggering of global cascades increases the speed from the level that it is during the transition, where state 1 spreads to a significant portion of the networks and then stops spreading. The high value of the mean degree in the case of $z = 8$ causes a slower speed of cascading, because the state-1 nodes first need to

slowly "reach" a critical portion of the network, which happens at 9 or 10 iterations, and then the whole network (ρ reaches 1 in this case contrary to the $z = 4$ case) adopts. Note that when this happens, the red curve has almost the same slope as the black dotted line, which suggests that the acceleration of propagation at that portion is very similar.

In Figure 4.4, we repeat the same calculations but with an initial seed size of $\rho_0 = 10^{-3}$, corresponding to 100 initial seed nodes, to examine how having less initial seed nodes affects the speed of propagation.

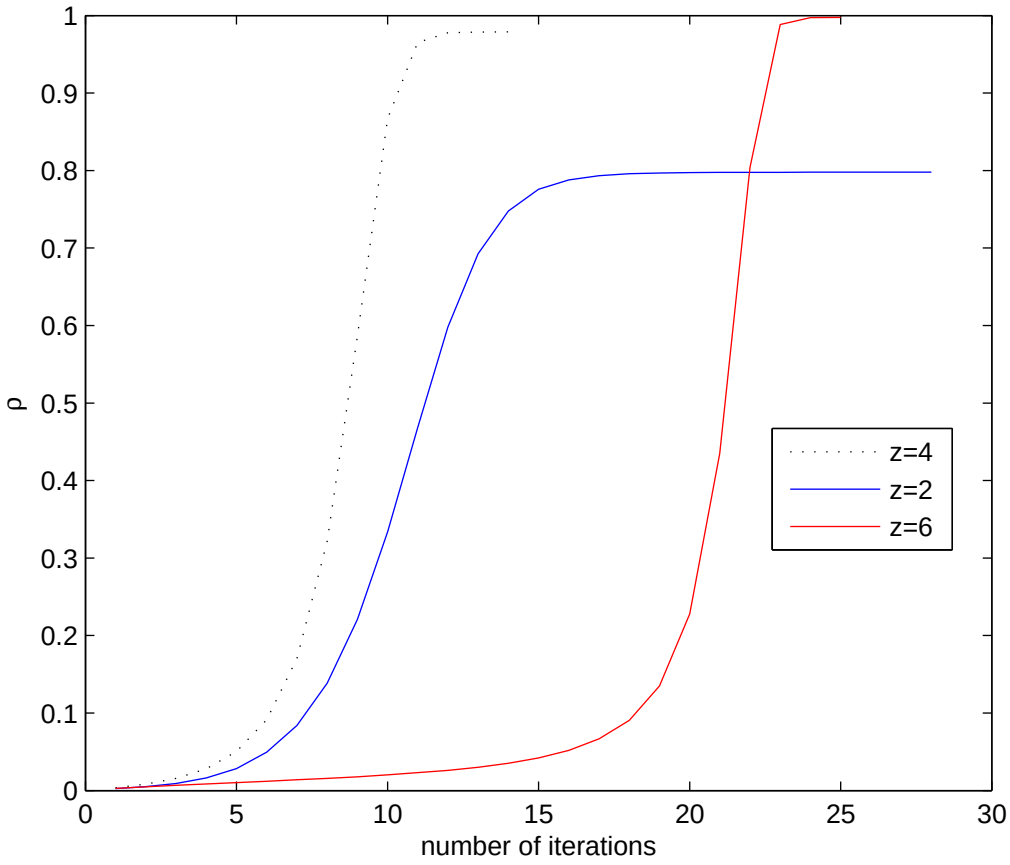

Figure 4.4: Fraction of adopted nodes ρ versus the number of iterations until steady-state for $\rho_0 = 10^{-3}$ ($N = 10^5$). We give plots for $z = 2$ (blue line), $z = 4$ (black dotted line), and $z = 6$ (red line). We note that the transition for this seed size occurs at about $z = 6.5$.

We observe a very similar pattern, but we can see that all three cases need more time to reach cascades or the steady state. This makes sense, as a smaller number of

initial seed nodes than before are trying to influence a network of the same size.

Having established a working code and having investigated some aspects of threshold dynamics on static networks, we consider dynamically evolving networks and the effect they have on the dynamical process (and vice versa).

4.5 Numerical Results for Threshold Models on Dynamic Networks

We consider the edge-breaking and rewiring rules for updating network structure in time as described in Section 4.1. We will use Algorithm 2 in our numerical simulations, use a configuration model with a Poisson degree distribution as the random graph on which the dynamical process occurs, and choose to use some combination of the edge-breaking and the edge rewiring mechanisms to examine the dynamics numerically.

The main result we are going to present is given in Figure 4.5, where we have computed the fraction of adopted nodes ρ versus the mean degree z for various values of the breaking probability τ_b (in particular, $\tau_b = 0, 0.1, 0.2, \ldots, 1$). The aim is to see how the results in Figure 4.2 change due to the breaking of edges, and examine how this mechanism might be affecting the contagion process.

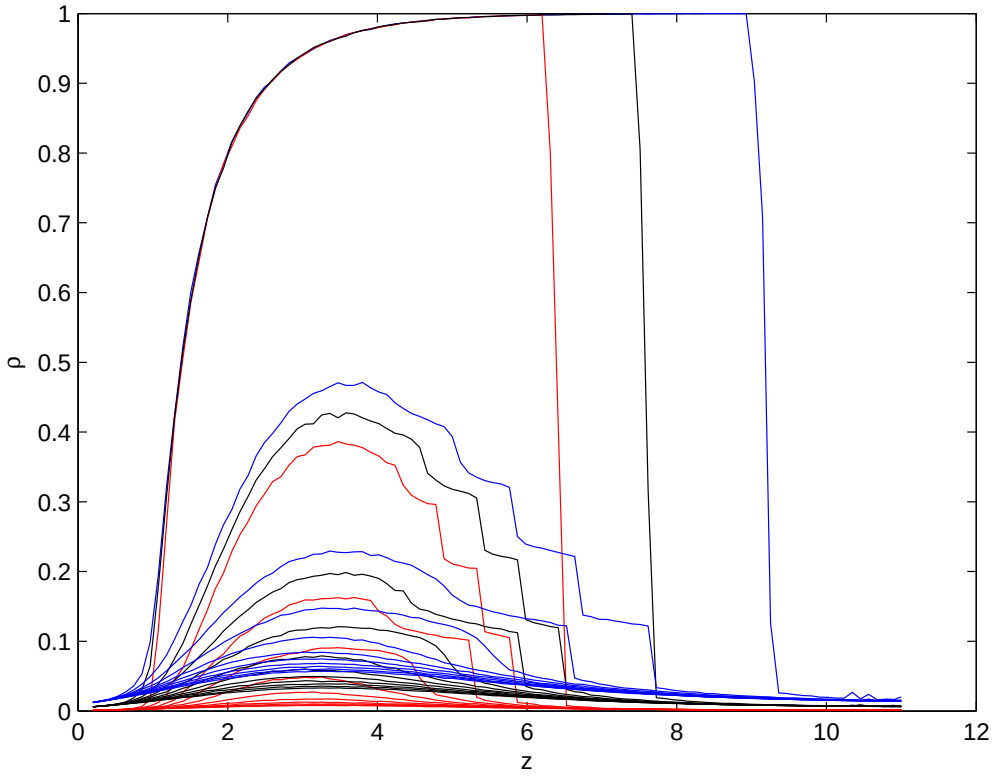

Figure 4.5: Fraction of adopted nodes ρ versus mean degree z with $N = 10^5$ and $R = 0.18$. The seed sizes are $\rho_0 = 10^{-3}$ (red lines), $\rho_0 = 5 \cdot 10^{-3}$ (black lines), and $\rho_0 = 10^{-2}$ (blue lines). We plot results for breaking probabilities $\tau_b = 0, 0.1, 0.2, \ldots, 1$. The three curves reaching a cascade size close or equal to 1 correspond to $\tau_b = 0$, the three curves reaching a lower cascade size (around 0.45) correspond to $\tau_b = 0.1$, and so on.

Above, we see the results for $\tau_b = 0, 0.1, 0.2, \ldots, 1$. The case of $\tau_b = 0$ is the same as and coincides with Figure 4.2, as required by construction. Each triplet of coloured curves corresponds to another breaking probability, with τ_b increasing as the curves are "moving" downwards. The breaking of 1-0 edges, where the node with state 0 did not adopt in the previous time step, severely restricts the spread of state 1, causing the cascade size to fall. As we will see shortly, every node in a network can reach state 1 even with edge breaking, assuming that the network size is relatively small and the initial seed size is larger than what we have been considering thus far. This, however, would not be called a cascade, since a relatively large portion of the network would initially have state 1 instead of a small seed that triggers a global cascade. Returning to Figure 4.5, we see that the behaviour of the curves as they

decrease towards the equilibrium $\rho \approx \rho_0$ is interesting, as, instead of displaying a discontinuous transition for high z as in the static case, they seem to instead display a series of discontinuous transitions. We take a closer look in Figure 4.6, where we depict the curve corresponding to $\rho_0 = 10^{-2}$ (blue) and $\tau_b = 0.1$.

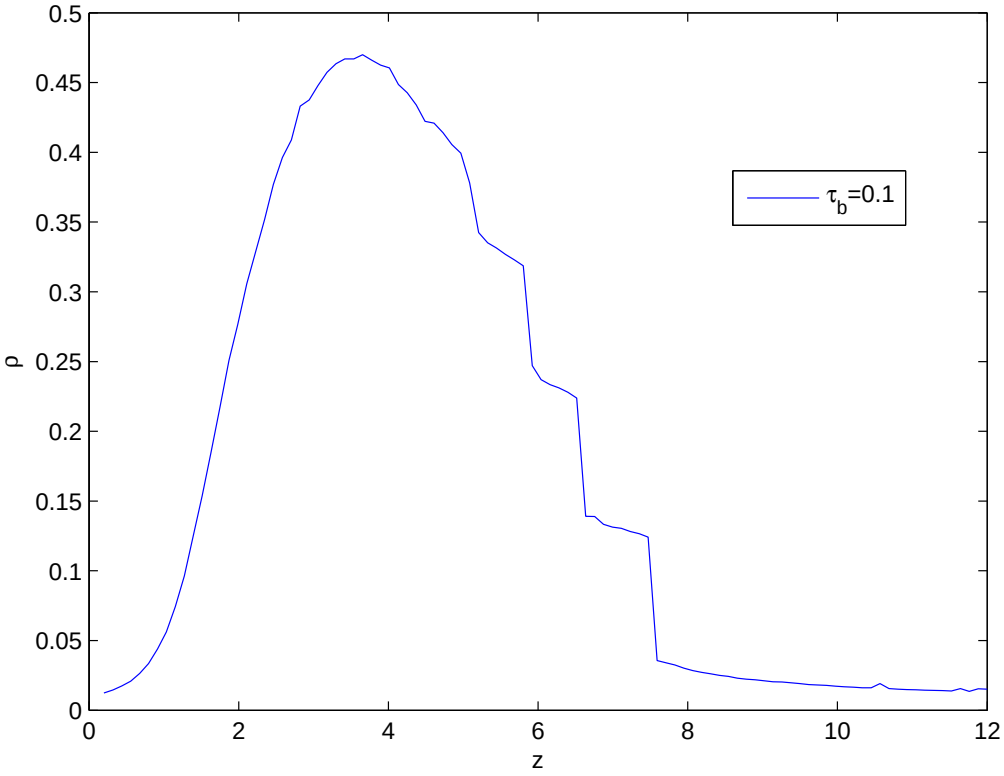

Figure 4.6: The curve corresponding to $\tau_b = 0.1$ and $\rho_0 = 10^{-2}$ from Figure 4.5.

Here we can see more clearly the discontinuous jumps that occur as z increases. The reasoning that we used to explain the discontinuous transition in Figure 4.2, suggests a breaking mechanism in which nodes become isolated, in the sense of being unable to change their state due to the local threshold dynamics, more quickly (*i.e.* for lower connectivities), thus moving the overall transition to $\rho = \rho_0$ to the left. The fact that these transitions happen one after the other might be explained in the following way. As the edges that are being broken are 1-0 edges, this causes the state-1 nodes of those edges to lose connections with neighbouring nodes and thus they become isolated. This reduces the spread of contagion, but only locally, for those neighbourhoods where the state-1 nodes have had their links with other state-0

nodes broken. Therefore, the fraction of adopted nodes ρ decreases but does not immediately drop to ρ_0; instead, it only drops to a lower level. As the connectivity increases further and the local threshold conditions become harder to overcome for state 0 nodes, the breaking mechanism isolates even more state 1 nodes, resulting in even reduced contagion, with a sudden drop happening at some critical z value. This process continues until finally ρ drops to ρ_0. The question that remains, however, is why these drops have to be discontinuous, and how is this related to the manner in which the breaking mechanism alters the network topology? Let us present some analytical results from [11] to suggest a possible research direction. Note that these results are taken straight out of [11] and we do not duplicate their derivation but only present them due to their result to the aforementioned problem.

In [11], considering the Watts threshold model with varying seed sizes ρ_0 on static networks, analytical results for the fraction of adopted nodes ρ have been derived and the following approximate equation for ρ was found:

$$\rho = \rho_0 + (1-\rho_0)\sum_{k=1}^{\infty} p_k \sum_{m=0}^{k} \binom{k}{m} q_\infty^m (1-q_\infty)^{k-m} F\left(\frac{m}{k}\right), \tag{4.1}$$

where $F(m/k)$ is the response function for the Watts model given by Equation 2.1, and q_∞ is the fixed point of the equation

$$q_{n+1} = \rho_0 + (1-\rho_0)G(q_n) \quad \text{for } n = 0, 1, 2, \ldots, \tag{4.2}$$

with $q_0 = \rho_0$, and G is a nonlinear function defined by

$$G(q) = \sum_{k=1}^{\infty} \frac{k}{z} p_k \sum_{m=0}^{k-1} \binom{k-1}{m} q^m (1-q)^{k-1-m} F\left(\frac{m}{k}\right). \tag{4.3}$$

What was found in [11], was that the fixed point q_∞ of Equation 4.2, has a discontinuous transition from $q_\infty \approx 1$ to $q_\infty \approx 0$ that drives the transition of ρ and is caused by a saddle-node bifurcation [29]. This suggests that there might be a series of similar saddle-node bifurcations that drive the transitions in Figure 4.6.

Lastly, we consider a simple effect of breaking and rewiring edges. In Figure 4.7, we consider a case in which edge breaking occurs with breaking probability $\tau_b = 0.1$ (without rewiring), a case in which edge rewiring occurs with rewiring probability $\tau_r = 0.1$ (but with no edge breaking), and a static case with neither rewiring nor breaking of edges. We then plot the fraction of adopted nodes ρ versus the number of iterations until a steady state is reached.

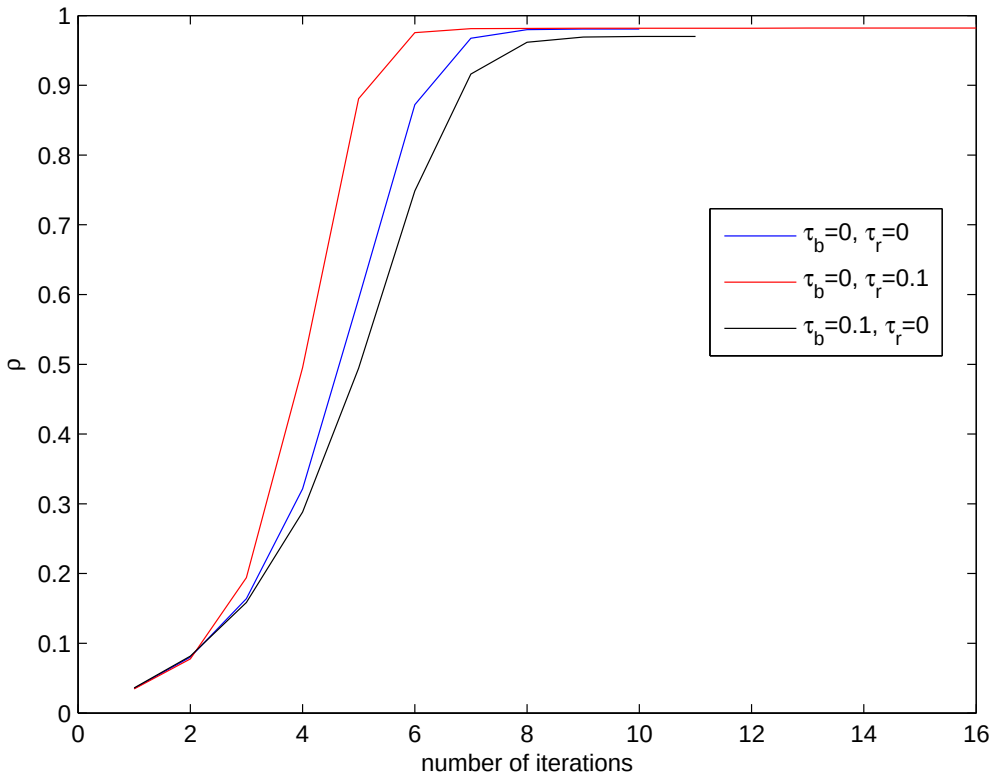

Figure 4.7: Fraction of adopted nodes ρ versus the number of iterations until steady state for $\rho_0 = 10^{-2}$ and $N = 10^4$. Only breaking with probability $\tau_b = 0.1$ (black line), only rewiring with probability $\tau_r = 0.1$ (red line), no breaking or rewiring (blue line). The uniform threshold used is $R = 0.18$ and the mean degree is $z = 6$.

We choose a smaller network of $N = 10^4$ edges, high connectivity ($z = 6$), and a large seed size $\rho_0 = 10^{-2}$, so that–even if edges are broken according to the breaking mechanism–a large portion of the network can be influenced by state 1. We see that edge breaking both reduces the speed of social contagion, and limits its size. Conversely, rewiring edges seems to speed up the process.

Chapter 5

Analytics

5.1 Motivation and Initial Theory

Most of the time when studying dynamical systems on networks, it is not possible in practice to derive exact analytical equations governing the evolution of the system we are considering. This is due to the complexity of the interactions in networks, as the interdependence and connectivity of nodes usually makes it impossible to keep track of all the effects a dynamical system has on a network and the effect that the network itself has on the dynamical system. Therefore, one has to look for analytical approximations of various kinds. Two common such approximating methods are mean-field theories and pair-approximation theories.

Mean-field theories were most prominently developed in statistical physics, where they are used in studying many-body problems [23]. If we consider a large number of particles, we pretend that are all in a bath, in which each particle interacts with the bath instead of interacting with all the other particles individually. The properties of this bath arise from the average quantities derived from all of the particles collectively, and each particle is coupled with other particles only via these average quantities. Similarly, if particles are replaced by nodes and interactions by edges, mean-field theories consider the interaction of nodes with the average state of the network as a whole. Pair-approximation theories are generalisations of mean-field theories that try to include pairwise dynamical interactions by considering the states of the nodes at the ends of an edge and the joint distributions of states of nodes in random edges [23].

There are, however, certain assumptions that need to hold in order for mean-field or pair-approximation theories to be valid (although, mean-field theories, for example, sometimes perform well even in systems where condition (1) below is violated [12] [27]). The typical assumptions used in deriving mean-field theories in networks are the following [12]:

1. *Absence of local clustering.* When changing the state of some node i, it is assumed that the states of its neighbours are independent of each other. This is true, for example, for locally tree-like networks, where if we look at the local neighbourhood of a node i and see that it is connected to nodes k and j, then nodes k and j are not connected to each other. In other words, the edges do not form triangles between neighbouring nodes (more precisely, the probability of having triangles tends to 0 as $N \to \infty$).

2. *Absence of modularity.* It is assumed that we can accurately describe nodes of degree k by the average state of k-degree nodes. This assumption is violated in the presence of modularity. For example, when a network has distinct communities, the k-degree nodes of a certain community might have different properties than the k-degree nodes of another community.

3. *Absence of dynamical correlations.* When updating the states of nodes, we assume that we can consider the state of a node i and the states of its neighbours to be independent from each other. This thus refers to dynamical correlations and not structural correlations that have to do with statistical network properties, such as degree-degree correlations.

The third assumption is relaxed in the case of pair-approximation theories. We do this by considering the states of nodes connected by edges and how they are dynamically correlated. We can see that the dynamical correlations of node states plays an important role in the Watts threshold model with network dynamics proposed in the previous chapter. As 1-0 edges have a probability of breaking, and 0-0 or 1-1 edges have a probability to be rewired, the relative states between two nodes are the essential characteristic that drives the network dynamics. Therefore, we would have to include dynamical correlations in an analytical approximation of the model, and we could use pair-approximation theories to do this. In pair-approximation theories, we usually express the dynamics in terms of fractions of pair of nodes in certain states [36]. To do this, one includes correlations of a higher-order, such as triplets of nodes in various states, and then try to approximate these higher-order correlations in terms of lower-order ones, *i.e.* using pairs of nodes in certain states or individual nodes and their states. In this sense, mean-field theories are a simplified version of such a procedure in which pairwise state correlations are approximated by individual node states using some averaging over the whole network.

We will now briefly present two simple examples, one of mean-field theory and one of pair-approximation theory. They will both be in the context of epidemiological

models, such as the SI or the SIR models. As we saw in Chapter 2, some such models can be formulated as threshold models as well.

5.2 Mean-Field Approximations: An Example

We follow the discussion in [1] and present an example where a mean-field theory approximation is used to derive a differential equation for a simple class of models of biological epidemics. Note that this is not our derivation but merely a duplication of the derivation in [1].

Consider a population of N individuals comprising a network on which a disease is spreading. Assume that the population can be divided into discrete compartments according to their state with respect to the disease. We could, for example, divide the population into susceptible S individuals, infectious I, or recovered R. We also assume that within every compartment, the individuals are well-mixed and homogeneous, while, if we take a node i, then the more neighbours in state I (infected) this node has, the higher the probability that it will get infected as well. To be as general as possible, consider compartments as classes denoted by $[m]$, and denote the number of individuals in class $[m]$ at time t by $X^{[m]}(t)$. As every individual in the population is in exactly one compartment, $N = \sum_m X^{[m]}(t)$. Note that the notation used here corresponds to the notation used in Chapter 2 in the discussion of SI and SIS models by considering two comparments, $X^{[h]} = S$ and $X^{[m]} = I$, and a rate of infection β that corresponds to v_h^m, a variable which will be defined below.

Consider the processes in which nodes can be transferred from one compartment to another. One class of such processes is the spontaneous transition of a node from class $[m]$ to class $[h]$. Such a transition can be written as

$$X^{[m]} \to X^{[m]} - 1 \tag{5.1}$$
$$X^{[h]} \to X^{[h]} + 1. \tag{5.2}$$

An example of such a transition is the spontaneous recovery of an infected individual ($I \to R$). We calculate the change in the number of individuals $X^{[m]}$ as

$$\partial_t X^{[m]} = \sum_h v_h^m a_h X^{[h]}, \tag{5.3}$$

where a_h is the rate of transition from class $[h]$, and $v_h^m = 1$, 0 or -1 is the change in the number of $X^{[m]}$ due to the spontaneous transition either from or to the class $[h]$.

A second process by which a transition can occur is the binary interactions between nodes, such as the contagion of a disease by a node in S from a node in I ($S+I \to 2I$). These are interactions that we will approximate in a mean-field theory sense. Consider an individual in class $[h]$. Assume that all nodes are in a homogeneous bath, so that

the probability of a node in class $[h]$ to interact with a node in class $[g]$ is the density $X^{[g]}/N$ of class $[g]$ individuals in the population. In this case, the change of $X^{[m]}$ is

$$\partial_t X^{[m]} = \sum_{h,g} v^m_{h,g} a_{h,g} N^{-1} X^{[h]} X^{[g]}, \tag{5.4}$$

where $a_{h,g}$ is the transition rate and $v^m_{h,g}$ is the change in the number of $X^{[m]}$ due to the interaction. We can add the two equations to write the general reaction-rate equations for the mean number of individuals in class $[m]$:

$$\partial_t X^{[m]} = \sum_{h,g} v^m_{h,g} a_{h,g} N^{-1} X^{[h]} X^{[g]} + \sum_{h,g} v^m_{h,g} a_{h,g} N^{-1} X^{[h]} X^{[g]}, \tag{5.5}$$

where $X^{[m]}$ are continuous variables representing the mean number of individuals in class $[m]$. Assuming that the total number of individuals in the network remains constant, we can also write

$$\sum_m \partial_t X^{[m]} = 0. \tag{5.6}$$

This constitutes an example of the application of mean-field theory method to derive an approximate master equation for the temporal evolution of a dynamical system on a network.

5.3 Pair-Approximation: An Example

In this section, we present an example of a pair-approximation (following [23]). Note that we duly follow the discussion in [23], which in turn is a modification of the discussion in [21], and that this is not our derivation.

Consider the SI model as presented in Chapter 2, and add dynamical correlations to Equation 2.2. To do this, we first write

$$\begin{aligned} \dot{S}_i &= -\beta S_i \sum_j A_{ij} I_j = -\beta S_i \sum_j A_{ij}(1 - S_j) \\ \dot{I}_i &= \beta S_i \sum_j A_{ij} I_j = \beta(1 - I_i) \sum_j A_{ij} I_j. \end{aligned} \tag{5.7}$$

This ignores dynamical correlations between the states of node i and its neighbours. To add those, we write the quantities S_i and I_i explicitly as expected values (as that is what they are), or mean probabilities that a node i is susceptible and infected, respectively. That is, $S_i = \langle S_i \rangle$, $I_i = \langle I_i \rangle$, and $\langle S_i I_j \rangle$ is then the mean probability that i is susceptible and j is infected at the same time. We can take some dynamical correlations into account by writing the following analog of Equation 5.7:

$$\frac{d\langle S_i \rangle}{dt} = -\beta \sum_j A_{ij} \langle S_i I_j \rangle. \tag{5.8}$$

As an aside, a mean-field assumption could be made on this equation, namely that $\langle S_i I_j \rangle \approx \langle S_i \rangle \langle I_j \rangle$, which ignores the dynamical correlations and gives back Equation 2.2 from Equation 5.8. This is a *moment-closure* approximation.

As we don't yet have an expression for the temporal evolution of $\langle S_i I_j \rangle$, Equation 5.8 is not closed. To do this, we will need higher-order dynamical correlations. Thus, let us figure out the ways that $\langle S_i I_j \rangle$ can change in time.

1. To reach a state in which node i is susceptible and node j is infected, we could have started with both nodes i and j in a susceptible state.The interaction of j with its neighbours other than i might have then caused j to become infected. The total rate that this can happen is $\beta \sum_{k \neq i} A_{jk} \langle S_i S_j I_k \rangle$.

2. $\langle S_i I_j \rangle$ can decrease if i becomes infected. This can happen either due to its neighbour j or due to one of its other neighbours $l \neq j$. Adding these two contributions gives the total rate for $\langle S_i I_j \rangle$ to decrease: $\beta \sum_{l \neq j} A_{il} \langle I_l S_i I_j \rangle + \beta \langle S_i I_j \rangle$. The first term refers to the neighbours $l \neq j$ and the second term refers to node j infecting i.

Therefore, the rate of change of $\langle S_i I_j \rangle$ is

$$\frac{d\langle S_i I_j \rangle}{dt} = \beta \sum_{k \neq i} A_{jk} \langle S_i S_j I_k \rangle - \beta \sum_{l \neq j} A_{il} \langle I_l S_i I_j \rangle - \beta \langle S_i I_j \rangle. \tag{5.9}$$

The problem, however, is that now we do not know the temporal evolution of the 3-node correlations we now appear on the right-hand side. Therefore, we need to do a moment closure, and we do this by approximating these 3-node correlations. Just as mean-field considerations give a moment closure in 2-node correlations by approximating them in terms of single node states (by assuming independence), the pair-approximation method provides a moment closure in 3-node correlations by approximating them in terms of the states of individual nodes and of pairs of nodes.

To do this, we are going to use Bayes' theorem to express joint probabilities in terms of conditional probabilities. We have

$$\langle S_i S_j I_k \rangle = P(i, j \in S; k \in I) = P(i, j \in S) P(k \in I \| i, j \in S). \tag{5.10}$$

At this point, we will make a simplifying assumption: we assume that the state of node k is independent of the state of i. This would be actually true in the case in which any path between k and i goes through j, as then the state of k would depend only on the state of j, because it would have to interact with it before interacting with i anyway. We can then write

$$P(k \in I \| i, j \in S) \approx P(k \in I \| j \in S) = \frac{P(j \in S, k \in I)}{P(j \in S)} = \frac{\langle S_j I_k \rangle}{\langle S_j \rangle},$$

which implies that

$$\langle S_i S_j I_k \rangle \approx \frac{\langle S_i S_j \rangle \langle S_j I_k \rangle}{\langle S_j \rangle}. \tag{5.11}$$

Similarly,

$$\langle I_l S_i I_j \rangle \approx \frac{\langle I_l S_i \rangle \langle S_i I_j \rangle}{\langle S_i \rangle}. \tag{5.12}$$

Therefore, the approximation to the temporal evolution of $\langle S_i I_j \rangle$ is

$$\frac{d\langle S_i I_j \rangle}{dt} \approx \beta \frac{\langle S_i S_j \rangle}{\langle S_j \rangle} \sum_{k \neq i} A_{jk} \langle S_j I_k \rangle - \beta \frac{\langle S_i \rangle \langle I_j \rangle}{\langle S_i \rangle} \sum_{l \neq j} A_{il} \langle S_i I_j \rangle - \beta \langle S_i I_j \rangle. \tag{5.13}$$

In this way, we have reduced 3-node correlations to 2-node correlations, but we still have to express things in terms of $\langle S_i I_j \rangle$ instead of other 2-node correlations. We can write

$$\langle S_i S_j \rangle = \langle S_i (1 - I_j) \rangle = \langle S_i \rangle - \langle S_i I_j \rangle.$$

We then define $p_{ij} = P(j \in I \| i \in S) = \frac{P(i \in S, j \in I)}{P(i \in S)} = \frac{\langle S_i I_j \rangle}{\langle S_i \rangle}$, and write a differential equation for p_{ij}

$$\begin{aligned}
&\dot{p}_{ij} = \frac{d}{dt}\left(\frac{\langle S_i I_j \rangle}{\langle S_i \rangle}\right) = \\
&= \beta(1 - p_{ij}) \sum_{k \neq i} A_{jk} p_{jk} - \beta p_{ij} \sum_{l \neq j} A_{il} p_{il} - \beta p_{ij} + \beta p_{ij} \sum_{l} A_{il} p_{il} \\
&= \beta(1 - p_{ij})(-p_{ij} + \sum_{k \neq i} A_{jk} p_{jk}),
\end{aligned} \tag{5.14}$$

which allows us to write

$$\frac{d\langle S_i \rangle}{dt} = -\beta \langle S_i \rangle \sum_{j} A_{ij} p_{ij}. \tag{5.15}$$

Therefore, through this pair-approximation method, we have derived an approximate analytic equation for the dynamics, and Equation 5.15 can be solved to give a complete time-dependent approximate solution to the problem. We see here that the correlations between infected and susceptible nodes (*e.g.*, $\langle S_i I_j \rangle$) are important in deriving an approximate equation. In the model we have proposed in Chapter 4, the correlations between state-1 and state-0 nodes are the building blocks for the rules for changing the network and pair-approximations such as the above would take these correlations into account, thus providing a more accurate picture of the system.

5.4 General Considerations

There are tradeoffs between mean-field and pair-approximation methods on the problem at hand. In our case, as we are concerned with social-influence threshold dynamics on dynamically evolving networks, there are a few things we can consider about the methods. Because we have not been able to produce our own analytical approximations to the problem, we can only assume how well (or not well) these methods would fit, as well as make an educated guess based on the work we've done in trying to derive such approximations. Firstly, the pair-approximation method adds complexity by adding 3-node correlations and then trying to approximate them, while mean-field methods rely on simpler approximations to get rid of 2-node correlations. Therefore mean-field theories are simpler, while pair-approximations could be more accurate. Because we are considering breaking and rewiring edges, the states between two nodes connected by an edge are crucial to the problem, and they could constitute a building block of the dynamics if a pair-approximation method that could approximate the state-edge time evolution could be determined. We have failed to do this, but it seems like a natural way forward. However, a simpler mean-field assumption might provide a first approximation to the problem and reveal some analytic behaviour which would be valuable since analytics have been completely absent from the numerical investigations of our dynamic model.

Lastly, we present our attempt at deriving some analytical approximations. Consider the Watts threshold model with network dynamics presented in Chapter 4. Let N_0 denote the fraction of nodes in state 0, N_1 denote the fraction of nodes in state 1, N_{00} denote the fraction of edges where both nodes have state 0, N_{11} denote the fraction of 1-1 edges, and N_{01} denote the fraction of 0-1 edges. Our aim is to derive differential equations for each of these quantities, describing how these fractions of states change in time. We can describe these equations in the following way

$$
\begin{aligned}
\dot{N}_1 &= (0 \to 1), \\
\dot{N}_{00} &= -(00 \to 01) - (00 \to 10) - (00 \text{ edge breaking}) + (00 \text{ edge forming}), \\
\dot{N}_{01} &= -(01 \to 11) + (00 \to 01) + (01 \text{ edge forming}) - (01 \text{ edge breaking}), \\
\dot{N}_{11} &= (01 \to 11) + (10 \to 11) - (11 \text{ edge breaking}) + (11 \text{ edge forming}).
\end{aligned}
$$

Here, $(. \to .)$ denotes all the possible ways that nodes can change from one state to another (or from a pair of states to a different pair of states). For example, a way for the fraction of 0-0 edges to go down is that 0-0 edges become 0-1 edges, and all the ways this can be done is denoted by $(00 \to 01)$. The challenge is to determine and count all these ways, and derive expressions for the probabilities of these transitions.

Another set of quantities for which one could seek governing equations is the following. Consider $N_{0,k}$ to be the fraction of nodes in state 0 with degree k. Accordingly, consider $N_{1,k}$, $N_{00,k,j}$, $N_{01,k,j}$, and $N_{11,k,j}$, where, for example, $N_{11,k,j}$ is the fraction of edges connecting nodes with states 1 and 1 and degrees k and j, respectively. By explicitly including the degrees, we could more easily include the response functions. For example, let $S_{0,k}$ and $S_{1,k}$ be the set of nodes with state 0 and degree k and the set of nodes with state 1 and degree k, respectively. Then, considering asynchronous updating, we can express the change in $N_{1,k}$ as

$$\begin{aligned} &N_{1,k}(t+dt) = N_{1,k}(t) + P(S_{0,k} \to S_{1,k}) N_{0,k} dt \\ &= N_{1,k}(t) + \sum_{l=0}^{k} F(l,k) P(\text{state } 0; \text{degree } k; l \text{ state 1 neighbours}) N_{0,k} dt, \end{aligned}$$

where $F(l,k)$ is the response function given by Equation 2.1. Therefore

$$\dot{N}_{1,k} = \sum_{l=0}^{k} F(l,k) P(\text{state } 0; \text{degree } k; l \text{ state 1 neighbours}) N_{0,k}. \tag{5.16}$$

In the same way, one can express the change of $N_{00,k,j}$, $N_{01,k,j}$, and $N_{11,k,j}$ as well, but then one must include more complicated probabilities, as well as the edge breaking and edge rewiring mechanisms of our model.

Chapter 6

Conclusion

6.1 Summary

We have approached the topic of cascades on temporal networks mainly from a computational perspective. We started by introducing the field of network science in general, we explained some important concepts of network properties, examples and the formulation of dynamical systems on networks, as well as random graphs, namely the Erdős Rényi random graph and the configuration model. We then proceeded by describing various threshold models, explaining their usage. Drawing from examples of real networks and real dynamical processes, we considered the significance and the applications of such models. We then focused on the Watts threshold model and some of its extensions. We presented some of the analytical and numerical results of this model, in order to use it as the cornerstone on which to build and compare against in later stages of the thesis.

Afterwards, we concerned ourselves with describing the emerging field of temporal networks. We considered some of the various issues and questions related to modelling the temporal structure of such networks and the issues with combining both a dynamical system on top of the network, and the topology already present in static networks. To this effect, we mentioned different modelling techniques that have been proposed in the study of temporal networks, before settling on a numerical approach that aims to account for temporal patterns in the case of information cascades in evolving social networks, such as the viral marketing of products on an online communications network.

We continued by performing numerical simulations on random networks using MATLAB. We described algorithms for implementing the Watts threshold model and its extensions, including our dynamic model which combines threshold dynamics with edge activation patterns. We briefly considered the choice between synchronous and

asynchronous network updating, before presenting our numerical examples, noting any issues or insights where appropriate. We found that our breaking mechanism creates an interesting pattern of a series of seemingly discontinuous drops in the spread of information in relation to network connectivity, and we speculated on the reasons for this behaviour by using previous related work as a guideline.

Lastly, we considered two analytical approximating techniques, mainly mean-field theory and pair-approximation theory, and applied them on two specific examples drawn from a class of threshold models, namely models of biological epidemics. We derived analytical approximations for these examples, presented some of our own efforts at analysing our proposed model, and we briefly reflected on the advantages and disadvantages of mean-field and pair-approximation theories and how they relate to our main topic of information cascades on social networks.

6.2 Futher Work

This thesis leaves an incredible amount of further study to be desired. We have only scratched the surface of social influence spreading in temporal networks, and in general of dynamical systems applied on networks with an explicit temporal structure. Since this is a young and emerging field, there are already various directions that one can take, and this thesis probably provides even more directions for further work rather than proposing a certain direction to follow.

There is, especially, a need for analytical results that could accompany the model of coupling threshold models and dynamically changing network edges and provide some insight into the effects that the dynamic model has both on the spreading dynamics as well as on the network topological and temporal structures. In particular, it would be interesting to see some analytics that could explain the behaviour observed in Figure 4.6. A possible suggestion is to look for saddle-node bifurcations in an analog of Equation 4.2 for a network evolving according to the breaking rule proposed in this thesis. This is motivated by previous work [11], but we cannot rule out that the effect might be happening due to some completely different process.

Another direction of future work is the investigation of other approaches for modelling dynamical processes coupled with temporal networks, such as creating new edges, assigning a memory function to node states, considering a function that determines time intervals of edge activation, or, in general, looking at real temporal networks and the patterns in which the network changes in time for inspiration. It

has not been possible for us to follow any of these lines of thought in this thesis, but they seem like interesting directions for further work.

Appendix A

MATLAB Code

MATLAB code for constructing a random network according to the configuration model

```
function [Adj DEGREE_LIST]= FConfMod( Pk, P0, N )
% Generates random network according to its degree distribution
% Pk - degree distribution (1st element contains degree 1);
% P0 - fraction of 0-degree nodes in the network.

% Pk = Pk / ( sum(Pk)+P0) % Normalization: Pk is normalized to 1-P0 ...
...(i.e. sum(Pk) + P0 = 1);

Nk = round( N .* Pk); % number of nodes of degree k
[c,imax] = max(Nk); Nk(imax) = Nk(imax) + (N - sum(Nk) - round(N*P0));.
    ...%% adjust Nk so that sum(Nk) = N
kmax = length(Pk);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%...
...%%%%%%%
%% Need to fill NodePairs with node numbers according to Pk, ...
...then shuffle and split into two columns
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%...
...%%%%%%%%%
if mod(sum(Nk.*(1:kmax)'),2) % if the number of stubs is odd,...
...make it even
    Nk(imax) = Nk(imax)-1; % decrease peak nodes
    if ( imax + 1 < kmax )
        Nk(imax+1) = Nk(imax+1)+1; % increase number of nearest ...
        ...degree nodes
    else Nk(imax-1) = Nk(imax-1)+1; end % increase number of ...
...nearest degree nodes (other side)
end
%%
NodeN_cur = 0;
```

```
idx_cur = 1; % current index in NodePairs vector
NodePairs = -ones(sum(Nk.*(1:kmax)'),1); % allocate memory for ...
...NodePairs

for k = 1 : kmax % loop for all possible degrees
    draft  =  NodeN_cur+1 : NodeN_cur+Nk(k); % generate node numbers...
    ...for nodes of degree k
    NodeN_cur = NodeN_cur + Nk(k); % update current node counter
    stubs_deg_k = reshape( repmat(draft,k,1), 1, Nk(k)*k ); % create ..
    ...stubs for all nodes of degree k
    NodePairs( idx_cur : ( idx_cur + Nk(k)*k - 1 ) ) = stubs_deg_k; ...
    ...% insert them into half-edges array
    idx_cur = idx_cur + Nk(k)*k; % update index for insertion
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%...
...%%%%%%%%%%%%
%% Shuffling NodePairs before splitting into two columns
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%...
...%%%%%%%%%%%%
[ NodePairs, IX ] = shuffle_alt( NodePairs ); % shuffle the degree...
...vector kk
NodePairs = reshape( NodePairs, length(NodePairs) / 2, 2 ); % split ...
...into two equal columns
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%...
...%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%...
...%%%%%%%%%%%%

%% Remove self- and multi-connections
NodePairs ( find( NodePairs(:,1) == NodePairs(:,2) ), : ) = []; ...
    ...% remove self-connections
NodePairs = sort (NodePairs, 2);  % sort so that smaller node number...
...comes 1st in a pair
NodePairs = unique (NodePairs, 'rows'); % take only unique pairs ...
...(The resulting vector is sorted in ascending order (1st column))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%  ARTIFICIALLY INCLUDE RIGHT AMOUNT OF ZERO DEGREE NODES !!!!!
Nk0 = round(N*P0);
[Adj, NnodesTot] = FNodePairs2Adj ( NodePairs, Nk0);

%% This calculates a degree for each node %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DEGREE_LIST = sum(Adj);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

MATLAB code corresponding to Algorithm 1

```
clear
tic

clear
tic

N=10^5;
z=linspace(0.05,10,100);

rho0=[10^(-2),10^(-3), 5*10^(-3)]; % fraction of seeds wanted
Runiform=0.18;% assuming uniform thresholds here
for w=1:3
for l=1:100

p=z(l)/(N-1);
y=random('poiss',z(l),[N,1]);
unv = unique(y);
a = [unv histc(y,unv)];
a1=a(2:end,:);

Pk = a1(:,2) / sum(a(:,2));
A = FConfMod( Pk,1-sum(Pk), N);

degree=A*ones(N,1); % degree(i) is the degree of node i
actualz=mean(degree);
[deg,indx]=max(degree);
thresh=Runiform*ones(N,1); % thresh(i) is the threshold of node i
rho0N=floor(rho0(w)*N);

for q=1:100

v=zeros(N,1); % initialise state vector (v(i) is state of node i)
shuffledindices=randperm(N);
v(shuffledindices(1:rho0N))=ones(rho0N,1);
vuln_v=A*v>degree.*thresh; % sets vuln_v(i) to 1 if number ...
...of active neighbours exceeeds thresh(i)*degree(i)
new_v=max(vuln_v,v); % assumes always on, so cannot turn off nodes

while max(new_v~=v)>0 % while new nodes have just been activated...
  v=new_v;
  vuln_v=A*v>degree.*thresh;
  new_v=max(vuln_v,v);
end
```

```
num_on=sum(v);
frac_on(q)=(num_on/N);
frac_off(q)=(length(v)-sum(v));

end
Density(w,l)=mean(frac_on);
end
end
plot(z,Density(1,:),'b')
hold on;
plot(z,Density(2,:),'r')
hold on;
plot(z,Density(3,:),'g')
toc
```

MATLAB code corresponding to Algorithm 2

```
clear
tic

N=10^5;
z=linspace(0.2,12,100);
tau_b=0.1;
tau_r=0.4;
 %breaking probability is tau_b
 %creating probability is tau_r
rho0=10^(-2); % fraction of seeds wanted
Runiform=0.18;% assuming uniform thresholds here
Density=zeros(1,100);
for l=1:100

p=z(l)/(N-1);
y=random('poiss',z(l),[N,1]);
unv = unique(y);
a = [unv histc(y,unv)];
a1=a(2:end,:);

Pk = a1(:,2) / sum(a(:,2));
A = FConfMod( Pk,1-sum(Pk), N);

degree=A*ones(N,1); % degree(i) is the degree of node i
actualz=mean(degree);
[deg,indx]=max(degree);
thresh=Runiform*ones(N,1); % thresh(i) is the threshold of node i
rho0N=floor(rho0*N);
[left right]=find(A);
```

```
frac_on=zeros(100,1);
frac_off=zeros(100,1);
for q=1:100

v=zeros(N,1); % initialise state vector (v(i) is state of node i)
shuffledindices=randperm(N);
v(shuffledindices(1:rho0N))=ones(rho0N,1);
vuln_v=A*v>degree.*thresh; % sets vuln_v(i) to 1 if number of active...
...neighbours exceeeds thresh(i)*degree(i)
new_v=max(vuln_v,v); % assumes always on, so cannot turn off nodes


while max(new_v~=v)>0 % while new nodes have just been activated...
  v=new_v;
  vuln_v=A*v>degree.*thresh;
  new_v=max(vuln_v,v);
%     Update network according to breaking rule
    ju=find(new_v(left)+new_v(right) == 1);
    for k=1:length(ju)
        if (rand <= tau_b)
            B=sparse(left(ju(k)),right(ju(k)),1,N,N);
            A=A-B;
        end
    end
[left right]=find(A);

  %Update network according to re-wiring rule for 1-1 and 0-0 edges
  mu=find(new_v(left)-new_v(right) == 0);
  for k=1:length(mu)
      if (rand <= tau_r)
          right(mu(k))=randi(N);
      end
  end
[left right]=find(A);
end
toc
num_on=sum(v);
frac_on(q)=(num_on/N);
frac_off(q)=(length(v)-sum(v));

end
Density(1,l)=mean(frac_on);
end
plot(z,Density(1,:))
```

toc

Bibliography

[1] A. Vespignani A. Barrat, M. Barthélemy. *Dynamical Processes on Complex Networks*. Cambridge University Press, 2008.

[2] P. Cohen A. Galstyan. Cascading dynamics in modular networks. *Physical Review E*, **75**(036109).

[3] J.L. Lebowitz A.B. Bortz, M.H. Kalos. A new algorithm for monte carlo simulation of ising spin systems. *Journal of Computational Physics*, **17**(1):10–18.

[4] A. Barabási. The origin of bursts and heavy tails in human dynamics. *Nature*, **435**:207–211.

[5] A.R. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, **74**:47–97.

[6] M. Macy D. Centola. Complex contagions and the weakness of long ties. *American Journal of Sociology*, **113**(3):702–734.

[7] P. Holme F. Karimi. Threshold model of cascades in temporal networks. arXiv:1207.1206v1, 2012.

[8] J.P. Gleeson. Cascades on correlated and modular random networks. *Physical Review E*, **77**(046117):1.

[9] J.P. Gleeson. Cascades on correlated and modular random networks. *Physical Review E*, **77**(046117).

[10] S.R. White J.O. Kephart. Directed-graph epidemiological models of computer viruses. *Proc. 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 343–359.

[11] D.J. Cahalane J.P. Gleeson. Seed size strongly affects cascades on random networks. *Physical Review E*, **75**(056103).

[12] J.A. Ward M.A. Porter P.J. Mucha J.P. Gleeson, S. Melnik. Accuracy of mean-field theory for dynamics on real-world networks. *Physical Review E*, **85**(026106).

[13] M. Tomassini L. Luthi, M. Giacobini. Synchronous and asynchronous network evolution in a population of stubborn prisoners. *IEEE Symposium on Computational Intelligence and Games*.

[14] M. Morris M. Kretzschmar. Measures of concurrency in networks and the spread of infectious disease. *Mathematical Biosciences*, **133**:165–195.

[15] M.S. Capcarrere M. Sipper, M. Tomassini. Evolving asynchronous and scalable non-uniform cellular automata. *Proceedings of International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 67–71.

[16] E.M. Izhikevich E. Ott M.A. Porter, K. Josic. *Small-World Network*. http://www.scholarpedia.org/article/Small world.

[17] S. Milgram. The small world problem. *Psychology Today*, **1**(1):60–67.

[18] R. Wilson N. Biggs, E. Lloyd. *Graph Theory*. Oxford University Press, 1986.

[19] M.E.J. Newman. Mixing patterns in networks. *Physical Review E*, **67**(026126).

[20] M.E.J. Newman. The structure and function of complex networks. *SIAM Review*, **45**(2).

[21] M.E.J. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[22] J. Saramaki P. Holme. Temporal networks. arXiv:1108.1780v2, 2011.

[23] M.A. Porter. Networks course lecture notes. University of Oxford, 2012.

[24] M. Thelwall R. Lambiotte, M. Ausloos. Word statistics in blogs and rss feeds: Towards empirical universal evidence. *Journal of Infometrics*, **1**:277–286.

[25] J. Saramaki R.K. Pan. Path lengths, correlations, and centrality in temporal networks. *Physical Review E*, **84**(016105).

[26] J.P. Gleeson M.A. Porter S. Melnik, J.A. Ward. Multi-stage complex contagions. arXiv:1111.1596v1, 2011.

[27] M.A. Porter P.J. Mucha J.P. Gleeson S. Melnik, A. Hackett. The unreasonable effectiveness of tree-based theory for networks with clustering. *Physical Review E*, **83**(036112).

[28] R.F. Simmons. Synthetic language behavior. *Data Processing Management*, **5**(12):11–18.

[29] S.H. Strogatz. *Nonlinear Dynamics and Chaos*. Perseus Books, Cambridge, 1994.

[30] G. Paul H.E. Stanley S. Havlin S.V. Buldyrev, R. Parshani. Catastrophic cascade of failurs in interdependent networks. *Nature*, **464**:1025–1028.

[31] R. Lambiotte T. Hoffman, M.A. Porter. Generalized master equations for non-poisson dynamics on networks. arXiv:1112.3324v1, 2011.

[32] P. Holme T. Takaguchi, N. Masuda. Bursty communication patterns facilitate spreading in a threshold-based epidemic dynamics. arXiv:1206.2097v1, 2012.

[33] R.L. Buvel T.E. Ingerson. Structure in asynchronous cellular automata. *Physica D*, **10**:59–68.

[34] D.J. Watts. A simple model of global cascades on random networks. *PNAS*, **99**(9):5766–5771.

[35] J-C. Rochet X. Freixas, B.M. Parigi. Systemic risk, interbank relations, and liquidity privision by the central bank. *Journal of Money, Credit and Banking*, **32**(3):611–638.

[36] L. Kucera Z. Nikoloski, N. Deo. Correlation model of worm propagation on scale-free networks. *ComPlexUs*, **3**:169–182.