Universidade de Aveiro Departamento de Física,
2017

Emanuel
Sousa Vieira

Processos em cascata em redes complexas direcionadas

Cascade processes in directed complex networks

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Mestre em Engenharia Física, realizada sob a orientação científica de Alexander Goltsev, Professor do Departamento de Física da Universidade de Aveiro

o júri / the jury

presidente / president

MARGARIDA M. R. V. FACÃO
Professora Auxiliar da Universidade de Aveiro

vogais / examiners committee

ALEXANDER V. GOLTSEV
Investigador da Universidade de Aveiro (orientador)

FRANCISCO C. SANTOS
Professor associado do Instituto Superior Técnico, Universidade de Lisboa

agradecimentos / acknowledgements

I would like to thank my thesis supervisor Alexander Goltsev for his guidance, patience and motivation throughout the making of this work. I thank him for all the insightful conversations during the development of the thesis and the improvement of the text.

Thanks to my family for all the support throughout the years. This accomplishment would not have been possible without them.

resumo

Nesta tese estudamos analiticamente e numericamente o processo de *bootstrap percolation* em redes complexas direcionadas. Formulamos e analisamos o processo de *bootstrap percolation* em ambas redes com pesos e sem pesos e também estudamos um processo de *bootstrap percolation* baseado em probabilidades. O processo de *bootstrap percolation* considerado tem um parâmetro de ativação associado k onde um nó é ativado se tiver pelo menos k nó vizinhos ativos. Comparamos os nossos resultados com resultados analíticos e numéricos obtidos para redes complexas não direcionadas. Analisamos também como as propriedades topológicas dos componentes das redes complexas direcionadas, como o *giant strongly connected component* e a periferia, influenciam o processo de *bootstrap percolation*. Aplicamos a nossa teoria no estudo do processo de *bootstrap percolation* em redes complexas reais. Mostramos que a nossa teoria desenvolvida para redes complexas aleatórias e não correlacionadas está em bom acordo com simulações numéricas do processo de *bootstrap percolation* em redes complexas reais que são correlacionas e agrupadas.

abstract

In this thesis we study analytically and numerically the bootstrap percolation process in random uncorrelated directed complex networks. We formulate and analyze the bootstrap percolation process on both unweighted and weighted networks and also study a probability based percolation process. The considered percolation process has an associated activation threshold k where a node only gets active if it has at least k active neighboring nodes. We compare our results with analytical and numerical results obtained for undirected complex networks. We also analyze how topological properties of the directed network components, such as the giant strongly connected component and the periphery, influence on the bootstrap percolation process. We apply our theoretical approach for studying the bootstrap percolation on real complex networks. We show that our theoretical approach developed for the case of random uncorrelated directed networks is in a good agreement with numerical simulations of the bootstrap percolation process on real complex networks which actually are correlated and clustered.

TABLE OF CONTENTS

Page

List of Figures **xiii**

1 Introduction to complex networks **1**
1.1 Complex networks basics 3
1.2 Types of complex networks 4
1.2.1 Undirected networks 4
1.2.2 Directed networks 4
1.2.3 Weighted networks 5
1.3 Components 6
1.3.1 Undirected networks 6
1.3.2 Directed networks 6

2 Boostrap percolation in undirected complex networks **11**

3 Bootstrap percolation in directed complex networks **17**
3.1 Components 19

4 Bootstrap percolation in weighted directed networks **23**

5 Probability based boostrap percolation in directed complex networks **29**

6 Bootstrap percolation in real networks **31**
6.1 Twitter 31
6.2 Google+ 32
6.3 *Caenorhabditis elegans* neural network 33

7 Conclusion **35**

Bibliography **37**

List of Figures

Figure **Page**

1.1 An example of a complex network. The circles are the nodes and the connections between them are the edges. 3

1.2 An example of a directed complex network. The arrows indicate in which direction the flow of information is possible. 4

1.3 An example of weighted complex network. The thickness of the edges indicate represent the weights. 5

1.4 General structure of a undirected complex network. 6

1.5 Schematic structure of a directed network. 8

1.6 Relative component sizes versus the mean in-degree $\langle q_i \rangle$ in Erdős-Rényi graphs. . . . 9

2.1 Fraction of active nodes versus number of iterations (results of 200 simulations) of the boostrap percolation process in random undirected Erdős-Rényi graphs. Here the fraction of seeds is $f = 0.1$, the number of nodes $N = 10^4$, the mean degree $\langle q \rangle = 15$ and activation threshold $k = 2$. This figure ilustrates the cascade behaviour of the boostrap percolation process. 11

2.2 An example of a boostrap percolation process for $k = 2$. i) a random complex network. ii) some nodes are selected to be the seeds, here represented by the shaded circles. iii) first iteration of the percolation process, every node neighbouring at least $k = 2$ active nodes is activated. iv) second and last iteration of the percolation process, the process stops since there are no nodes that have at least $k = 2$ active neighbouring nodes. . . 12

2.3 The fraction of activated nodes S_a as function of the fraction of initial seeds f in random undirected networks of mean degree $\langle q \rangle = 5$ and activation threshold $k = 3$. Simulation data for $N = 10^4$ nodes. The theoretical result calculated from equation 2.1 is also represented (solid curve). 13

2.4 Contribution of each term of the boostrap percolation equation 2.1 for the fraction of total activated nodes S_a at the end of the process. The results of equation 2.1 is plotted (yellow squares) and the contribution of the seeds (black circles) and nodes activated through the avalanche process (blue triangles) 14

3.1 An example of a boostrap percolation process for $k = 2$ in a directed complex network. i) a random complex directed network. ii) some nodes are selected to be the seeds, which are represented by the shaded circles. iii) first iteration of the percolation process, every node with an incoming connection from at least $k = 2$ active nodes is activated. iv) second and last iteration of the percolation process, the process stops since there aren't any more nodes that have incoming connections from at least $k = 2$ active neighbouring nodes. 17

3.2 The fraction of activated nodes S_a versus the fraction of initial seeds f. Simulation data (circles) for $N = 10^4$ nodes and mean in-degree $\langle q_i \rangle = 4.5$. The theoretical result calculated by 3.1 is also plotted (solid curve). 18

3.3 Comparison of S_a between simulations of undirected Erdős-Rényi networks with $\langle q \rangle = 10$ (dashed curve) and directed Erdős-Rényi networks with $\langle q \rangle = 10$ (solid curve). The size of the network is $N = 10^4$ nodes. 18

3.4 Comparison of S between simulations of undirected Erdős-Rényi networks with $\langle q \rangle = 5$ (solid curve) and directed Erdős-Rényi networks with $\langle q \rangle = 10$ (circles). The size of the network is $N = 10^4$ nodes. 19

3.5 Fraction of active nodes S_a versus fraction of seeds f for Erdős-Rényi graphs for several mean degrees for $k = 2$. The seeds were only placed in the OUT. f is the fraction of nodes used as seeds in the OUT. The size of the network is $N = 10^4$ nodes. 20

3.6 Fraction of active nodes S_a versus fraction of seeds f for Erdős-Rényi graphs for several mean degrees for $k = 2$. The seeds were only placed in the IN. f is the fraction of nodes used as seeds in the IN. The size of the network is $N = 10^4$ nodes. 20

3.7 Fraction of active nodes S_a versus fraction of seeds f for Erdős-Rényi graphs for several mean degrees for $k = 2$. The seeds were only placed in the G_S. f is the fraction of nodes used as seeds in the G_S. The size of the network is $N = 10^4$ nodes. 21

4.1 Theoretical (solid curve) and simulation (circles) data for $N = 10^4$ nodes, mean in-degree $\langle q_i \rangle = 5$, and weights distribution $W \sim N(1, 0.04)$. 24

4.2 Simulation data for bootstrap percolation in unweighted directed networks (diamonds) and weighted directed networks (stars) with weight distribution $W \sim N(1, 0.04)$ for $N = 10^4$ nodes, mean in-degree $\langle q_i \rangle = 5$ and activation threshold $k = 3$. 25

4.3 $P(s)$ for 3 edges which weights follow a normal distribution $W \sim N(1, 0.04)$. 25

4.4 Simulation data for bootstrap percolation in random weighted directed complex networks for several weight standard deviations σ_w and mean weight $\langle w \rangle = 1$. Here, the mean in-degree $\langle q_i \rangle = 5$, the activation threshold $k = 3$ and $N = 10^4$ nodes. 26

4.5 Edges weight distribution for several weights standard deviations σ_w with mean weight $\langle w \rangle = 1$. 27

4.6 The distributions $P_s(s)$ of the sum of three weights $s = w_1 + w_2 + w_3$ with distributions $P(w)$ equal to the ones in figure 4.5. When the standard deviation σ_w of the weights increases, the mean sum of weights $\langle s \rangle$ increases. 27

5.1 Probability based percolation in Erdős-Rényi with probability $p = 0.7$, activation threshold $k = 3$ and mean average in-degree $\langle q_i \rangle = 5$. Both the theoretical results (solid line) given by 5.2 and simulation data (diamonds) for $N = 10^4$ nodes are plotted. In the simulations each node has at maximum $q_i - k + 1$ chances to become activated. 30

6.1 Boostrap percolation in a Twitter directed network with an activation threshold $k = 3$ and $N = 81305$ nodes (blue diamonds). The theoretical result calculated through equation 3.1 is also shown (black line). Both results are in a good agreement. 32

6.2 Boostrap percolation in a Google+ directed network with an activation threshold $k = 3$ and $N = 107614$ nodes (blue diamonds). The theoretical result calculated through equation 3.1 is also shown (black line). 33

6.3 Boostrap percolation in a *Caenorhabditis elegans* weighted directed network with an activation threshold $k = 8$ and $N = 297$ nodes (blue diamonds). The theoretical result calculated through equation 4.2 is also shown (black line). 34

CHAPTER 1

INTRODUCTION TO COMPLEX NETWORKS

A complex network (or a graph) is a set of items called nodes connected to each other by bonds called edges. In general, a real system composed by modules that interact with each other can be represented by a complex network. For example, the World Wide Web can be represented by a complex network [1], where each website is a node and the links in a website that direct you to another one can be represented by edges. In an ecological community, the interconnection of food chains can be represented by a complex network where [2] each species is a node which is connected by an edge to another species indicating whether it is its predator or prey. Airplane traffic can be represented as a complex network [3], where each airport is a node and the flight paths between each airport are the edges. The human brain can also be represented as a complex network [4] where each neuron is a node and the axons connecting the neurons body cells are the edges. Social networks can be represented as a complex networks too [5], where each person is a node and the friendship between two persons can be represented as a node.

Complex network theory is a powerful tool for analyzing the structure of systems. Knowing how nodes interact with each other in a network can help the user studying the network take a decision on how to act. For example if we want to maximize the spread of information in a network we may want to select the nodes that overall have more connections to the other ones. This spread of information where a node with a certain state called the *active* state successively activates the other neighbouring *unactive* nodes is called a cascade process. Examples of cascade processes include the spread of a particular type of information, like the viral phenomenon, in social networks such as Twitter and the activation of neurons in the human brain. Other example is the cascade failure which is a process by which a node successively triggers the failure of the other nodes neighboring nodes that depend on it, which may happen in power distribution systems and financial systems.

Some work was already done regarding cascade processes in complex networks such as by

Dorogovtsev et al. [6][7] and [8]. In this work we mainly analyze cascade processes in directed complex networks which is a topic that hasn't been formulated in the literature yet. Directed complex networks are a type of complex networks where the edges are directed, meaning that the information can only go in one direction, i.e., a node i with a outgoing connection to another node j may be able to communicate with the node j but the node j may not be able communicate with the node i. The cascade processes we analyze in this work have an important parameter k: a node will only become active if it has at least k active neighbouring nodes. We study the cascade process in complex networks through the process of boostrap percolation, in which, in the beginning of the cascade process, a fraction f of random nodes are chosen to be active. These nodes are called *seeds*.

Directed networks have fundamental differences in comparison with undirected networks. Due to the edge directness, in a directed network we can identify some sets of nodes, called components, with different topological properties. In the general case a network is composed by a giant strongly connected components which is the main component of the network. In this component every node has a path to every other node. In the periphery, every node that can reach the giant strongly connected component belongs to the IN component and every node that can be reached by the giant strongly connected component belongs to the OUT component. Furthermore we can identify tubes that connect the IN and OUT components, and tendrils which are connected to these components. The structure of directed networks will be discussed in detail in subsection 1.3.2 (see Figure 1.5). One objective of this thesis is to find out how the mentioned components affect the bootstrap percolation process. In real networks these components play a very important role in their functions. For example in a paper about global corporate control [9], the top economical actors, which control most of the economic power, are placed in the giant strongly connected component, mentioned as the core of the network. These powerful companies placed at the core control other companies placed in the OUT component.

In this work we also analyse the bootstrap percolation process in weighted directed complex networks where every edge has an associated weight [9] [10]. In unweighted networks the weight of every edge would be one. In this type of networks, during the percolation process, one node gets active if it has at least active neighboring nodes which combined edges' weights are larger than the activation threshold k. Following our study on these type of networks, we analyze a bootstrap percolation process where each each edge has a probability p of transmitting the activation signal.

This work is divided in several chapters. In this chapter we consider the basics of complex networks, the several types of complex networks that we analyze in this work and the components of each type of network. In chapter 2 we discuss the boostrap percolation process in undirected complex networks. In chapter 3 we analyze the boostrap percolation process in directed complex networks. In chapter 4 we study the boostrap percolation process in weighted directed complex networks. In chapter 5 we analyze a boostrap percolation process where the activation is based

on a probability. In chapter 6 we apply the theory of the previous chapters to real networks such as a Twitter and Google+ networks and the neural network of the worm *Caenorhabditis elegans*.

1.1 Complex networks basics

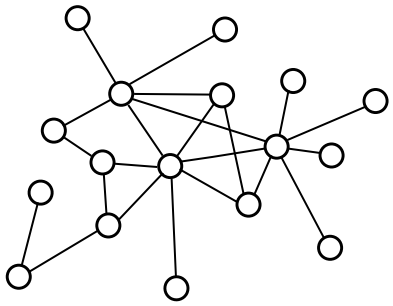

Figure 1.1: An example of a complex network. The circles are the nodes and the connections between them are the edges.

Each node can be characterized by the number of nodes to which it is connected to, this number is called the degree q of the node. Generally a network is composed by nodes of different degrees and to this distribution, we call it the degree distribution, which is denoted by $P(q)$. Complex networks in which the connections between nodes are attributed randomly are called Erdős-Rényi networks.

In Erdős-Rényi networks the degree distribution $P(q)$ follows a binomial distribution $B(N,p)$,

$$P(q) = \binom{N}{q} p^q (1-p)^{(N-q)} \tag{1.1}$$

where N is the number of nodes in the network and p the probability of the existence of an edge between two nodes. p is equal to $\frac{\langle q \rangle}{N}$ where $\langle q \rangle$ is the the mean degree of the network, $\langle q \rangle = \frac{1}{N}\sum_{i=1}^{N} q_i$.

In Erdős-Rényi networks with a large number of nodes N and low p, the degree distribution $P(q)$ can be approximated by the Poisson distribution with parameter $\lambda = Np = \langle q \rangle$,

$$P(q) = \frac{e^{-\langle q \rangle}\langle q \rangle^q}{q!} \tag{1.2}$$

A complex network with N nodes can be represented by a square matrix A, called the adjacency matrix,

$$A_{N,N} = \begin{bmatrix} 0 & a_{12} & \dots & a_{1N} \\ a_{21} & 0 & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \\ a_{N1} & a_{N2} & & 0 \end{bmatrix}$$

this matrix is symmetric with zeros on its diagonal. Each element a_{ij},

$$a_{ij} = a_{ji} = \begin{cases} 1, \text{ if nodes } i \text{ and } j \text{ are connected} \\ 0, \text{ if nodes } i \text{ and } j \text{ are not connected} \end{cases}$$

1.2 Types of complex networks

The types of complex networks studied in this work differ in the type of edges that connect the nodes, we analyze undirected networks, directed networks and weighted networks.

1.2.1 Undirected networks

Undirected networks are the most simple type of complex networks as well as being the most studied [11][12][13], from the evolution of scale-free networks [14] to the study of the bootstrap percolation in random networks [6]. These are the most basic form of networks and have common properties with other, more elaborated, types of networks. Bootstrap percolation theory of random undirected networks [6] will be used in this work as a basis to study the bootstrap percolation in other types of networks such as directed networks.

If two nodes are connected they can share information with each other. The edges in undirected networks are bidirectional, in contrast with the edges in directed networks.

1.2.2 Directed networks

The main type of networks that we are studying in this work are the directed complex networks. In this type of networks the edges only allow the transmission of information in one direction. Examples include the World Wide Web [1], neural networks (such as the human brain) [15], food chain networks [2] and many other real systems.

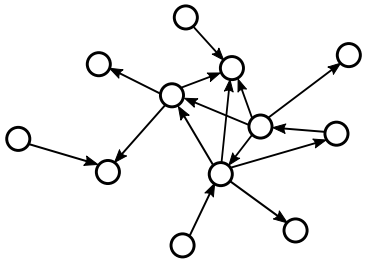

Figure 1.2: An example of a directed complex network. The arrows indicate in which direction the flow of information is possible.

Every node has two types of degrees: the in-degree q_i, which is the number of edges that are incoming to the node, and the out-degree q_o, which is the number of edges that are outgoing from the node. The in- and out- degree distributions can be different and we denote them by $P_i(q_i)$

and $P_o(q_o)$, respectively. In general the degree distribution in directed complex networks can be represented by the joint distribution of q_i and q_o, denoted by $P(q_i, q_o)$, where,

$$P_i(q_i) = \sum_{q_o} P(q_i, q_o) \quad \text{and} \quad P_o(q_o) = \sum_{q_i} P(q_i, q_o) \tag{1.3}$$

If P_i and P_o are uncorrelated, which is the case in some random networks such as Erdős-Rényi graphs, we have that,

$$P(q_i, q_o) = P_i(q_i) P_o(q_o) \tag{1.4}$$

To check if two variables are correlated such as the out and in-degrees we can calculate the Pearson linear correlation coefficient r,

$$r = \frac{\mathbf{E}[(q_o - \langle q_o \rangle)(q_i - \langle q_i \rangle)]}{\sigma_{q_o} \sigma_{q_i}} \tag{1.5}$$

where σ_{q_o} and σ_{q_i} are the standard deviations of the out-degree q_o and in-degree q_i and $\mathbf{E}$ is the expected value. r ranges between 1 meaning total positive correlation and -1 meaning total negative correlation. If $r \approx 0$, the two variables are not correlated.

In directed networks the mean degree $\langle q \rangle$ is the total mean degree. The mean in-degree $\langle q_i \rangle$ should be equal to the mean out-degree $\langle q_o \rangle$ since every edge is both outgoing from a node and incoming to another node. Therefore we have,

$$\langle q_i \rangle = \langle q_o \rangle = \frac{\langle q \rangle}{2} \tag{1.6}$$

The adjacency matrix of directed networks is not symmetric since a node i may not have both an incoming and an outgoing connection to another node j.

1.2.3 Weighted networks

In weighted networks, each edge as a different weight w. As examples, these weights may represent the trust between friends in social networks, the ammount of airplane traffic between airports [16] and the strenghts of the connections between neurons in the human brain [17].

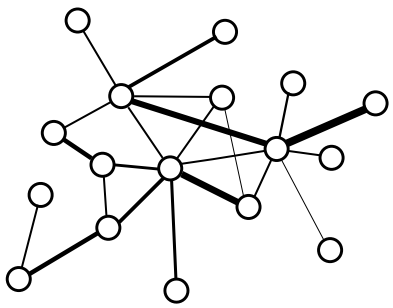

Figure 1.3: An example of weighted complex network. The thickness of the edges indicate represent the weights.

The weight probability density is represented by $P(w)$.

The elements of the adjacency matrix of a weighted network a_{ij},

$$a_{ij} = \begin{cases} \mathbb{R}_{>0} \sim P(w) \text{ , if nodes } i \text{ and } j \text{ are connected} \\ 0 \qquad\qquad\quad \text{, if nodes } i \text{ and } j \text{ are not connected} \end{cases}$$

A weighted network may have directed edges. In this work we also analyze the boostrap percolation process in random weighted directed networks.

1.3 Components

1.3.1 Undirected networks

An undirected complex network is generally composed by components or clusters which are sets of nodes that are joined together. The biggest cluster of nodes is called the giant component, G_S, if its size is a finite fraction of the whole networks (nonzero) and the other rest of nodes, which are not connected to the giant component are called the disconnected clusters, C.

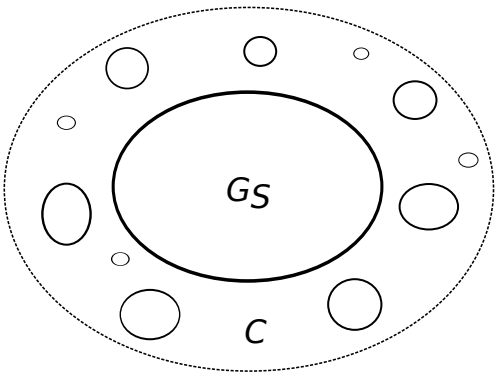

Figure 1.4: General structure of a undirected complex network.

In random uncorrelated random complex networks, the giant component almost surely exists if [18] ,

$$\sum_q q(q-2)P(q) > 0 \ . \tag{1.7}$$

1.3.2 Directed networks

A directed network $\mathscr{G}$ can be divided into several types of subnetworks called components [19]. We can identify,

- a giant strongly connected component ($GSCC$, or G_S for brevity) where all nodes can be reached from every other node following directed edges;

- a giant in-component (G_{in}) composed by G_S and all nodes that can reach G_S;
- a giant out-component (G_{out}) composed by G_S and all nodes that can be reached by G_S;
- tendrils (T), composed by nodes connected to the mentioned components but do not belong to none of them;
- a giant weakly connect component (G_W) composed by all the previously mentioned structures;
- and some disconnected clusters (C) which are not connected to the main network.

The general structure of a directed network is represented in figure 1.5. As G_{in} and G_{out} share some nodes with G_S there is a share and mixing of topological properties. We must separate these components from the G_S to more accurately study the role of each component. We can identify IN, as the set of nodes of G_{in} that do not belong to G_S.

$$G_{in} = G_S \cup IN \tag{1.8}$$

and OUT, as the set of nodes of G_{out} that do not belong to G_S,

$$G_{out} = G_S \cup OUT \quad . \tag{1.9}$$

All the tendrils T are connected to either the OUT and IN components. We call F to the union of the tendrils T and disconnected clusters C.

$$F = T \cup C \tag{1.10}$$

We have,

$$G_W = IN \cup OUT \cup G_S \cup T = \mathcal{G} \setminus C \tag{1.11}$$

The giant weak component G_W is the directed version of the giant component in an undirected network if we ignore the edge directness.

Edge directness influences the activation of the network. Knowing which nodes belong to each component helps deciding the optimal seed placement as to activate the most nodes possible.

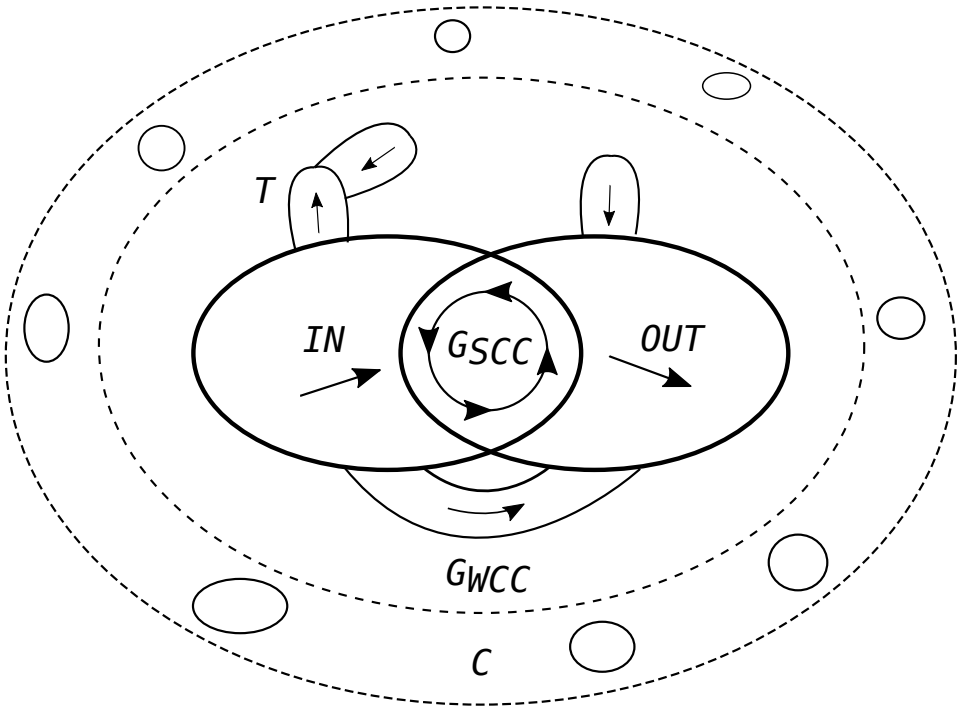

Figure 1.5: Schematic structure of a directed network.

To find the *OUT* and *IN* components one may use tree search algorithms such as the breath-first search or depth-first search. For the *OUT* component, the depth-first search (DFS) algorithm is,

1. Let V be an empty list of the visited nodes.
2. Start in any node v of the G_S.

   Let $i = 1$. While $i > 0$

   a) If F_i does not exist, create a list F_i made up from all the unvisited out-neighbors of v, add v to V.

   b) Delete v from all previous F_k, $k < i$.

   c) If F_i is not empty, let v be the first node of F_i and $i = i + 1$. Else, $i = i - 1$.
3. V is now composed by all the nodes of the G_{out}. To find the *OUT* component, remove all the nodes from V that also make part of the G_S.

To find the *IN* component we just need to invert all the edges of the network and run the algorithm again.

To find what nodes belong to the giant strongly connected component G_S one may use the Kosaraju algorithm [20]. Through this algorithm we can find out all the strongly connected

components, being G_S the largest. This algorithm is divided in two parts where in the first one we do a depth-first search and in the second we run another depth-first search but with the network reversed.

1. Let L be an empty list of the visited nodes.
2. Run a depth-first search in the network. Every time F_i is empty (in 2.c in the DFS algorithm) add v to L.
3. Reverse all the network edges. Let S be a list of sublists where each sublist will be a strongly connected component.
4. Let $i = 1$ (index of S). While size of $L > 0$,
   a) Run a DFS starting at the last node of L.
   b) Every node the DFS reaches add it to $S[i]$ and remove it from L.
   c) When the DFS ends, $i = i + 1$.
5. i indicates the number of strongly connected components found. The largest $S[i]$ is the giant strongly connected component, G_S.

To find the giant weakly strongly connected component G_{WCC} we just need to run DFS in the network while ignoring edge directness.

The size of each component strongly depends on the mean degree $\langle q \rangle$. In figure 1.6 some of the relative component sizes versus of the mean in-degree in Erdős-Rényi networks. As we can see, both *IN* and *OUT* reach their maximum size at around $\langle q_i \rangle \approx 1.35$, and for bigger $\langle q_i \rangle$ the size of these components decreases as the G_S begins to dominate in the network.

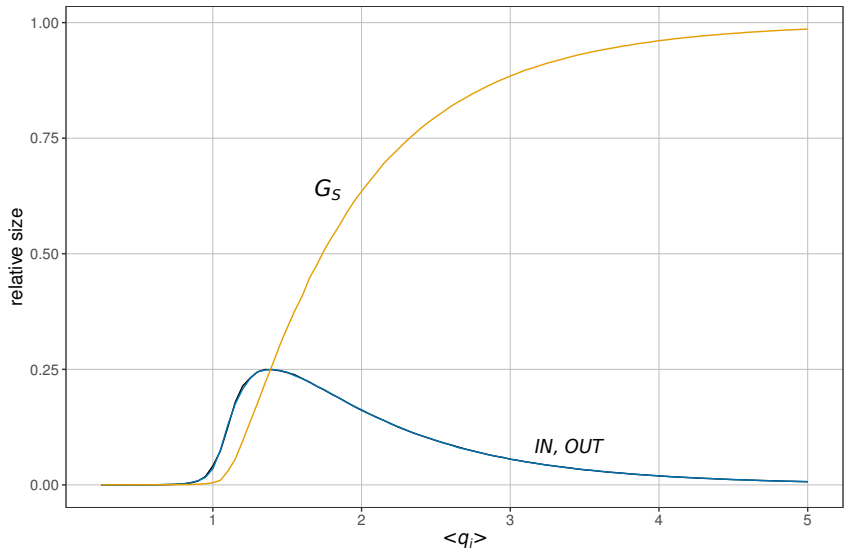

Figure 1.6: Relative component sizes versus the mean in-degree $\langle q_i \rangle$ in Erdős-Rényi graphs.

CHAPTER 2

BOOSTRAP PERCOLATION IN UNDIRECTED COMPLEX NETWORKS

Boostrap percolation is a process in which a random set of initial nodes called seeds *activate* their neighbouring nodes. These neighbouring nodes only get activated if a specific condition is met. In this work this condition is the number k of active nearest neighboring nodes that are necessary to activate a given node. The process reaches a final state when the maximum number of possible active nodes is reached.

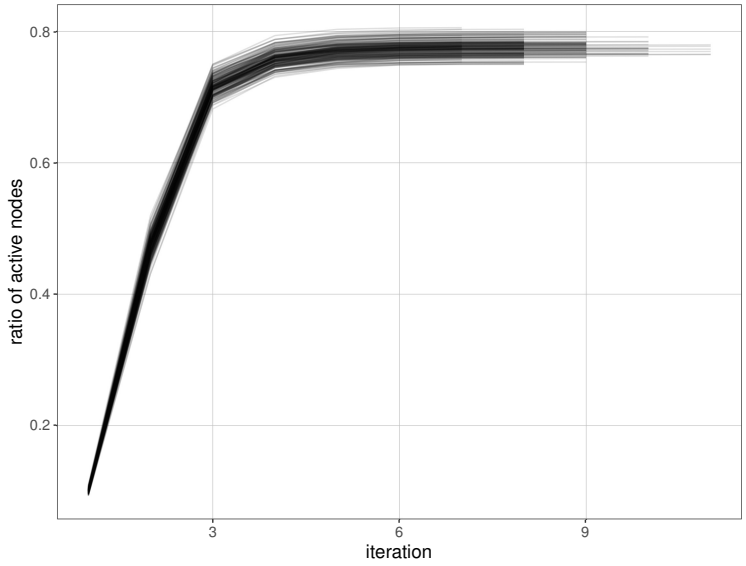

Figure 2.1: Fraction of active nodes versus number of iterations (results of 200 simulations) of the boostrap percolation process in random undirected Erdős-Rényi graphs. Here the fraction of seeds is $f = 0.1$, the number of nodes $N = 10^4$, the mean degree $\langle q \rangle = 15$ and activation threshold $k = 2$. This figure ilustrates the cascade behaviour of the boostrap percolation process.

It is necessary to mention that the cases $k = 1$, and $k \geq 2$ are very different. If $k = 1$ we consider that there is no activation threshold and a single seed gradually activates all nodes that belong to the same finite and giant cluster. On the other hand for $k \geq 2$ this cannot happen and the activation process is generally slower and can demonstrate a discontinuous behavior.

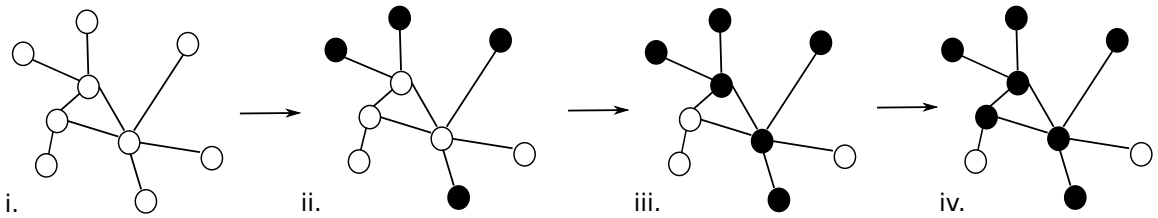

Figure 2.2: An example of a boostrap percolation process for $k = 2$. i) a random complex network. ii) some nodes are selected to be the seeds, here represented by the shaded circles. iii) first iteration of the percolation process, every node neighbouring at least $k = 2$ active nodes is activated. iv) second and last iteration of the percolation process, the process stops since there are no nodes that have at least $k = 2$ active neighbouring nodes.

We can find the fraction of activated nodes S_a at the end of the percolation process in dependence on the fraction of initially activated nodes f (seeds) [6],

$$S_a = f + (1-f)\sum_{q=k}^{\infty} P(q) \sum_{l=k}^{q} \binom{q}{l} Z^l (1-Z)^{q-l}, \tag{2.1}$$

where Z is the probability that following an arbitrary edge in the graph we reach a node that is a seed or has at least k neighbours that are active. In this equation, the first term f corresponds to the initially activated nodes (seeds). The second term describes the bootstrap percolation process. $\binom{q}{l} Z^l (1-Z)^{q-l}$ is the probability that a node with degree q has l active neighbours. Also we need to account for all the possible degrees when the activation is possible ($q \geq k$), where each node with degree q appears in the graph with a probability $P(q)$, that is given by the summation $\sum_{q=k}^{\infty} P(q)$,

$$Z = f + (1-f)\sum_{q=k}^{\infty} \frac{q+1}{\langle q \rangle} P(q+1) \sum_{l=k}^{q} \binom{q}{l} Z^l (1-Z)^{q-l} \quad . \tag{2.2}$$

The solution of this equation is exact in the infinite size, where an uncorrelated complex network as a tree-like structure and finite loops can be neglected. These loops are characteristic of correlated networks, normally with a non-negligible clustering coefficient.

For a Erdős-Rényi graph in the infinite size limit, the degree distribution $P(q)$ is the Poisson distribution,

$$P(q) = \frac{e^{-\langle q \rangle} \langle q \rangle^q}{q!} \quad , \tag{2.3}$$

Changing the order of summation $\sum_{q=k}^{\infty}\sum_{l=k}^{q} = \sum_{l=k}^{\infty}\sum_{q=l}^{\infty}$, the equation 2.2 to calculate Z can be further simplified.

$$
\begin{aligned}
(2.4)\quad Z &= f + (1-f)\sum_{l=k}^{\infty}\sum_{q=l}^{\infty}\frac{q+1}{\langle q\rangle}\frac{\langle q\rangle^{q+1}}{(q+1)!}e^{-\langle q\rangle}\frac{q!}{(q-l)!l!}Z^l(1-Z)^{q-l} \\
&= f + (1-f)\sum_{l=k}^{\infty}\frac{e^{-\langle q\rangle}Z^l\langle q\rangle^l}{l!}\sum_{q=l}^{\infty}\frac{\langle q\rangle^{q-l}(1-Z)^{q-l}}{(q-l)!} \\
&= f + (1-f)\sum_{l=k}^{\infty}\frac{e^{-\langle q\rangle}Z^l\langle q\rangle^l}{l!}e^{\langle q\rangle(1-Z)} \\
&= f + (1-f)e^{-\langle q\rangle}\left[e^{\langle q\rangle Z} - \sum_{l=0}^{k-1}\frac{Z^l\langle q\rangle^l}{l!}\right]e^{\langle q\rangle(1-Z)} \\
&= f + (1-f)\left[1 - e^{-\langle q\rangle Z}\sum_{l=0}^{k-1}\frac{Z^l\langle q\rangle^l}{l!}\right]
\end{aligned}
$$

In figure 2.3 the final fraction of activated nodes S_a is displayed for several boostrap percolation simulations at different fractions of seeds in Erdős-Rényi networks with $N = 10^4$ nodes. In these simulations the random number generator used to create the networks is the Mersenne Twister seeded by /dev/random. Before the percolation process begins f nodes are chosen randomly to be the seeds. In each iteration of the simulation we check if there is any node that has at least k active neighboring nodes, and we activate the ones that satisfy this condition.

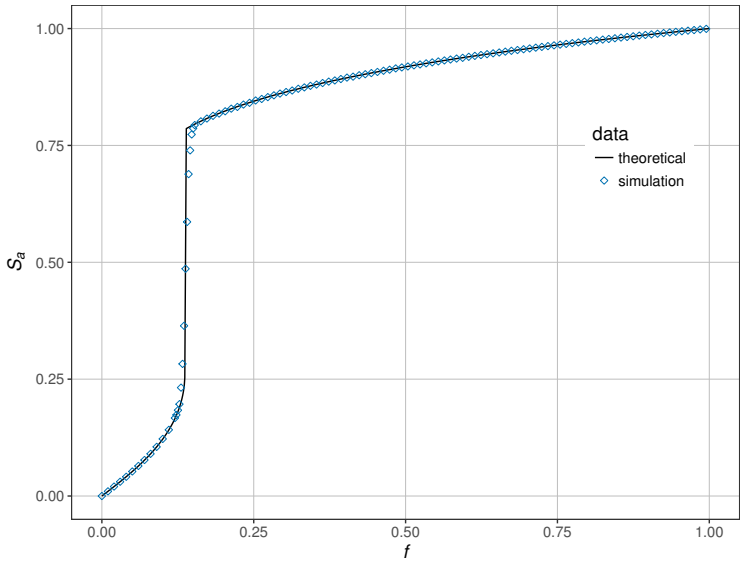

Figure 2.3: The fraction of activated nodes S_a as function of the fraction of initial seeds f in random undirected networks of mean degree $\langle q\rangle = 5$ and activation threshold $k = 3$. Simulation data for $N = 10^4$ nodes. The theoretical result calculated from equation 2.1 is also represented (solid curve).

The number of nodes used in all the simulations is always $N = 10^4$, picked as a balance of computational time and exactness of the equation 2.2 according to the size of the network. Larger networks require more computational time but the differences between the simulation results and the theoretical results calculated through equation 2.2 are negligible as $N \to \infty$, as mentioned before. The theoretical result given by equation 2.1 is also plotted in the figure 2.3, and as we can see both plots are in a very good agreement.

The simulation stops at iteration n when the number of active nodes is not changed between iterations, that is, when S_a at iteration $n-1$ is equal to S_a at n. Due to the random nature that is creating a Erdős-Rényi network and the placement of seeds, the results of the simulations presented here are averaged over 100 times. It is interesting to know how many seeds should

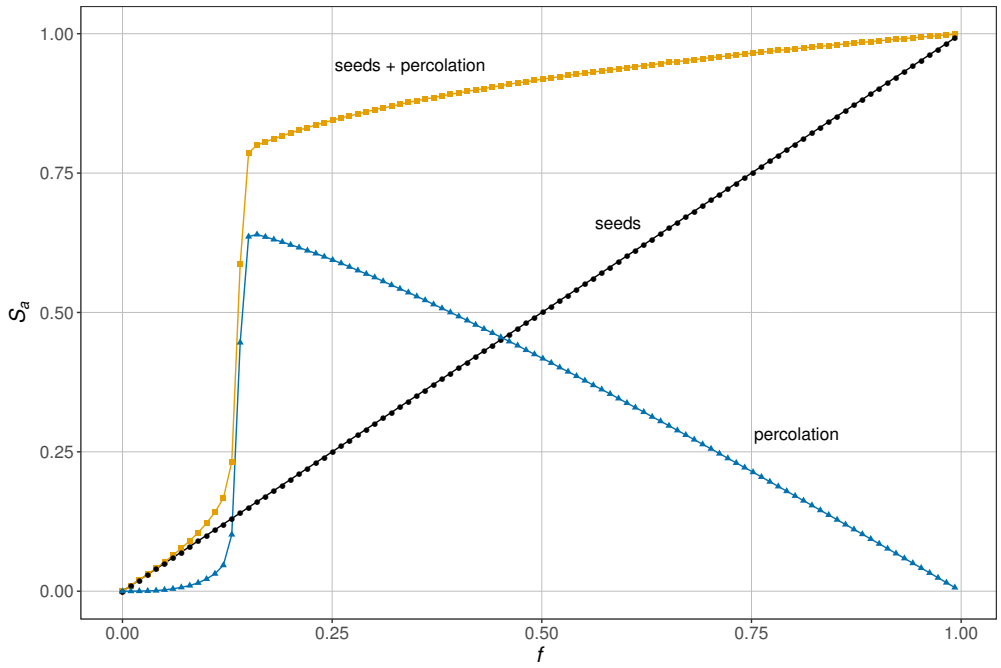

Figure 2.4: Contribution of each term of the boostrap percolation equation 2.1 for the fraction of total activated nodes S_a at the end of the process. The results of equation 2.1 is plotted (yellow squares) and the contribution of the seeds (black circles) and nodes activated through the avalanche process (blue triangles) .

be used as to efficiently activate a large part of a network. In figure 2.4 we can see the same simulation data represented as the ones in figure 2.3 and the contributions of the seeds and the nodes activated during the avalanche process to the total fraction of activated nodes S_a. The number of nodes activated through the percolation process increases following a power law reaching a maximum at the critical point, $f_c = 0.2$, decreasing in a linear fashion until $f = 1$. Therefore if we want to increase a large part of the network using the smallest number of seeds possible, we should use, in this case, $f = 0.2N$ seeds. This jump in the activation occurs due to a phase transition [6]. This happens when a large part of the graph is in a sub-critical state, where

the nodes belonging to this state are not seeds and are inactive nodes with $k-1$ active neighbors. This jump does not appear in all networks, being dependent on the network parameters, such as the mean degree $\langle q \rangle$ and the activation threshold k. The larger k is, for larger f values the jump will appear [6].

CHAPTER 3

BOOTSTRAP PERCOLATION IN DIRECTED COMPLEX NETWORKS

Now let us consider the bootstrap percolation in directed complex networks. An example of this type of percolation is represented in figure 3.1. The bootstrap percolation process is now affected by the in-degree and out-degree distributions. If both distributions are not correlated, the process is only affected by the in-degree distribution, as it is the case in random networks. Generalizing equation 2.1 for undirected networks, we find the equation for the probability Z in directed networks:

$$Z = f + (1-f) \sum_{q_i=k, q_o=0}^{\infty} P(q_i, q_o) \sum_{l=k}^{q_i} \binom{q_i}{l} Z^l (1-Z)^{q_i - l} \,. \tag{3.1}$$

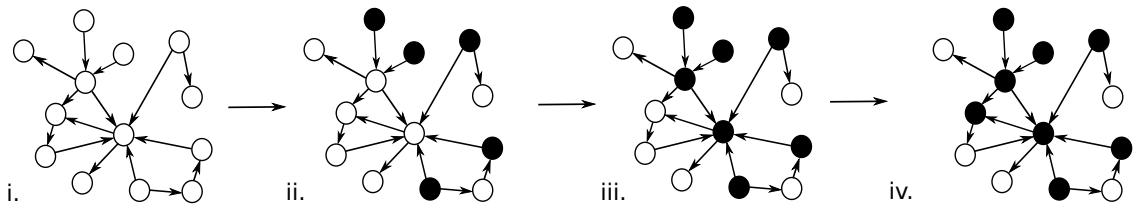

Figure 3.1: An example of a boostrap percolation process for $k = 2$ in a directed complex network. i) a random complex directed network. ii) some nodes are selected to be the seeds, which are represented by the shaded circles. iii) first iteration of the percolation process, every node with an incoming connection from at least $k = 2$ active nodes is activated. iv) second and last iteration of the percolation process, the process stops since there aren't any more nodes that have incoming connections from at least $k = 2$ active neighbouring nodes.

It is important to note that equation 3.1 assumes that all nodes are topologically equivalent to each other. This assumption is correct if the directed complex network consists of only the G_S

and that the other components *IN*, *OUT*, T and *DC* are absent, which generally is not the case. In Erdős-Rényi networks this assumption is correct if the mean degree is large enough $\langle q \rangle$, as we can see in figure 1.6. In this work we use a mean in-degree $\langle q_i \rangle = 5$ in the simulations. In this case we can neglect the size and the contribution of the components apart from the G_S to the bootstrap percolation process.

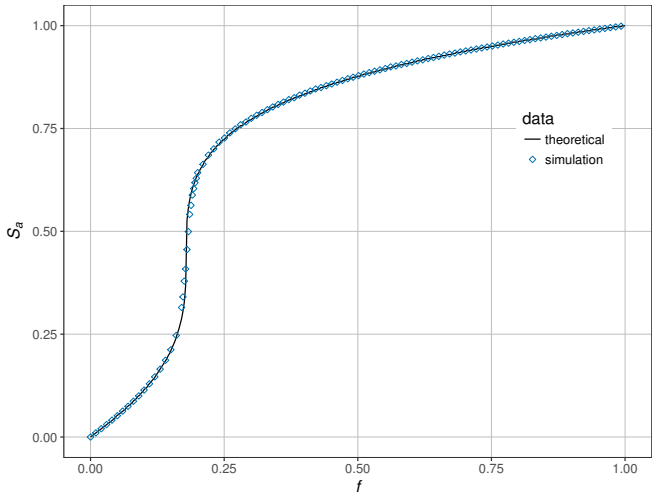

Figure 3.2: The fraction of activated nodes S_a versus the fraction of initial seeds f. Simulation data (circles) for $N = 10^4$ nodes and mean in-degree $\langle q_i \rangle = 4.5$. The theoretical result calculated by 3.1 is also plotted (solid curve).

In directed networks $\langle q_i \rangle = \frac{\langle q \rangle}{2}$ therefore the bootstrap percolation process in a directed network with $\langle q_i \rangle$ should be similar to the process in a undirected network with mean degree $\langle q \rangle = 2\langle q_i \rangle$. By comparing figure 3.3 and figure 3.4 we can see that this is the case.

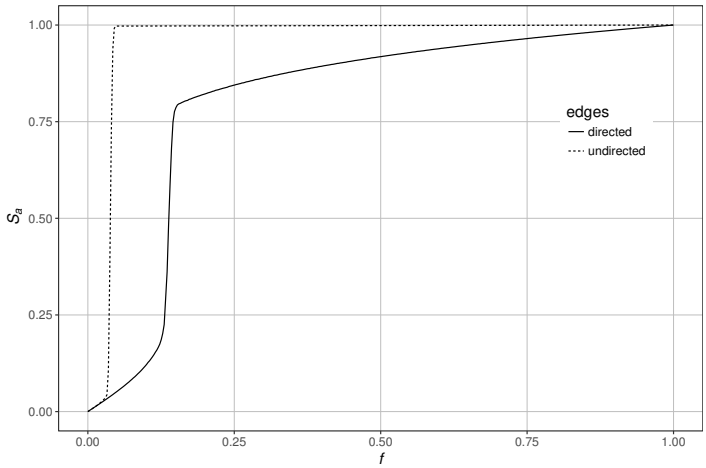

Figure 3.3: Comparison of S_a between simulations of undirected Erdős-Rényi networks with $\langle q \rangle = 10$ (dashed curve) and directed Erdős-Rényi networks with $\langle q \rangle = 10$ (solid curve). The size of the network is $N = 10^4$ nodes.

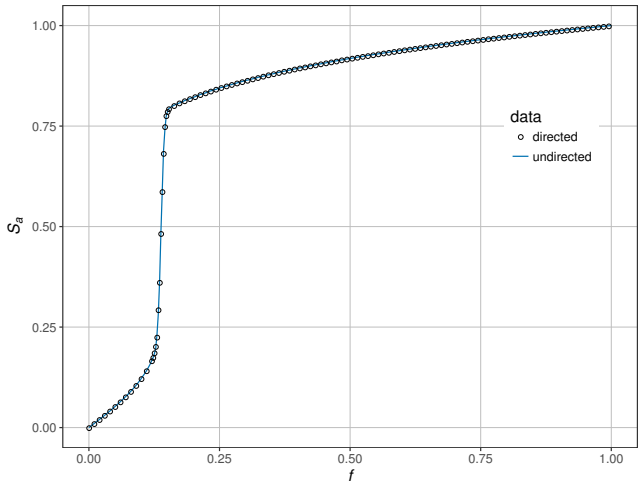

Figure 3.4: Comparison of S between simulations of undirected Erdős-Rényi networks with $\langle q \rangle = 5$ (solid curve) and directed Erdős-Rényi networks with $\langle q \rangle = 10$ (circles). The size of the network is $N = 10^4$ nodes.

3.1 Components

As we discussed before, equation 3.1 gives a good description for the bootstrap percolation in directed complex networks if $\langle q \rangle >> 1$ because one can neglect the existence of all other components apart of the giant strongly connected component G_S. In the general case this is not correct since many real networks are characterized by non-empty IN, OUT, T and DC components.

Due to the existence of different types of components, the placement of seeds strongly affects the number of nodes that are activated at the end of the percolation process. In this section we analyze the bootstrap percolation process in directed complex networks with seeds which are put only in the specific components: G_S, IN and OUT. In figures 3.5, 3.6 and 3.7 results of simulations of bootstrap percolations in Erdős-Rényi for the OUT , IN and G_S cases, respectively, for an activation threshold $k = 2$ are plotted. The parameter f is the fraction of nodes of the chosen components which are seeds.

For the OUT component (figure 3.5) case we can see that we achieve a bigger S_a for $\langle q_i \rangle = 1.5$ (even though $k = 2$) which, for the several mean degrees $\langle q_i \rangle$ plotted, is the closest to the value 1.35 which is the maximum size of the component OUT as one can see in figure 1.6. Since nodes in the OUT cannot activate nodes in the rest of the graph, the activation of other nodes is almost non-existent. The number of activated nodes in the end is approximately equal to the number of seeds.

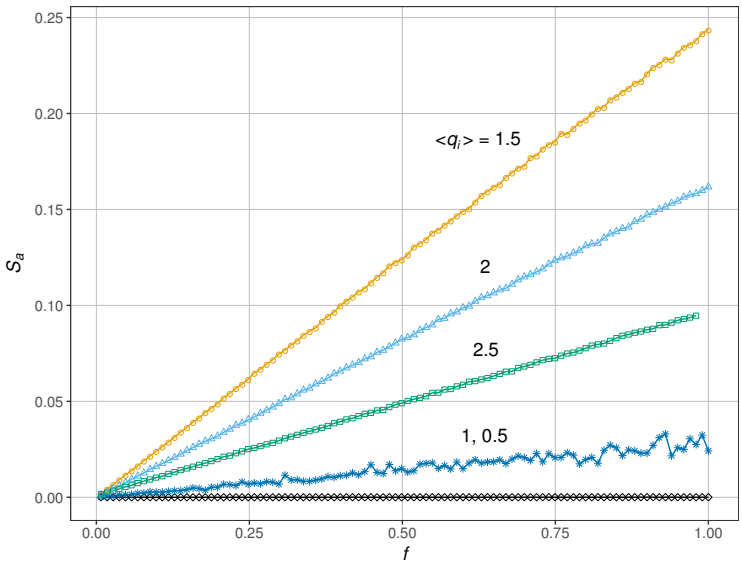

Figure 3.5: Fraction of active nodes S_a versus fraction of seeds f for Erdős-Rényi graphs for several mean degrees for $k = 2$. The seeds were only placed in the OUT. f is the fraction of nodes used as seeds in the OUT. The size of the network is $N = 10^4$ nodes.

For the IN component (figure 3.6) case we also have a bigger number of activated nodes for $\langle q_i \rangle = 1.5$. Contrary to the OUT case there is some activation of other nodes by the seeds as the IN can activate nodes in the G_S and the OUT and some tendrils that are connected to IN and to OUT.

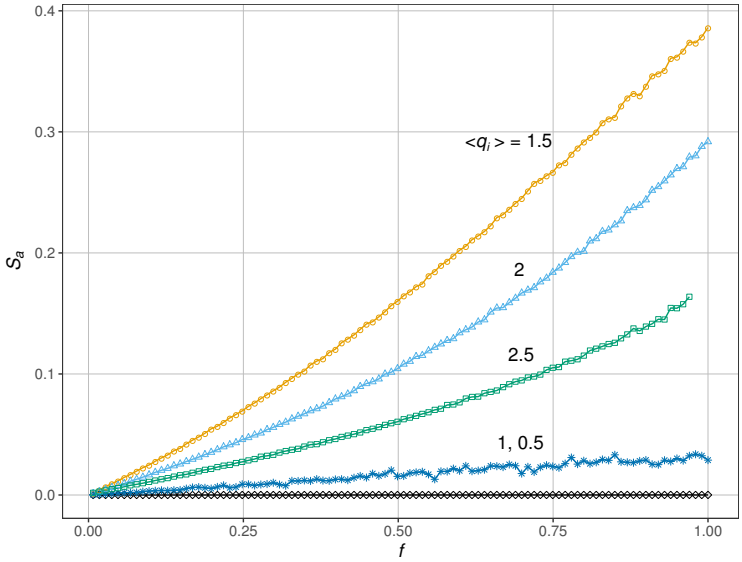

Figure 3.6: Fraction of active nodes S_a versus fraction of seeds f for Erdős-Rényi graphs for several mean degrees for $k = 2$. The seeds were only placed in the IN. f is the fraction of nodes used as seeds in the IN. The size of the network is $N = 10^4$ nodes.

In the case of the giant strongly connected component G_S (figure 3.7), S_a is overall greater than the fraction of activated nodes found for the same mean in-degrees $\langle q_i \rangle$ in the previous cases . For $\langle q_i \rangle = 1$, $S_a(f) \approx 0$ since for this and lower mean degree the G_S does not exist as one can see in figure 1.6. As $\langle q_i \rangle$ is increased we get a larger S_a due to the increase in size of the G_S. Also in the G_S, nodes are more easily activated due to the larger node connectivity, in comparison with the other components.

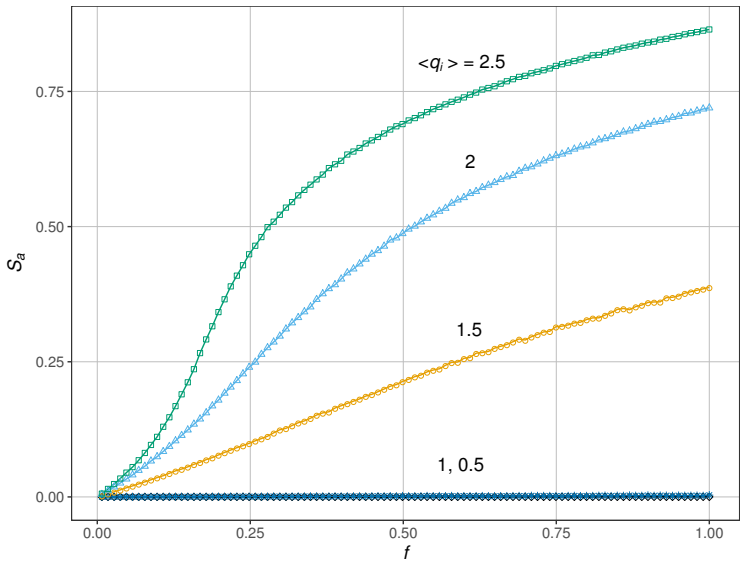

Figure 3.7: Fraction of active nodes S_a versus fraction of seeds f for Erdős-Rényi graphs for several mean degrees for $k = 2$. The seeds were only placed in the G_S. f is the fraction of nodes used as seeds in the G_S. The size of the network is $N = 10^4$ nodes.

CHAPTER 4

BOOTSTRAP PERCOLATION IN WEIGHTED DIRECTED NETWORKS

Many real complex systems such as neural networks [10] are weighted networks. In directed weighted networks the cascade process is different from one in directed unweighted networks. In weighted networks each edge has an weight w with a probability density $P(w)$ while on unweighted networks each edge is considered to have a weight $w = 1$.

Let us define the function $f(i)$ of whether a neighboring incoming node i is active or unactive,

$$f(i) = \begin{cases} 1, \text{ if } i \text{ is active} \\ 0, \text{ if } i \text{ is unactive} \end{cases}$$

during the cascade process, a node with q_i neighboring incoming nodes will become active if,

$$\sum_{i}^{q_i} w_i f(i) = ls \geq k \tag{4.1}$$

where w_i is each neighboring incoming node weight, l is the number of active incoming nodes and s is the sum of their weights. The probability density function of the sum of weights w is denoted by $P_{s,q_i}(s)$.

Since every combination of weights of which sum ranging from k till ∞ can activate a node, we add the integral of $P_s(s)$ from k to ∞ to the equation of the boostrap percolation 3.1. The main equation for the boostrap percolation in weighted directed networks is,

$$Z = f + (1-f) \sum_{q_i=1, q_o=0}^{\infty} P(q_i, q_o) \sum_{l=1}^{q_i} \binom{q_i}{l} Z^l (1-Z)^{q_i - l} \int_k^{\infty} P_{s,q_i}(s)\, ds \tag{4.2}$$

Since one edge alone can weight enough to activate a node, both the sums for the variables q_i and l must start at 1. $P_{s,q_i}(s)$ is the probability density function of the sum of the edges weights incoming from l active nodes. If the weights and the mean in-degree distributions are correlated

then we need to account for this correlation, $P_{s,q_i}(s)$ will have a different distribution for each q_i because there will be a different weight distribution $P(w)$ for each q_i.

If the probability density of edge weights follow a normal distribution with mean $\langle w \rangle$ and variance σ_w^2, $W \sim N\left(\langle w \rangle,\ \sigma_w^2\right)$, the sum of l weights follows a normal distribution $S \sim N\left(l\langle w \rangle,\ l\sigma_w^2\right)$,

$$P_{s,q_i}(s) = \frac{1}{\sqrt{2\pi l}\,\sigma_w} \exp\left[-\frac{(s - l\langle w \rangle)^2}{2l\sigma_w^2}\right] \tag{4.3}$$

If w is not normally distributed one also can apply the Central Limit Theorem to approximate $P_{s,q_i}(s)$. In this case, the mean in-degree must be sufficiently large for a given mean edge weight in order to neglect the cases of small values of l when the normal distribution (Eq. 4.3) is not a good approximation for $P_{s,q_i}(s)$.

In this work we use a weight distribution $P(w)$ which is approximately equal to the normal distribution. Since we don not want any negative weights, in the simulations, when the random number generator generates a weight smaller than zero, the generator is run again until it generates a number greater than zero. The shape of this distribution is different from a normal distribution. This is specially true for large variances and we should take it into account in the theoretical formula where we need to model the real weight distribution and not use any normal approximations. Since the weights are generated randomly we will denote $P_{s,q_i}(s)$ by $P_s(s)$.

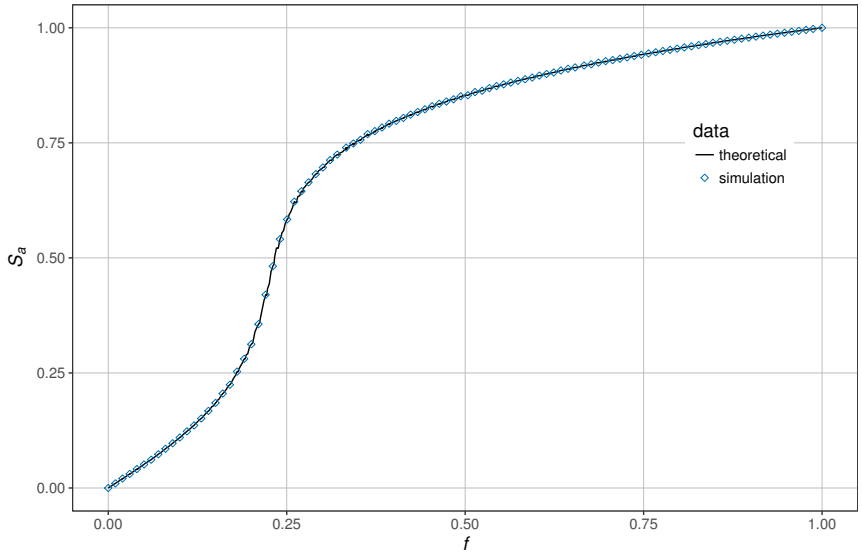

Figure 4.1: Theoretical (solid curve) and simulation (circles) data for $N = 10^4$ nodes, mean in-degree $\langle q_i \rangle = 5$, and weights distribution $W \sim N(1, 0.04)$.

Figure 4.2 represents results of bootstrap percolation simulations results for unweighted directed networks with $\langle q_i \rangle = 5$ and weighted directed networks with the same mean in-degree and weight distribution $W \sim N(1, 0.04)$.

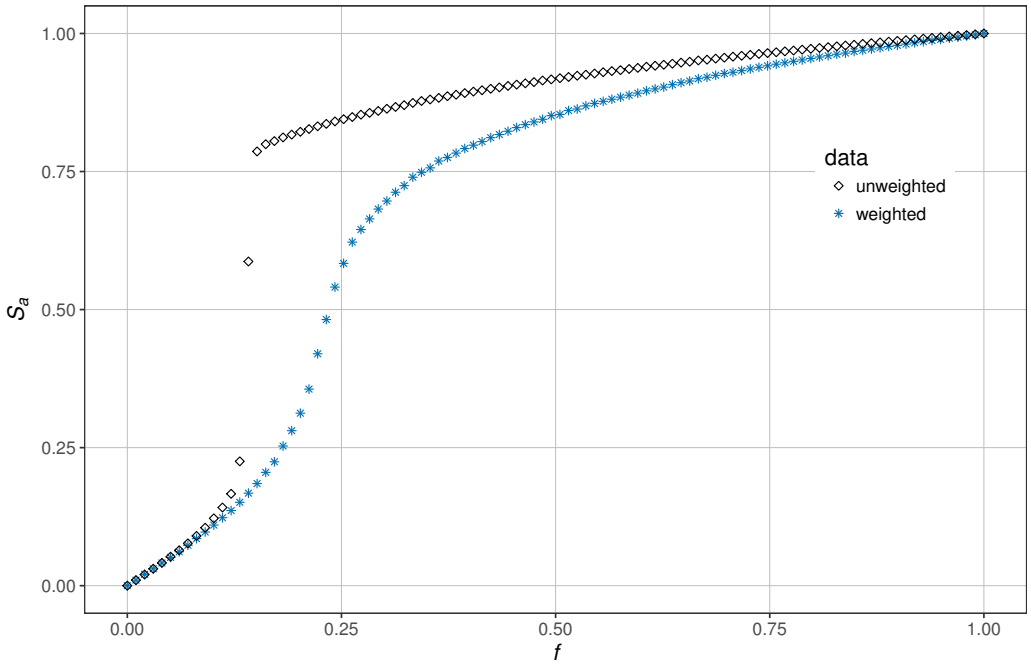

Figure 4.2: Simulation data for bootstrap percolation in unweighted directed networks (diamonds) and weighted directed networks (stars) with weight distribution $W \sim N(1, 0.04)$ for $N = 10^4$ nodes, mean in-degree $\langle q_i \rangle = 5$ and activation threshold $k = 3$.

Analyzing figure 4.2 we can see that S_a in the unweighted case is greater (or equal for $f \to 1$ and $f \to 0$) than in the weighted case. Since the mean in-degree and the threshold k are both the same in both cases and that $\langle w \rangle = 1$ in the weighted case, it should be expected that when we decrease the weights variance σ_w^2, the bootstrap percolation in the weighted case must become closer to the bootstrap percolation in the unweighted case.

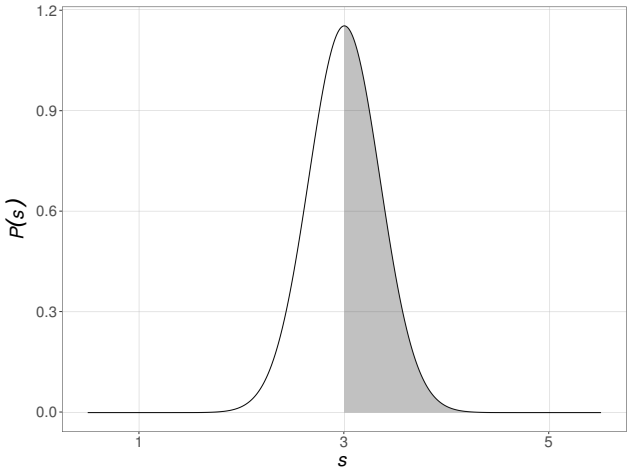

Figure 4.3: $P(s)$ for 3 edges which weights follow a normal distribution $W \sim N(1, 0.04)$.

Since the activation threshold k is an integer, in the case $k = 3$ and the mean weight $\langle w \rangle = 1$, there is some probability $P_{s,}(s < l)$ that the unactive node may not become active when the number of active neighbouring nodes is $l = k$. In the unweighted case the unactive node will always be activated. In figure 4.3 it is represented the distribution of the sum of 3 weights following a normal distribution $W \sim N(1, 0.04)$. Only when $s > 3$ the node will be activated.

In figure 4.4 we can see bootstrap percolation simulation data for several weight standard deviations σ_w. As σ_w increases, the larger S_a becomes and a jump in the activation appears starting at $\sigma_w = 1$. In figure 4.5 we can see the distributions of the weights with standard

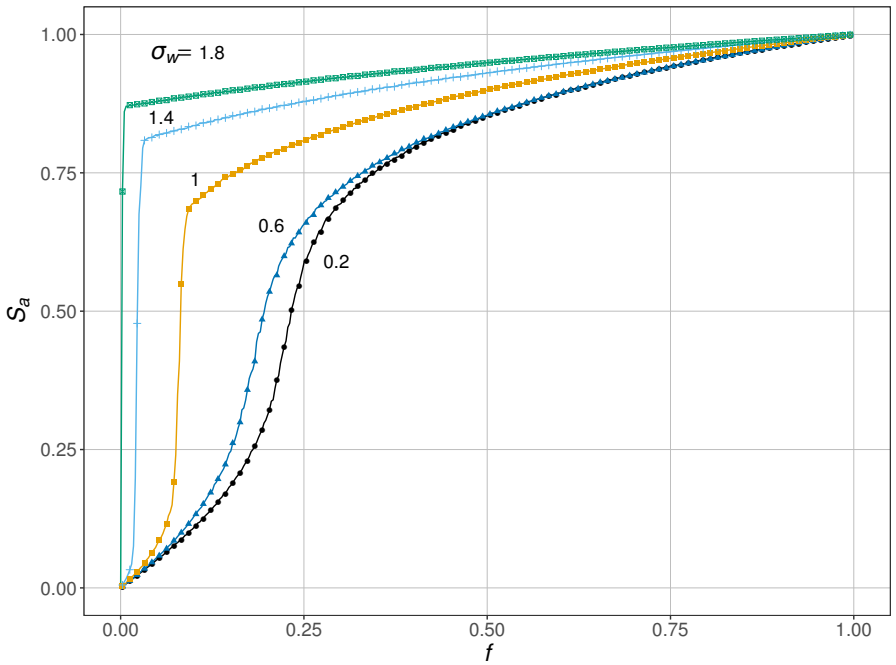

Figure 4.4: Simulation data for bootstrap percolation in random weighted directed complex networks for several weight standard deviations σ_w and mean weight $\langle w \rangle = 1$. Here, the mean in-degree $\langle q_i \rangle = 5$, the activation threshold $k = 3$ and $N = 10^4$ nodes.

deviations equal to the ones used in 4.4. In figure 4.6 we can see the distributions of the sum of 3 weights for the same standard deviations in the two previous figures.

The increase of S_a with increasing σ_w can be explained due to the increase of $\langle w \rangle$ as σ_w increases. This is due to the effective distribution of weights $P_s(s)$ being generated from the initial normal distribution with the variance σ_w by forbiddance to have negative weights. This leads to increasing $< w >$ and therefore $\langle s \rangle$ with increasing σ_w (see Fig. 4.5 4.6).

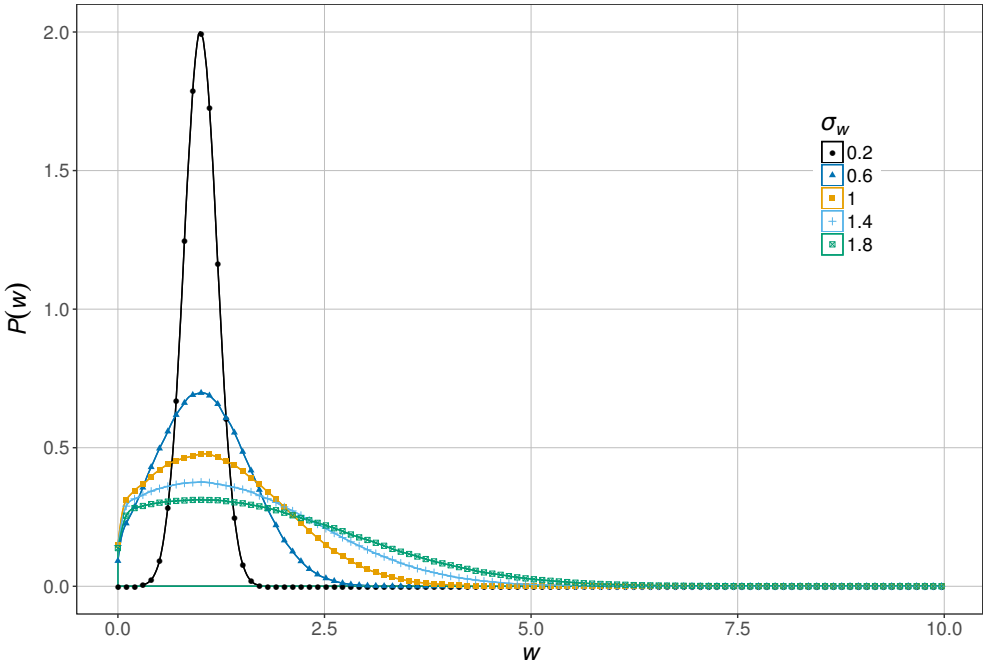

Figure 4.5: Edges weight distribution for several weights standard deviations σ_w with mean weight $\langle w \rangle = 1$.

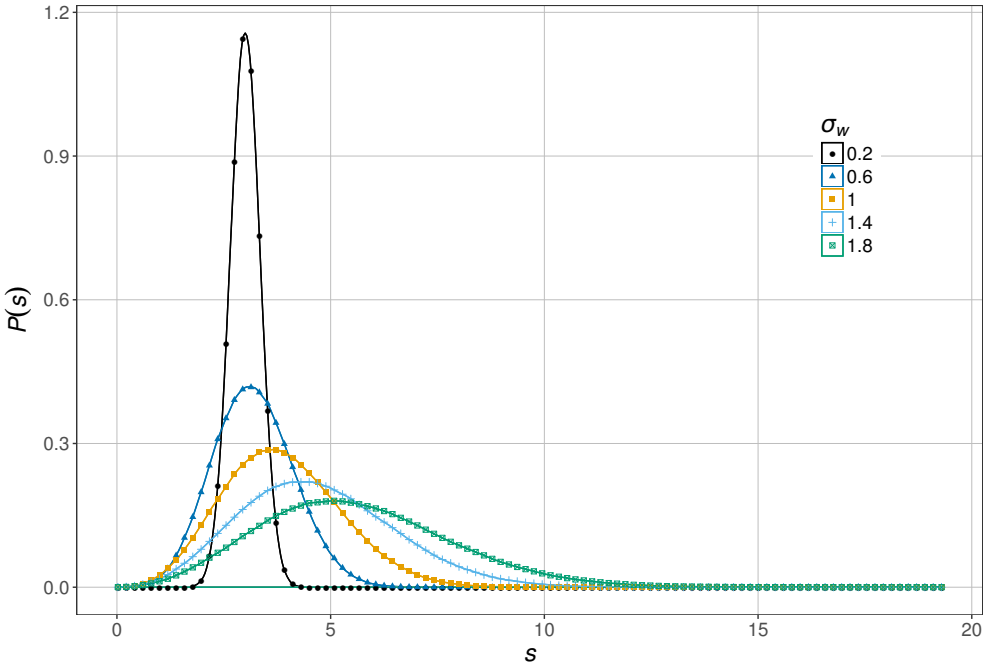

Figure 4.6: The distributions $P_s(s)$ of the sum of three weights $s = w_1 + w_2 + w_3$ with distributions $P(w)$ equal to the ones in figure 4.5. When the standard deviation σ_w of the weights increases, the mean sum of weights $\langle s \rangle$ increases.

CHAPTER 5

PROBABILITY BASED BOOSTRAP PERCOLATION IN DIRECTED COMPLEX NETWORKS

Let us consider another kind of cascade model on unweighted networks a so called probability based percolation process. In this model every edge has the same probability p of transmitting a signal. A node will become active if at least k active incoming neighboring nodes send a signal simultaneously. The activation probability is given by a binomial distribution,

$$\sum_{t=k}^{l} \binom{t}{l} p^t (1-p)^{l-t} \tag{5.1}$$

where l is the number of active incoming neighboring nodes and t the number of active incoming neighboring nodes that sent the signal.

Instead of equation 3.1, the boostrap percolation equation is now:

$$Z = f + (1-f) \sum_{q_i=k, q_o=0}^{\infty} P(q_i, q_o) \sum_{l=k}^{q_i} \binom{l}{q_i} Z^l (1-Z)^{q_i-l} \sum_{t=k}^{l} \binom{t}{l} p^t (1-p)^{l-t} \quad . \tag{5.2}$$

In the simulations, if we use the condition used so far for the end of the boostrap percolation process where the process ends if the fraction of active nodes S_a in the last iteration n, is the same as S_a in the previous iteration $n-1$, the previous equation 5.2 does not hold. Whereas the avalanche process would always stop in a non-probability based percolation at an iteration $n-1$, in a probability based percolation, between iterations $n-1$ and n, we are giving another chance of activation of the nodes that previously did not activate due to simple probability, therefore the process will not stop. The number of total iterations n in a probability based percolation will be larger than n in a non-probability based percolation. If we ignore our boostrap percolation ending condition and let the process run for a large enough n, S_a in a probability based percolation

process will be the same as in a non-probability based process. To limit the number of chances a node gets to become active, every time a neighboring incoming node becomes active, the node will get another chance at the activation. Therefore a node will get a maximum of $q_i - k + 1$ chances at the activation. Other condition of activation was tested where each node would only get one chance at the activation across the whole simulation, when it would have at first at least k active neighbors, but this condition has proven to be worse when compared to the theoretical result of the previous formula.

In figure 5.1 we can see the comparison between the theoretical result given by the previous equation and simulation data for $p = 0.7$ for the model in which each node has at maximum $q_i - k + 1$ chances to become activated. The results show some disagreement in the interval $f \approx [0.2, 0.6]$. Even though $\langle q_i \rangle = 5$ and the size of the $G_S \approx N$, the difference in the results could be due to the non-negligible size of the other network components. This difference can also be due to activation condition.

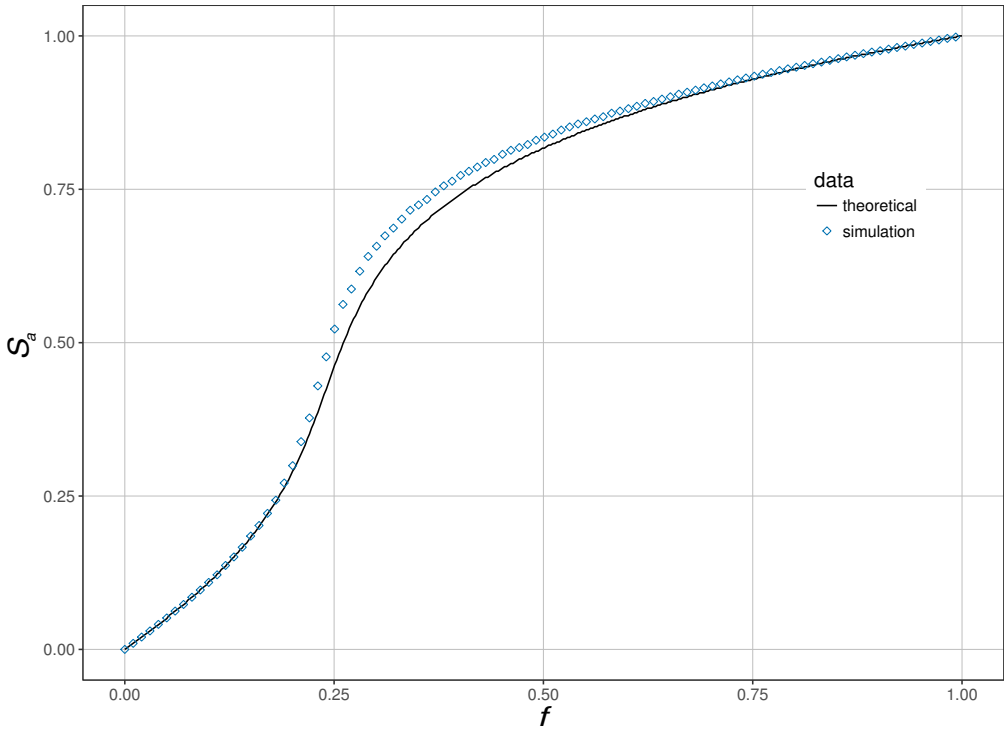

Figure 5.1: Probability based percolation in Erdős-Rényi with probability $p = 0.7$, activation threshold $k = 3$ and mean average in-degree $\langle q_i \rangle = 5$. Both the theoretical results (solid line) given by 5.2 and simulation data (diamonds) for $N = 10^4$ nodes are plotted. In the simulations each node has at maximum $q_i - k + 1$ chances to become activated.

CHAPTER 6

BOOTSTRAP PERCOLATION IN REAL NETWORKS

In order to apply the theoretical approaches developed above to real networks, we now compare some bootstrap percolation theoretical results and simulations results for a Twitter network, Google+ network and the neural network of the worm *Caenorhabditis elegans*. The first two networks, which are social networks, the activation threshold can be viewed as a "trust" level where a person becomes "active" if he/she receives the same type of information from at least k other people. For the neural network of the worm *Caenorhabditis elegans* the neurons can be viewed as nodes and the synapses and gap junctions between them as edges. This network is also weighted where the weights are the conductances of chemical and electrical connections between neurons.

6.1 Twitter

The Twitter network which we used [21] is composed by $N = 81305$ nodes. The in-degree and out-degree distributions follow a power-law distribution with the mean in-degree $\langle q_i \rangle \approx 29.8$ and the mean out-degree $\langle q_o \rangle \approx 34.5$. This network consists only in the giant strongly connected component G_S having size of 68413 nodes (84.14% of N) and the OUT component having size of 12891 nodes (15.86% of N).

Results of bootstrap percolation simulations run in this network are presented in figure 6.1. The theoretical result obtained through equation 3.1 is also presented in the same figure.

As we have discussed, the larger is the giant strongly connected component G_S, the more accurately the bootstrap percolation equation 3.1 represents a real bootstrap percolation process. As we can see in figure 6.1 both theory and simulation results are in a very good agreement due to most of the network being composed by G_S. This agreement is not perfect though, we can see some slight differences between the two lines. The size of the OUT component is not

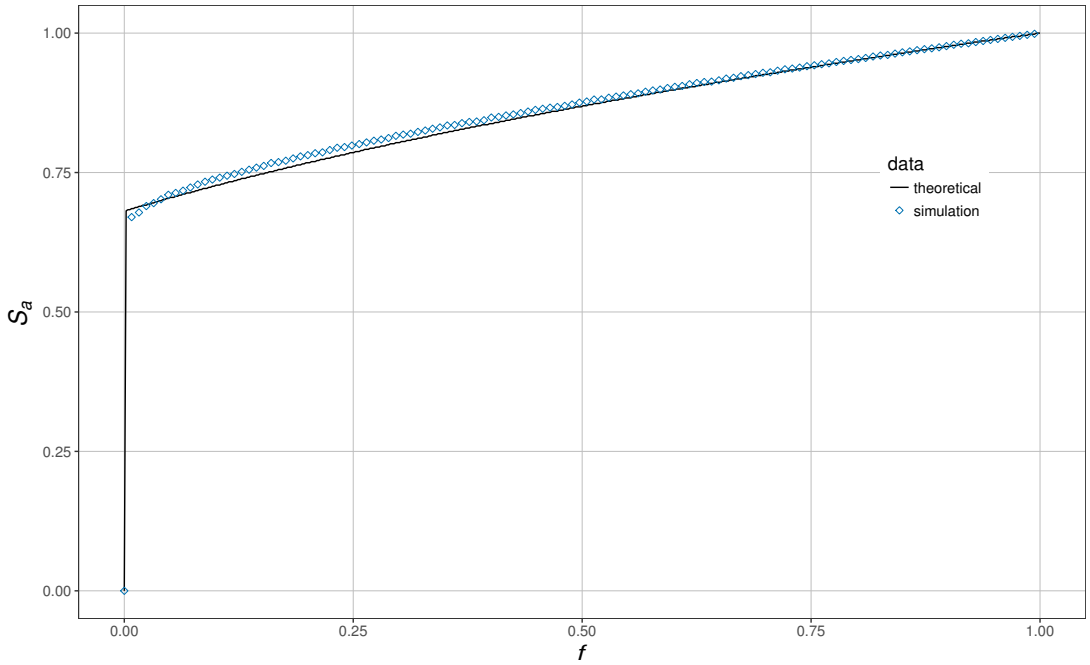

Figure 6.1: Boostrap percolation in a Twitter directed network with an activation threshold $k = 3$ and $N = 81305$ nodes (blue diamonds). The theoretical result calculated through equation 3.1 is also shown (black line). Both results are in a good agreement.

negligible (15.86% of N) and this is why both plotted theoretical and simulation lines are a little bit different from each other.

6.2 Google+

The Google+ network which we used [21] is composed by $N = 107614$ nodes. The in-degree and out-degree distributions follow a power-law distribution with the mean in-degree $\langle q_i \rangle \approx 4.7$ and the mean out-degree $\langle q_o \rangle \approx 7.9$. This network has a giant strongly connected component G_S with size of 69501 nodes (64.58% of N), a OUT component having size of 7924 nodes (7.36% of N) and a IN component having size of 338 nodes (0.31% of N), while the rest of the network consists in tendrils.

Results of bootstrap percolation simulations run in this network are presented in figure 6.2. The theoretical result obtained through equation 3.1 is also presented in the same figure.

Contrary to the Twitter network, the size of the giant strongly connected G_S is rather small. The OUT and IN components are also small in size. As we can see in figure 6.2 the theoretical and simulation results are very different which is due to the fact that the remaining part of the network is occupied by tendrils which are not taken into account in equation 3.1. The fraction of activated nodes S_a in the theory is bigger than in the simulations: the theory expects a bigger

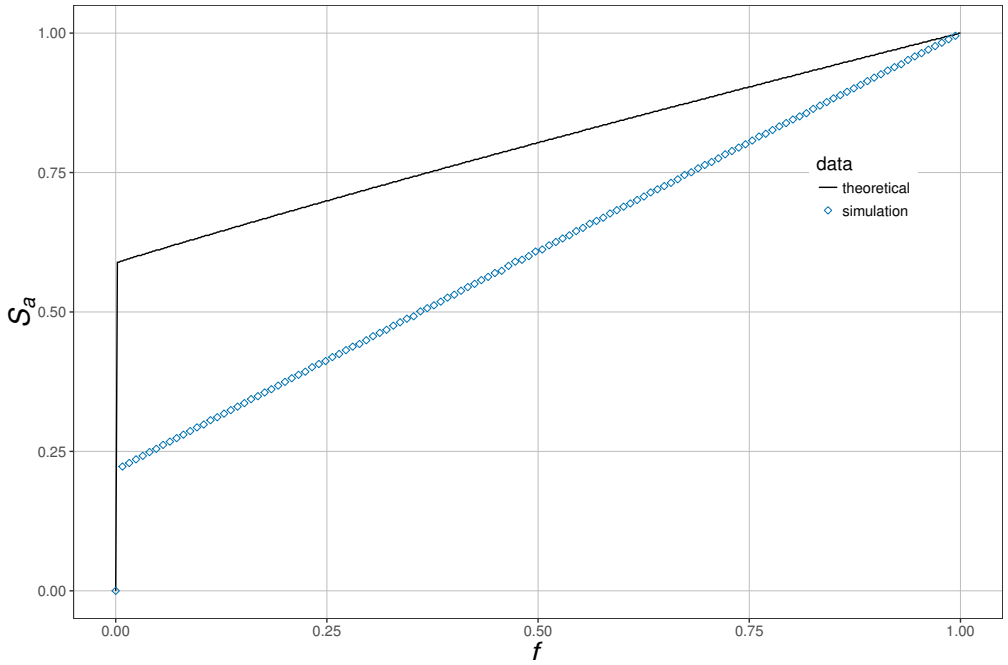

Figure 6.2: Boostrap percolation in a Google+ directed network with an activation threshold $k = 3$ and $N = 107614$ nodes (blue diamonds). The theoretical result calculated through equation 3.1 is also shown (black line).

connectivity in the network, characteristic of a large giant strongly connected component G_S, which in this network is very small.

6.3 *Caenorhabditis elegans* neural network

The *Caenorhabditis elegans*'s neural network we used [22] is composed by $N = 297$ nodes. The in-degree and out-degree distributions follow a power-law distribution with the mean in-degree $\langle q_i \rangle \approx 8.65$ and the mean out-degree $\langle q_o \rangle \approx 7.9$. The edge weights distribution has mean value $\langle w \rangle = 3.76$ and is correlated with the degree distributions.

This network has a giant strongly connected component G_S having size of 239 nodes (80.47% of N), a *OUT* component having size of 27 nodes (9.09% of N) and a *IN* component having size of 16 nodes (5.38% of N).

Results of bootstrap percolation simulations run in this network are presented in figure 6.3. The theoretical result obtained through equation 4.2 is also presented in the same figure. Since the edge weights and the mean in-degrees are correlated we need to model a different $P_s(s)$ for each q_i. The results are in good agreement for $f > 0.4$, while for lower values of f there is some noticeable difference between the two plots. The giant strongly connected component G_S only occupies 80% of the network being this the main reason for the disagreement. It is worth to mention that this network is relatively small, $N = 297$ nodes, and our bootstrap percolation

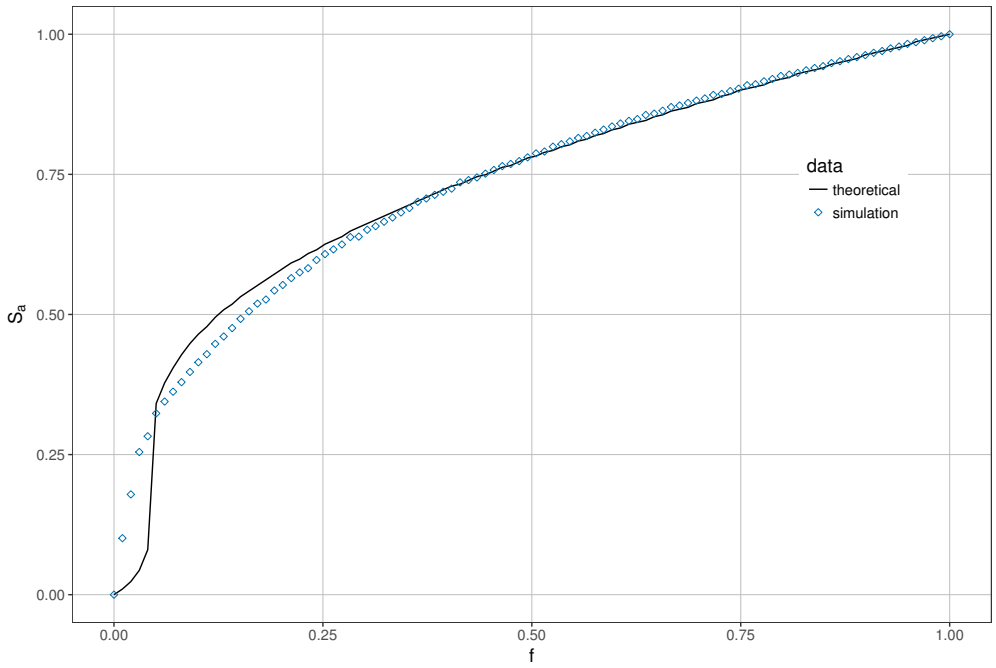

Figure 6.3: Boostrap percolation in a *Caenorhabditis elegans* weighted directed network with an activation threshold $k = 8$ and $N = 297$ nodes (blue diamonds). The theoretical result calculated through equation 4.2 is also shown (black line).

formula works best when $N \to \infty$. Moreover, our approach was developed for tree-like networks while the *C. elegans* network has a large clustering coefficient [10]. The number of nodes activated by the percolation process in networks with a large clustering coefficient is larger than those activated in tree-like networks. In our case, this is noticeable for $f = [0, 0.05]$ where S_a in our theoretical results are smaller than the simulations results.

CHAPTER 7

CONCLUSION

In this work we have increased our understanding of the bootstrap percolation process in random directed networks with an activation threshold k. We have begun in Chapter 1 by giving a basic introduction to complex networks and in Chapter 2 reviewing the bootstrap percolation in undirected complex networks. In Chapter 3 we have entered the main objective of this work that was to study the bootstrap percolation process in random complex directed networks. We formulated the bootstrap percolation process in this type of networks and found a good agreement between our theory and simulations. Next we've seen how the giant strongly connected component G_S and the OUT and IN components affect the percolation process when seeds are put only in these components. In random networks, as the mean degree $\langle q \rangle$ becomes large enough, G_S gets the leading contribution to the percolation process since it dominates all the network. The size of the OUT and IN components, and all the other components becomes negligible. Due to this fact, the role of the components in the percolation process is somewhat difficult to study in random networks with an activation threshold. While the mean degree should be relatively low enough so that the size of the IN and OUT components is noticeable, a low mean degree makes it difficult for the percolation process. The roles of the IN and OUT components in random networks can't be fully studied due to this fact. Nonetheless if we would like to activate the largest part possible of a network with all the components with the same size and if we could only choose one component to place the seeds, this would be generally the IN component since it can activate G_S and then the OUT component.

In Chapter 4 we have formulated the bootstrap percolation process in weighted directed networks where the edges weights follow an arbitrary distribution. It was found a good agreement between the theory and the simulations. In Chapter 5 we studied the probability based percolation also with an activation threshold where the nodes have a probability p of sending an activation

signal to their out neighbors. The theory and the simulations are in a good agreement, having only a small quantitative disagreement. Since the activation is probability based, if we could give infinite chances at the activation, the number of activated nodes of this type of bootstrap percolation would be the same as the normal process studied before. The correctness of our proposed bootstrap percolation process formula is dependent on the number of chances a node gets to become active.

In Chapter 6 we applied the theory from the previous chapters to real directed networks. Real networks are noticeably different from random networks. One of the main differences is the relative sizes of the components. While random complex networks (Erdős-Rényi networks) with $\langle q \rangle \geq 5$ are mainly composed by the giant strongly connected component G_S and the size of the periphery is negligible, on real networks with a large mean degree we can't neglect the size of the components. Our bootstrap percolation formulas are based on a large network connectivity (a large G_S) and this connectivity is much smaller the more components there are on a network. Networks with a large G_S such as our tested Twitter network, even though it had a non-negligible OUT component, the bootstrap percolation simulations results are in good agreement with the theory. In the case of the Google+ network, our bootstrap percolation equation disagrees with the real results of the simulations since the size of the network components (apart from the G_S) are non-negligible. On the neural network of the worm *Caenorhabditis elegans* the theory and the simulation results are in a good agreement with some quantitative disagreement for lower seed fractions f due to G_S being accompanied by other types of components. Also this small disagreement is due the network itself being relatively small and having a large clustering coefficient, while our approach was developed for tree-like networks with infinite size.

For further investigation one can study the bootstrap percolation process with a variable threshold [6], where each node has a different activation threshold k. Concerning the probability based percolation, one can also study the case where each edge has a different probability p akin to the weighted network type.

BIBLIOGRAPHY

[1] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, A. T. Raymie Stata, and J. Wiener, "Graph structure in the web," *Computer Networks*, vol. 33, pp. 309–320, 2000.

[2] J. A. Dunne, R. J. Williams, and N. D. Martinez, "Food-web structure and network theory: The role of connectance and size," *Proceedings of the National Academy of Sciences*, vol. 99, no. 20, pp. 12917–12922, 2002.

[3] H.-S. Alam, "A complex network approach towards modeling and analysis of the australian airport network," *Journal of Air Transport Management*, 2017.

[4] A. V. Goltsev, F. V. de Abreu, S. N. Dorogovtsev, and J. F. F. Mendes, "Stochastic cellular automata model of neural networks," *Physical Review E*, vol. 81, 2010.

[5] D. Kempe, J. Kleinberg, and Éva Tardos, "Maximizing the spread of influence through a social network," *Proceeding KDD '03 Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146, 2003.

[6] G. J. Baxter, S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, "Bootstrap percolation on complex networks," *Physical Review E*, vol. 82, 2010.

[7] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, "Critical phenomena in complex networks," *Review of Modern Physics*, vol. 80, 2008.

[8] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Network robustness and fragility: Percolation on random graphs," *Physical Review Letters*, vol. 85, pp. 5468–5471, 2000.

[9] S. Vitali, J. B. Glattfelder, and S. Battiston, "The network of global corporate control," *CoRR*, vol. abs/1107.5728, 2011.

[10] T. A. Jarrell, Y. Wang, A. E. Bloniarz, C. A. Brittin, M. Xu, J. N. Thomson, D. G. Albertson, D. H. Hall, and S. W. Emmons, "The connectome of a decision-making neural network," *Science*, vol. 337, pp. 437–443, 2012.

[11] M. E. J. Newman, "The structure and function of complex networks," *Society for Industrial and Applied Mathematics*, vol. 45, 2003.

[12] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, “Critical phenomena in complex networks,” *Rev. Mod. Phys.*, vol. 80, pp. 1275–1335, Oct 2008.

[13] S. N. Dorogovstev and J. F. F. Mendes, *Evolution of Networks*.
American Society of Plant Biologist, 2003.

[14] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[15] O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag, “Organization, development and function of complex brain networks,” *Trends Cogn Sci.*, 2004.

[16] G. Bagler, “Analysis of the airport network of india as a complex weighted network,” *Physica A: Statistical Mechanics and its Applications*, 2008.

[17] Y. Iturria-Medina, R. C. Sotero, E. J. Canales-Rodríguez, Y. Alemán-Gómeza, and L. Melie-Garcá, “Studying the human brain anatomical network via diffusion-weighted mri and graph theory,” *NeuroImage*, 2008.

[18] M. Molloy and B. Reed, “A critical point for random graphs with a given degree sequene,” *Random Structures and Algorithms*, vol. 6, pp. 161–180, 2000.

[19] A. V. Goltsev, G. Timar, and J. F. F. Mendes, “Sensitivity of directed networks to the addition and pruning of edges and vertices,” 2017.

[20] S. R. Kosaraju, “Strongly connected components algorithm.” Unpublished, 1978.

[21] J. Leskovec and J. J. Mcauley, “Learning to discover social circles in ego networks,” in *Advances in Neural Information Processing Systems 25* (P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), pp. 548–556, 2012.

[22] D. J. Watts and S. H. Strogatz, “Collective dynamics of "small-world" networks,” *Nature*, pp. 440–442, 1998.