Sapienza University of Rome

Ph.D. program in Computer Engineering

XXVII Cycle - 2015

Computing on Evolving Social Networks

Francesco Ficarola

Sapienza University of Rome

Ph.D. program in Computer Engineering

XXVII Cycle - 2015

Francesco Ficarola

Computing on Evolving Social Networks

Thesis Committee

Prof. Andrea Vitaletti (Advisor)
Prof. Luca Iocchi (Co-Advisor)

Reviewers

Prof. Giancarlo Ruffo
Prof. Mirco Musolesi
Dr. Emiliano Miluzzo

Author's address:
Francesco Ficarola
Dipartimento di Ingegneria Informatica, Automatica, Gestionale
Sapienza Università di Roma
Via Ariosto 25, 00185 Roma, Italy
e-mail: ficarola@dis.uniroma1.it
www: http://www.dis.uniroma1.it/~ficarola

For my grandmother, Maria, who is the most generous person I've ever known.

Acknowledgements

First and foremost I offer my honestest gratitude to my supervisor, Prof. Andrea Vitaletti, who introduced me to the wonderful world of research and inspired me throughout my thesis with his precious advice and inventiveness.

I am also immensely thankful to Prof. Luca Becchetti and Prof. Aris Anagnostopoulos, two fantastic researchers. Along with Andrea, Luca and Aris have offered me a lot of their time, knowledge and friendship (as well as a lot of coffees). Moreover, I want to thank Prof. Alberto Marchetti-Spaccamela for supporting me over and over again.

My authentic gratitude goes to my friends Stefano Puglia and Luigi Teodonio, who spent their time in reading my thesis and giving me priceless suggestions. I wish to thank also all other friends for their constant moral support.

I am endless grateful to my mother and my father, who have made possible my dream.

Finally, I want to heartily thank Michela, my better half, who has followed me to the ends of the earth.

Abstract

Over the past decade, participation in social networking services has seen an exponential growth, so that nowadays most individuals are "virtually" connected to others anywhere in the world. Consistently, analysis of human social behavior has gained momentum in the computer science research community. Several well-known phenomena in the social sciences have been revisited in a computer science perspective, with a new focus on phenomena of emerging behavior, information diffusion, opinion formation and collective intelligence. Furthermore, the recent past has witnessed a growing interest in the dynamics of these phenomena and that of the underlying social structures.

This thesis investigates a number of aspects related to the study of evolving social networks and the collective phenomena they mediate. We have mainly pursued three research directions.

The first line of research is in a sense functional to the other two and concerns the collection of data tracking the evolution of human interactions in the physical space and the extraction of (time) evolving networks describing these interactions. A number of available datasets describing different kinds of social networks are available on line, but few involve physical proximity of humans in real life scenarios. During our research activity, we have deployed several social experiments tracking face-to-face human interactions in the physical space. The collected datasets have been used to analyze network properties and to investigate social phenomena, as further described below.

A second line of research investigates the impact of dynamics on the analytical tools used to extract knowledge from social networks. This is clearly a vast area in which research in many cases is in its early stages. We have focused on centrality, a fundamental notion in the analysis and characterization of social network structure and key to a number of Web applications and services. While many social networks of interest (resulting from "virtual" or "physical" activity) are highly dynamic, many Web information retrieval algorithms were originally designed with static networks in mind. In this thesis, we design and analyze decentralized algorithms for computing and maintaining centrality scores over time evolving networks. These algorithms refer to notions of centrality which are explicitly conceived for evolving settings and

which are consistent with PageRank in important cases.

A further line of research investigates the wisdom of crowds effect, an important, yet not completely understood phenomenon of collective intelligence, whereby a group typically exhibits higher predictive accuracy than its single members and often experts. Phenomena of collective intelligence involve exchange and processing of information among individuals sharing some common social structure. In many cases of interest, this structure is suitably described by an evolving social network. Studying the interplay between the evolution of the underlying social structure and the computational properties of the resulting process is an interesting and challenging task. We have focused on the quantitative analysis of this aspect, in particular the effect of the network on the accuracy of prediction. To provide a mathematical characterization, we have revisited and modified a number of models of opinion formation and diffusion originally proposed in the social sciences. Experimental analysis using data collected from some of the social experiments we conducted allowed to test soundness of the proposed models. While many of these models seem to capture important aspects of the process of opinion formation in (physical) social networks, one variant we propose achieves higher predictive accuracy and is also robust to the presence of outliers.

Contents

Introduction

In the past decade the computer science research community has paid increasing attention to the study of complex phenomena occurring in social networks (we use this term in a broad sense for the moment). This interest has certainly been fostered by the explosive growth and popularity of online social networking platforms and services [91] (e.g. Facebook, LinkedIn, Google+ and so on). Currently, research in the area involves a growing number of researchers and social media professionals, whose interests cover virtually every aspect of this topic and related ones, such as social dynamics and collective behavior. Understanding these phenomena and using the acquired knowledge to make reliable predictions can be of the utmost importance and can provide commercial value to many applications and tasks, including reputation management, recommender systems, trend prediction or targeted advertising to name just a few.

In computer science, current research on social networks is mainly focused, or relies on, online social networking platforms [92]. In fact, the set of relationships and interactions involving users of a social networking platform is a special case of a more general concept. The expression *social network* itself can refer to related but slightly different notions. Its introduction dates back to the 30's and has its roots at the intersection of different research areas, including sociology, social psychology and statistics. In a broad sense, a social network is the set of pairwise (or dyadic) relationships, directly or indirectly involving a set of individuals. When referred to humans, a relationship can reflect any type of interaction we are interested in, such as friendships in Facebook, follower - following relationships on Twitter or physical face-to-face interactions, to name a few. While the notion naturally applies to human societies and groups, it has a far more general reach and is commonly used to describe possibly complex interaction patterns involving different entities. At the same time, the expression social network can be used with a more specific, graph-theoretic meaning to denote a graph structure used to represent relationships of interest among individuals of a group. In this case, the expression denotes an abstraction of the underlying social phenomenon, considered with respect to some features, possibly not reflecting its full complexity. In this

perspective, the two main entities modeling a social network are *nodes* (or vertices), representing individuals, and *edges* (or links, arcs), which symbolize connections between nodes. In this thesis, we are mainly interested in studying human activities and behaviors. Therefore, in what follows, we refer to a social network as a network of humans, unless otherwise specified. Furthermore, when we speak of a social network, we may refer to either the social structure we intend to analyze or its mathematical abstraction, or both. Our intentions will be clear from context.

This thesis mainly focuses on dynamic aspects of social networks and their impact. Many real social networks (i.e., obtained from real data about social interactions) considered in the past were represented as static, possibly weighted and labeled networks (one can find many examples on Stanford Network Analysis Project [27]). This was either because the interactions described by those networks were mostly static in nature, or because the data collection process itself did not allow to capture the evolution of the relationships under consideration. This picture has changed in the recent past, with the widespread adoption of social networking platforms and services, which has made data about users' online activities available (albeit mostly to companies running these services) at an unprecedented scale and time-granularity. As a result, most recent studies about the dynamics of social networks have focused on the analysis of online social networks (e.g., [156, 145, 125]). Only a few attempts to collect data on real-world (non-virtual) social interactions have been made ([120, 87]), mainly because of logistical and technological difficulties. Distributing efficient and suitable devices to a significant population of individuals is in fact an expensive and time-consuming task. Furthermore, commonly used devices, such as mobile phones or tablets, only allow a limited accuracy in terms of users' position tracking and proximity estimation. In many cases, proximity among users is only inferred on the basis of their location, while the accuracy of localization technologies is usually in the range of few meters. Investigating these issues is of paramount importance, since the properties of the collected evolving social networks depend on these aspects and can possibly affect their accuracy in well representing real interactions over time.

Usually, specific techniques and metrics from *graph theory* [66] are adopted to study and assess the structure of social networks. More recently, the interdisciplinary area of *social network analysis* (SNA) [198] has proposed a rich set of graph-based tools to better capture and describe behaviors and properties of populations of individuals organized in social networks. However, also for the above mentioned reasons, SNA typically relies on static graphs as abstractions of the real, underlying social structure of interest, whereas most social networks evolve over time and change their structures under the pressure of social forces. As an example, consider the set of friendship relationships in

Facebook or the physical interactions among students in a school: they continuously evolve and change over time. In this work, we call *evolving network* any network that describes the temporal evolution of a set of relationships over a population of interest.

Time variability adds a further dimension in the analysis of (evolving) social networks, so that metrics commonly used to quantitatively describe important properties in static networks may need to be reconsidered, while others have been recently proposed [118]. There is growing attention in the research community towards the challenge of adapting or reformulating existing metrics for the analysis of evolving networks. However, such a "translation" process is not always obvious. As an example, one of the most prominent measures of network centrality is undoubtedly PageRank [164], initially proposed by Brin and Page to evaluate importance of Web pages. There are alternative ways to define PageRank that naturally lend themselves to the case of evolving networks, but their meaning and how to use them in practice are less obvious than it seems. For instance, it is not completely obvious that tracking changes in PageRank is the best way to describe evolution of centrality/authority in an evolving network. Furthermore, this approach poses computational challenges on huge networks, where real-time management of Pagerank can be prohibitively expensive in a centralized setting, raising the issue of distributing the computational load [168, 197].

On the other hand, the ability to accurately monitor users' physical proximity can provide insights into the dynamics of important social processes, such as the spreading of infectious diseases [193, 195], the circulation of information [79] through word of mouth [75] or the role of social influence and homophily in reaching consensus or solving collective tasks [147]. An interesting aspect is the role played by the network and its dynamic structure in producing or affecting phenomena of *collective intelligence* (an interesting overview of this area is presented in [19]). In this thesis, we focus on the *Wisdom of Crowds* effect, a phenomenon that has attracted the interest of researchers in the recent past and has been popularized in a best-selling book by James Surowiecki, appeared in 2004 [187]. Sir Francis Galton is credited for first observing this phenomenon in 1905 while attending a country fair. In particular, he observed that a crowd was collectively able to estimate the weight of an ox with high accuracy [109, 110]. Since then, this intriguing statistical effect has seen a number of applications, the most recent ones including predictive markets [59] and crowdsourcing [175]. Simultaneously and in part independently, the related study of models of opinion formation in the social sciences has taken increasingly hold among researchers after the seminal work by DeGroot [83].

The nature and description of evolving social networks, as well as the tools and strategies to collect data about evolving (physical) interactions and to summarize their network structure, are investigated in Part I of this thesis. In Part II of this thesis, we first explore the impact of time variability on notions of centrality originally introduced for static networks. We then explore the effect of a dynamic network structure on opinion formation and the wisdom of crowds effect in real-world scenarios.

Our contributions are described in more details in the paragraphs that follow.

Outline and contributions

This work presents the research activity and the corresponding original research results of a three-year PhD program in Computer Engineering. We briefly give below an outline of the main contributions of this thesis and a more comprehensive overview in the paragraphs that follow.

Data collection and analysis. This first line of research is functional to the other two. As previously mentioned, collecting data on evolving social networks resulting from real interactions in the physical world is of paramount importance for research in the area. At the same time, it can be a challenging task. This is particularly true for networks that describe physical interactions among humans. We have improved and optimized data collection techniques relying on sensor-based F2F tracking platforms. Furthermore, given the typically sheer sizes of the collected temporal datasets, we have introduced optimized data formats suitable to represent them. These contributions are presented in Part I.

Computing and maintaining centrality scores over evolving networks. As already mentioned, while the mathematical description of the main structural properties of social networks is well-established in the static case and relies on a rich graph-theoretic toolbox, many of the proposed notions do not obviously extend to evolving networks, or at least doing this requires some caution. In this thesis, we focused on the notion of centrality. On one hand, centrality is of paramount importance in many information retrieval tasks. On the other hand, it is a well-established concept in the static case, with rigorous and widely accepted mathematical formulations. This contribution is presented in Part II.

Dynamics of collective intelligence phenomena. Phenomena of collective intelligence involve exchange and processing of information among en-

tities sharing some common social structure. In many cases of interest, the underlying social structure is suitably described by an evolving social network. Studying the connections between the structure of the underlying network, the dynamics of information exchange and the computational properties of the resulting process is an interesting and challenging task. We have considered a well-known and only partially understood case of this problem, namely, the wisdom of crowds effect. This contribution is also presented in Part II.

Part I

Introduction to evolving social networks. In Chapter 1 we provide a general introduction about social networks and the corresponding methodologies to analyze them. This preliminary part allows to make the thesis self-contained. We introduce several formats to represent a social network and we then describe the most relevant metrics to analyze their properties. In the second part of the chapter we introduce social networks in dynamic contexts (i.e., networks evolving over time) and the corresponding file-formats used to represent their structure at each instant of time. Specifically, a novel network file-format, named Dynamic Network Format (DNF) [6], is introduced. While several options have been analyzed to try and represent evolving networks by employing techniques of approximation and aggregation, DNF leaves the information in a human-readable format, although compresses data using a technique based on gaps between time-steps. In the last part of the chapter we discuss generative models used to create random networks. Some of them have been then used to generate synthetic datasets.

F2F social networks and collection of physical interactions. In Chapter 2 we introduce Face-To-Face (F2F) social networks, namely evolving networks in which nodes are humans and edges between nodes dynamically appear as F2F interactions between humans occur. Collecting data about physical interactions allows the description of evolving social networks that have received increasing attention in recent times. Unfortunately, tracking F2F interactions is a non obvious task, mainly due to difficult logistics and physical problems. First of all, a suitable tracking technology must be identified, then members of a population should be recruited as volunteers to deploy a social experiment. At the beginning of our research activity we have devoted much effort to achieving this first goal and we mainly focused the attention on suitable technologies that could be used for our purposes. Following some survey activity, we selected the SocioPatterns sensing infrastructure [26, 22], based on the RFID technology, as a suitable candidate for collecting F2F social networks data from real-world scenarios. The next step was designing and programming an alternative MAC protocol able to deal with social experiments character-

ized by fast-changing F2F interaction patterns. In the same period, we carried out two first social experiments, one at our department and the other at the MACRO museum in Rome, involving more than 100 volunteers in both cases. Actually, these two first experiments were mainly useful to start investigating preliminary aspects of our research, including the efficiency of the default MAC protocol, the participation of people in our experiments and a first study of F2F social networks. Afterwards, we collected data from additional four social experiments, all conveyed to the study of the wisdom of crowds phenomenon.

Part II

Computation of centrality scores over evolving networks. Defining centrality scores for time evolving networks is a non obvious challenge. This allows to better understand how centrality of each single entity in the network evolves over time. On the other hand, tracking the evolution of centrality is extremely important to characterize the full evolving history of the network structure over time. In Chapter 3 we tackle this problem by considering notions of centrality which are consistent with PageRank in important cases. In particular, they amount to computing Pagerank over *static*, weighted, directed graphs that at any time reflect the "expected" topology of the evolving network under consideration. As next step, we analyze fully decentralized Monte Carlo algorithms for computing and maintaining PageRank-like *centrality scores* over evolving networks. We further show that, when the evolving network follows a process that satisfies a stronger property of homogeneity, the heuristics we propose continuously maintain an accurate estimate of the Pagerank computed over the current "expected" network. Obviously, real-life evolving networks may exhibit significant non-stationary properties. We therefore propose a modified heuristic which addresses some of the issues posed by non-stationary behaviors. Finally, we perform an extensive experimental analysis on both synthetic and real, publicly available, evolving network datasets. The obtained results support the validity and feasibility of the approach we propose.

The wisdom of crowds phenomenon. Chapter 4 first introduces the wisdom of crowds phenomenon [187], then reports all the findings about the deployed social experiments and the related implications on models. So far, the wisdom of crowds effect was treated as a phenomenon to be studied and measured on disconnected, predefined or complete social networks [147, 132, 151]. In other terms, participants have been usually constrained to talk to individuals whereby virtual or indirect relationships were supervised by authors. Most work making use of these conditions reveals that the social influence can undermine the wisdom of a crowd. Conversely, we gave users the possibility

to freely interact with any other participant. Our findings prove that a F2F social network can improve the wisdom of a group of individuals. This is partially a consequence of the fact that, in physical real-world scenarios, individuals usually choose trustworthy people or friends to interact with. Physical behaviors and expressions of a person often suggests if an individual is truly confident about his own answer or not. This gives a greater evidence on the reliability and wisdom of an individual. In the second part of the chapter we investigate the ability of some models proposed in social sciences to describe the dynamics of social influence in opinion formation dynamics. Specifically, we analyze the most prominent models, starting from the DeGroot's original one [83], to understand how well they describe the dynamics of opinion formation on physical real-world social networks. Before this thesis, no work dealt with opinion formation models running on this kind of social networks. In the end, after observing the models' behaviors and performance, we propose a new model who is a generalization of the one presented in [61]. Our model seems to provide better fit the reality, in all the four social experiments that we conducted. This improvement is mainly due to a finer-grained characterization of people's "stubbornness".

Part I

Evolving Social Networks

Chapter 1

Evolving Networks

In this chapter we introduce the concept of *Evolving Social Network* and its corresponding types, measures and representations. This is basically the starting point of our research activity and the fundamental part behind our contributions. Introducing the fundamentals of social and evolving networks is essential to better understand the next chapters, in which we will refer to the concepts described in this part of the thesis.

Before starting in describing what a network formally is, a question needs to be answered: "Why is modeling networks so important?". Nowadays, social networks pervade our social and economic lives. Let's think, for instance, about job opportunities, arrangement of meetings and spread of information. All these events exist thanks to the connection of multiple entities, which are the basic elements of a network. Social networks are also relevant in determining and understanding how particular infectious diseases spread, how we vote or which products we buy. It is not a mystery that online social networking services, such as Facebook or Twitter, collect user's personal information to propose targeted advertisings. Therefore recently, the research community has started focusing its attention on studying how social networks affect our behaviors and which kind of network structures is likely to emerge in a society.

A brief clarification on the used terminology. In literature, social networks are often considered as "static" networks [198, 123]. Since in the remainder of this thesis we also consider networks following an evolving process, we call *social network* (or static social network) a network formed by relationships among humans and not necessarily having information about time, while we name *evolving social network* (or simply, evolving network) a network that changes and evolves over time. However sometimes, a general network that does not involve humans as principal actors, can be nicknamed social network if it satisfies properties typical of social networks (e.g., small diam-

eter, high clustering coefficient) or describes social behaviors. Furthermore, we must clarify that in literature evolving networks are also called dynamic networks [116] (or graphs [81]), time-evolving graphs [146], or temporal networks [118, 153]. Actually, all these denominations refer to the same kind of networks, defined and largely described in Section 1.2.

1.1 Social networks in the "static" case

Since the world of concepts and techniques around social networks is very vast, before dealing with formal definitions and models, it is useful to start with an example that helps to give some ideas of what social networks are and how they can be modeled. The example we illustrate is a representation of the relationships among the characters of *Les Misérables*, a French historical novel by Victor Hugo published in 1862. Les Misérables, which is considered one of the greatest novels of the 19th century, follows the lives and interactions of several characters, in particular the struggles of ex-convict Jean Valjean, being the protagonist of the novel. The network of the relationships is shown in Figure 1.1. The data used to build the network are from [135]. The kind of representation depicted in the figure is called *social graph*, where the circles, usually named *nodes* or *vertices*, are individuals, while the *edges* (or *links*) denote relationships between them. Moreover, in this particular case, the thickness of the edge depicts the intensity of the relationship between two nodes. The more the line is thick, the more the two nodes have a strong relationship. As simple reference instance, let's look at nodes "Valjean" and "Cosette" at the center of the graph. Since Cosette is one of the character having a key role in the novel, and Valjean is her surrogate father, their relationship is quite intense, consequently their edge is rather thick. Similarly, we have other two strong relationships between node "Marius", who is the suitor of Cosette, and Valjean and Cosette, respectively.

The just described example gives an idea of how a social network can be structured and what kind of characteristics it may have. Although a specific social network is usually pretty different from others, a common language able to describe, represent and measure all of them exists. We first start giving some fundamental definitions and properties of networks.

Definition 1. A Graph is defined as an object $G = (V, E)$, where $V = \{1, \ldots, n\}$ is the finite set of vertices or nodes, while $E \subseteq V \times V$ is the finite set of edges linking nodes. The two edges $e = (i, j), e = (j, i) \in E$ (or, in an equivalent notation, $e_{i,j}, e_{j,i} \in E$) if and only if a connection between nodes i and j exists.

The canonical form of a network is the *undirected graph*, like the one depicted in Figure 1.1, in which two nodes are either connected or they are not.

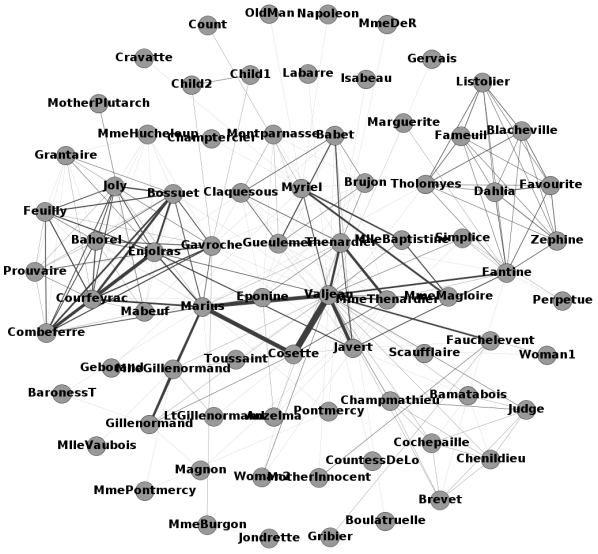


Figure 1.1: "Les Misérables" social network

However, a directed version of the social graph also exists, usually called *digraph* (i.e., directed graph). A digraph differs from an undirected graph for the presence of the directionality of its edges. In other terms, there are networks in which one node may be connected to a second without the second being connected to the first. Therefore, an edge $a = (i, j)$ is considered to be directed if a link from i to j exists, where j is called the *head* and i is called the *tail* of the edge. For convenience, we refer to a digraph as $D = (V, A)$ where A is the finite set of directed edges or arcs. In what follows the default is that the network is undirected, and we explicitly use the word digraph, or the notation $D = (V, A)$, when a directed network is considered.

Concepts and properties of networks. Networks are usually evaluated and categorized on the basis of some properties. While in Section 1.1.1.3 we examine in detail the essential measures used in graph theory and social network analysis, here we introduce some preliminary concepts and properties of networks.

Size of the network and cardinality of the edge set. As already discussed, a graph is an object formed by a set of nodes and a set of edges. The cardinality of the node set, namely $|V|$, denotes the size of the network, while the cardinality of the edge set, i.e., $|E|$ for graph or $|A|$ for digraph, strictly depends on the number of nodes and the density of the network. The maximum

number of edges in an undirected graph without self-loops is $|E| = \frac{n(n-1)}{2}$, while in the case of a digraph is $|A| = n(n-1)$, where $n = |V|$ is the total number of nodes. The difference between $|E|$ and $|A|$ is obvious because in a digraph each pair of nodes (i, j) could have two arcs, while in an undirected graph there could be only one edge.

Weight. The weight of an edge $w(e)$ denotes the intensity (or, in some cases, the cost or the length) of the relationship between the two nodes i and j in the edge e. A graph is a weighted graph if a weight is assigned to each of its edges [129]. Some authors call such a weighted graph a *network* [186]. However, we will use the world "network" as general term to refer to a graph or digraph. An example of weighted graph is the network shown in Figure 1.1.

Walks, paths and cycles. A *walk* in a network $G = (V, E)$ between two nodes i and j is a finite sequence of edges $(i_1, i_2), (i_2, i_3), \ldots, (i_{K-1}, i_K)$ such that $(i_k, i_{k+1}) \in E$ for each $k \in \{1, \ldots, K-1\}$, with $i_1 = i$ and $i_K = j$. Similarly, the *path* is defined as the walk, but with the additional constraint that each node in the sequence $i_1, \ldots, i_K$ must be distinct. The shortest path between nodes i and j is called *geodesic*, while the longest one is called *diameter*. Finally, a *cycle* is a walk that starts and ends at the same node and such that all other nodes are distinct. Therefore, the only node that appears more than once is the starting/ending node.

Common network structures. There are some particular network structures that have specific names and properties. A *tree* is an acyclic connected network, namely a graph without any cycles. A *forest* is a network such that every component (see next paragraph) is a tree. A forest with n nodes and k components has $n - k$ edges [165]. A specific case of forest is the star. A *star* is a network in which every edge in the network involves a specific node i. Finally, a *circle* is a graph having a single cycle, in which every node in the graph has exactly two neighbors.

Connectivity, clique and components. In many applications (e.g., spread of disease) there is the need to track if a node can reach any other node. Such a feature is related to the property of connectivity. A network is *connected* if every pair of nodes is connected by some path in the network. From this concept we can define a *component* of a network $G = (V, E)$ as a nonempty connected subnetwork $G' = (V', E')$ such that $0 \neq V' \subset V$ and $E' \subset E$. Therefore, a connected component is a maximal connected subgraph of G. In the case of a digraph, there is the need to distinguish between weakly connected component and strongly connected component. A *weakly connected component* is a maximal subgraph of a directed graph such that, for every pair of nodes i

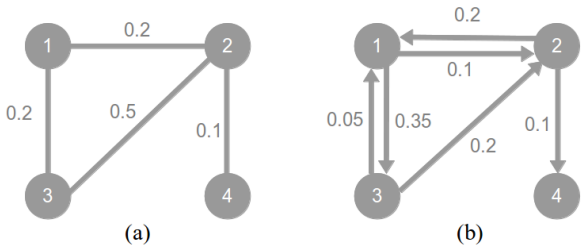


Figure 1.2: Two examples of networks: (a) undirected graph, (b) digraph

and j in the subgraph, there is an undirected path from i to j and a directed path from j to i. Vice versa, a *strongly connected component* is a maximal subgraph such that, for every pair of nodes i and j in the subgraph, both the paths from i to j and from j to i are directed. Finally, it is defined *complete network* a network having all possible edges, so that $E = \bigcup_{k=1}^{K-1} \bigcup_{z=k+1}^{K-1} (i_k, i_z)$ in undirected graphs, and $A = \bigcup_{k=1}^{K-1} \bigcup_{z=1, z\neq k}^{K-1} (i_k, i_z)$ in the case of digraphs. A *clique* [148] in $G = (V, E)$ is a subset of the vertex set $C \subseteq V$ such that for every pair of nodes in C, an edge connecting the couple exists. This is equivalent to saying that the subgraph induced by C is complete.

Neighborhood. Another fundamental concept of networks is the neighborhood. The *neighborhood* of a node i is the set of vertices that i is linked to. Formally, we define the neighborhood of node i belonging to an undirected graph, as $\mathcal{N}(i) = \{j : e_{i,j} = 1\}$. If we are dealing with weighted networks, then we define the neighborhood of i as $\mathcal{N}(i) = \{j : e_{i,j} > 0\}$. However, this notation is not very common because it is a modification of the original definition just for weighted networks. In the case of digraph we cannot refer to the concept of neighborhood as in undirected graphs because of the directionality of edges. Thus, we define the set of *successors* as $S(i) = \{j : e_{i,j} = 1\}$ and the set of *predecessors* as $P(i) = \{j : e_{j,i} = 1\}$.

1.1.1 Representation and measurement of networks

In this section we present some of the fundamentals on how networks are represented, measured and characterized. As we have already seen, in nature there are several kinds of networks, for this reason a general way does not exist to represent all of them. However, there are some representations widely used by many applications. Here we describe the two most popular ways of denoting networks.

The simplest formalism for representing a network is the *adjacency list*. An adjacency list is a collection of unordered lists, one for each vertex in the graph, describing the sets of neighbors. In other terms, the adjacency list of

node i is a list including i in the first position and then all the nodes belonging to the i's neighborhood: $\{i, j : j \in \mathcal{N}(i)\}$. Regarding digraphs, the adjacency list is defined considering the set of the successors: $\{i, j : j \in S(i)\}$. The corresponding adjacency lists of the example graph shown in Figure 1.2a is the following:

$$\{1, 2, 3\}, \{2, 1, 3, 4\}, \{3, 1, 2\}, \{4, 2\}$$

while the adjacency list of the digraph in Figure 1.2b is:

$$\{1, 2, 3\}, \{2, 1, 4\}, \{3, 1, 2\}, \{4\}$$

Clearly, the two lists are different due to the different type of the two networks. For example, in the case of the digraph, node 4 does not have any successor, therefore its list is formed by only itself. Vice versa, in the undirected graph, node 4 has as neighbor node 2 in its adjacency list.

The main benefits of adjacency lists are compactness and used space, which is $O(n + m)$, where $n = |V|$ and $m = |E|$. However, because of its simple formulation, the main limitation of an adjacency list is the inability to add further details about the network it represents. For instance, no information about weights can be included.

Another common way to represent graphs is the adjacency matrix. An *adjacency matrix* $\mathbf{M}$ is a $n \times n$ square matrix, where the element $m_{i,j}$ denotes the relationship between nodes i and j. Thus, $m_{i,j} = 1$ if nodes i and j are linked, $m_{i,j} = 0$ otherwise. Furthermore, in undirected graphs the equality $m_{i,j} = m_{j,i}$ is always true. Therefore, the adjacency matrix of an undirected graph is always *symmetric*, namely $M = M^T$. Vice versa, in digraphs it is likely that $m_{i,j} \neq m_{j,i}$, so no assumption about symmetry can be done. The adjacency matrix of the graph in Figure 1.2a is:

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

In the case a network is weighted, then the corresponding weighted adjacency matrix $\mathbf{W}$ should be considered, where each element $w_{i,j} \in \mathbb{R}^+$ indicates the weight of the edge (i, j). The graph shown in Figure 1.2a is a weighted networks, so its weighted adjacency matrix is:

$$\mathbf{W} = \begin{pmatrix} 0 & 0.2 & 0.2 & 0 \\ 0.2 & 0 & 0.5 & 0.1 \\ 0.2 & 0.5 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \end{pmatrix}$$

while the adjacency matrix of the weighted digraph shown in Figure 1.2b is:

$$\mathbf{W} = \begin{pmatrix} 0 & 0.1 & 0.35 & 0 \\ 0.2 & 0 & 0 & 0.1 \\ 0.05 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Notice that, in both the cases, the diagonal of $\mathbf{W}$ has all the elements $w_{i,i} = 0$. This is due to the fact that, in our examples, we are considering networks without self-loops. As opposed to adjacency lists, the adjacency matrix can represent weighted network; however, the space in memory to store a full adjacency matrix is $O(n^2)$. This could be a serious problem when considering networks having million of nodes.

1.1.1.1 Network file-formats

The simplest file-format to represent a network is the *comma-separated values (CSV)*, which stores tabular data in plain-text form. A CSV file consists of a certain number of records, separated by line breaks, where each record consists of fields separated by some character, most commonly a literal comma or tab (in this case the format is sometimes called TSV, acronym of tabular-separated values). CSV is widely used in many applications, but at present a general standard formalizing it does not exist, even though RFC 4180 [24] provides a sort of guidelines. Among its requirements we find that a) each line (optional for the last line) must end with (CR/LF) characters, b) an optional header record may be included, c) each record must contain the same number of comma-separated fields, d) a field may be double-quoted, e) fields containing a line-break, double-quote, and/or commas should be quoted and f) a double quote character in a field must be represented by two double quote characters.

The adjacency list relative to the undirected graph in Figure 1.2a can be stored in a CSV file in the following way:

```
1,2,3
2,1,3,4
3,1,2
4,2
```

while the adjacency matrix:

```
,1,2,3,4
1,0,0.2,0.2,0
2,0.2,0,0.5,0.1
3,0.2,0.5,0,0
4,0,0.1,0,0
```

where the first row and the first column of the CSV-matrix denotes the header of the file containing the nodes IDs.

Another popular notation is *GDF* [7], the file format used by GUESS [15]. It is built like a CSV, but it supports attributes to both nodes and edges. A standard file is divided in two sections, one for nodes and one for edges. Each section starts with a header line, which basically is the column title. Below, the GDF file representing the graph of Figure 1.2a:

```
nodedef>name INT
1
2
3
4
edgedef>node1 INT,node2 INT, weight DOUBLE
1,2,0.2
1,3,0.2
2,3,0.5
2,4,0.1
```

Since GDF only lists nodes and existing edges with corresponding weights, the used space is $O(n + 2m + t)$, where t is related to the weight attribute. Clearly, if the file explicits other attributes, such as labels, the space increases, but it always remains linear.

Graph Modelling Language (GML) [12] was one of the first attempt to propose a common language to model graphs. Indeed, the main goal of the authors was developing a format platform independent, easy to implement and able to represent arbitrary data structures, where it would have been possible to attach additional information to every object. A GML file consists of hierarchically organized key-value pairs. A key is a sequence of alphanumeric characters, while a value is either an integer, a floating point number, a string or a list of key-value pairs enclosed in square brackets. Graphs are represented by the keys “graph”, “node” and “edge”. The topological structure is modeled with the node’s “id” and the edge’s “source” and “target” attributes. The GML representation of the graph in Figure 1.2a is the following:

```
graph
[
node
[
id 1
label "1"
]
node
[
id 2
label "2"
]
node
[
id 3
label "3"
]
node
[
id 4
label "4"
]
edge
[
source 1
target 2
weight 0.2
]
edge
[
source 1
target 3
weight 0.2
]
edge
[
source 2
target 3
weight 0.5
]
edge
```

```
42 [
43 source 2
44 target 4
45 weight 0.1
46 ]
47 ]
```

GraphML [13] is another file-format for graphs. The main feature of GraphML is that it does not use a custom syntax, but it relies on XML [34]. The principal purpose of the authors was to make GraphML suitable for all types of programs and services processing graphs. Its main features include support for directed, undirected, and mixed graphs, hypergraphs, hierarchical graphs, graphical representations, references to external data, application-specific attribute data and light-weight parsers. The full GraphML syntax is defined by the GraphML schema [14]. Here, we just recap its basic elements. The file content is wrapped into a `graphml` element, while the whole network is included in the `graph` markup, in which it is also possible to set the graph type: `<graph edgedefault="directed">` or `<graph edgedefault="undirected">`. Nodes are denoted by the element `<node />` which usually includes the attribute `id`, and edges by the element `<edge />` containing the attributes `source` and `target`. Finally, each attribute is defined in a `key` element with an identifier, a name, a title, the scope for edge or node, and the type of data. The GraphML representations of the graph in Figure 1.2a is the following:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <graphml xmlns="http://graphml.graphdrawing.org/xmlns"
3  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4  xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
5  http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
6  <key id="d1" for="edge" attr.name="weight" attr.type="double
       "/>
7  <graph id="G" edgedefault="undirected">
8  <node id="1" />
9  <node id="2" />
10 <node id="3" />
11 <node id="4" />
12 <edge source="1" target="2">
13 <data key="d1">0.2</data>
14 </edge>
15 <edge source="1" target="3">
16 <data key="d1">0.2</data>
17 </edge>
18 <edge source="2" target="3">
19 <data key="d1">0.5</data>
20 </edge>
21 <edge source="2" target="4">
22 <data key="d1">0.1</data>
23 </edge>
24 </graph>
25 </graphml>
```

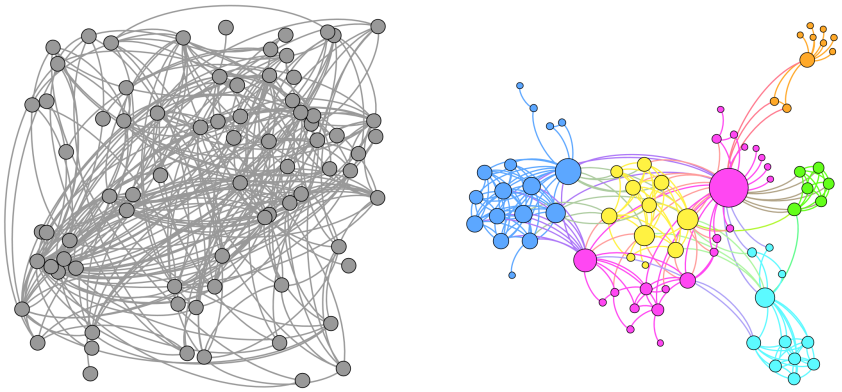
The last file-format we discuss is *Graph Exchange XML Format (GEXF)* [9]. GEXF is another XML-based language for describing graph structures, their associated data and dynamics. Indeed, with respect to previous formats, GEXF supports temporal dynamic networks. However, this feature will be better argued in Section 1.2.1.1 after introducing evolving networks. The GEXF project started in 2007 as reference network format for Gephi [8]. Its schema is defined in [10] where the 1.2 draft version is the recommended version to work with. As in GraphML, the graph structure is included in the element `graph`, and nodes and edges are denoted by `<node />` and `<edge />`, respectively. The fundamental difference with respect to GraphML is the grouping of the two entities. Indeed, all nodes are included in the element `<nodes>...</nodes>`, while all edges are grouped within the element `<edges>...</edges>`. Another important feature of GEXF is the default support for assigning weights to edges. Graphs in GEXF may be mixed as well. In other words, they can contain directed and undirected edges at the same time. The following code shows the GEXF representation of the graph in Figure 1.2a:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <gexf xmlns="http://www.gexf.net/1.2draft" xmlns:xsi="http
       ://www.w3.org/2001/XMLSchema-instance" xsi:
       schemaLocation="http://www.gexf.net/1.2draft http://www.
       gexf.net/1.2draft/gexf.xsd" version="1.2">
3  <graph mode="static" defaultedgetype="undirected">
4  <nodes>
5  <node id="1" />
6  <node id="2" />
7  <node id="3" />
8  <node id="4" />
9  </nodes>
10 <edges>
11 <edge source="1" target="2" weight="0.2" />
12 <edge source="1" target="3" weight="0.2" />
13 <edge source="2" target="3" weight="0.5" />
14 <edge source="2" target="4" weight="0.1" />
15 </edges>
16 </graph>
17 </gexf>
```

1.1.1.2 Software tools to analyze complex networks

Nowadays, the analysis of complex networks requires to analyze several metrics introduced in graph theory and in Social Network Analysis (SNA). However, before starting to measure a specific metric, sometimes it is useful to visually analyze the raw data. Indeed, thanks to the visualization, humans can easily find patterns in network structures as well as have a first visual perception of

the network structure. In Figure 1.3 we can see, for instance, two different ways of visualizing the same network of Les Misérables. Specifically, Figure 1.3a shows the network using a random layout, while Figure 1.3b arranged the graph through the ForceAtlas2 layout [124]. It appears quite obvious that the amount of information provided by Figure 1.3b is higher than the one provided by the randomly drawn graph. However, this kind of analysis usually requires an exploratory process [171]. Already in 1996, Scheiderman [181] summarizes the general process steps of graph visualization: "Overview first, zoom and filter, then details-on-demand". Therefore, in order to satisfy the requirements for such a process, visualization tools should support high quality layout algorithms, data filtering, clustering, statistics and visualization features.



(a) Random layout (b) ForceAtlas2 layout

Figure 1.3: Two different kinds of visualizations of the network Les Misérables: (a) shows a representation using a random layout, while (b) employs the ForceAtlas2 layout [124] where nodes are resized according to their degree and colored according to their maximum modularity class.

In the following we briefly list and introduce the most popular software tools and libraries for visualizing and analyzing complex networks.

- *NetworkX* [20] is a Python software package for creating, manipulating and studying structures and dynamics of social, biological, and infrastructure networks. NetworkX includes several mechanisms to easily work with graphs and, at present, is one of the most popular libraries among researchers.
- *iGraph* [16] is another software package that can be programmed in R, Python and C/C++. It includes a collection of network analysis tools with the emphasis on efficiency, portability and ease of use.

- *R* [28] is a language and environment for statistical computing and graphics. Due to its statistical nature, it is very used in the research community to study and analyze networks.

- *D3.js* [4] is a JavaScript library for manipulating documents based on data. It is one of the most recent tools available to work with raw data in order to transform them into a structured format.

- *sigmajs* [25] is another JavaScript library dedicated to graph drawing. It makes easy to publish networks on the Web and allows developers to integrate network exploration in rich Web applications in order to make network manipulation smooth and fast for the user.

- *GUESS* [15, 43] is a visualization and analysis tool based on Gython, a domain-specific embedded language which supports operators for directly working on graph structures in an intuitive way.

- *Pajek* [23, 57] is one of the first visual exploratory tools for graph visualization and analysis. It is freely available for noncommercial use.

- *Cytoscape* [3, 180], shown in Figure 1.4, is an open source software platform for visualizing and analyzing molecular interactions and biological networks. Cytoscape was originally designed for biological research, now it is a general platform for complex network analysis and visualization. At present, it is one of the most powerful tool for network analysis.

- *Gephi* [8, 56] is one of the youngest graph-viz projects for the manipulation and visualization of complex networks. Specifically, it is an open source software based on the NetBeans platform (netbeans.org), specialized in graph analysis and visualization thanks to several statistical plug-ins and a 3D render engine that speeds up the exploration and real-time rendering. In addition, one of the most powerful features of Gephi is the *timeline* component that allows to dynamically explore evolving networks. Gephi runs on Windows, Linux and Mac OS X and it is open-source and free. A screenshot of the main user interface is shown in Figure 1.5. Actually, Gephi has been our favorite tool to draw and measure networks throughout our research activity.

- *Gexf4j* [11] is a Java library to create and write GEXF files which can be used to visualize graphs in Gephi or other GEXF-supporting application. Gexf4j was initially designed by Javier Campanini and released under the Apache License 2.0. However, the Campanini's release only supported the old GEXF schema (1.1 draft), so in 2012 we decided to take over the project, under the same license, and develop future versions. At present,

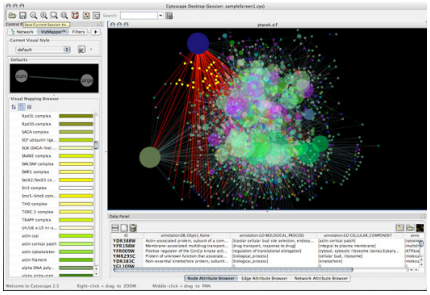


Figure 1.4: Cytoscape

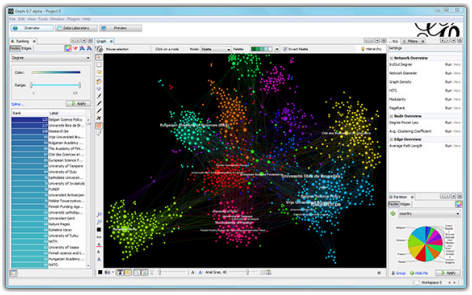


Figure 1.5: Gephi

gexf4j is continuously maintained and updated and the source code is hosted on Github [11]. The binary releases are published on Maven Central Repository [18] so that any Java developer can easily include gexf4j in her own Maven projects. In 2014 the library was downloaded more than 1000 times from Maven, while the Github repository has been cloned thousand and thousand of times since the beginning of the project.

1.1.1.3 Network measures

In mathematics and computer science, graph theory is a research area studying properties of graphs. We have seen that networks can be visualized and analyzed thanks to software tools, but, to make this possible, topological measures describing the main structural properties of the graph [198, 86, 123] have to be investigated. In the following we review some of them.

Degree. One of the basic metrics of graph theory is the degree. Let $G = (V, E)$ be a non-empty graph, where $V = \{1, \ldots, n\}$ is the node set and $E \subseteq V \times V$ the edge set. We have seen that the neighborhood of a vertex i in G is denoted as $\mathcal{N}(i)$. Then, the *degree* of node i, denoted $d(i)$, is:

$$d(i) = |\{j : e_{i,j} = 1\}| = |\mathcal{N}(i)|$$

In other words, the degree of node i is the number of its incident edges. A vertex of degree 0 is called *isolated*. If all the vertices of G have the same degree k, then G is called *k-regular* and we can speak of the degree of the graph. In the case of a digraph $D = (V, A)$, the degree of a node i is the sum of the *in-degree*, defined as $d^-(i) = |P(i)|$ and the *out-degree*, namely $d^+(i) = |S(i)|$. A vertex i having $d^-(i) = 0$ is called *source*, while a vertex with $d^+(i) = 0$ is called *sink*.

Sometimes, in addition to analyze the single node degrees, it is also useful to examine the average degree of the graph to better understand what kind of structure shapes the network. Thus, the number

$$d(G) = \frac{1}{|V|} \sum_{i \in V} d(i)$$

denotes the *average degree* of G. This quantity can be also expressed as $\varepsilon(G) = \frac{2|E|}{|V|}$. Indeed, counting all the nodes degrees in G is equivalent to count twice every edge:

$$2|E| = \sum_{i \in V} d(i) \tag{1.1}$$

(1.1) is called *degree sum formula* and from which it follows that:

$$\varepsilon(G) = d(G)$$

This leads to the following proposition, sometimes named *handshaking lemma*.

Proposition 1. The number of vertices of odd degree in a graph is always even.

Proof. A graph G has $\frac{1}{2} \sum_{i \in V} d(i)$ edges, so $\sum d(i)$ is an even number. □

Clearly, in the case of digraphs, if we split the degree in in-degree and out-degree, then the degree sum formula becomes:

$$|A| = \sum_{i \in V} d^-(i) = \sum_{i \in V} d^+(i)$$

However, if we consider the total degree $d(G)$ as the sum of in-degree and out-degree:

$$\sum_{i \in V} d(i) = \sum_{i \in V} d^-(i) + \sum_{i \in V} d^+(i)$$

then we get back to (1.1). Lastly, if for every node $i \in V$ we have $d^+(i) = d^-(i)$, then the graph is called *balanced digraph*.

Weighted degree. If we now focus on weighted networks, a fairer metric to consider is the *weighted degree*, sometimes called *node strength*. The weighted degree is defined similarly to the degree, but the weighted version also takes into account the weight of incident edges. Let $G = (V, E)$ be a non-empty weighted graph, where $V = \{1, \dots, n\}$ is the node set and $E \subseteq V \times V$ is the

edge set. Then, $\mathbf{W}$ is the $n \times n$ adjacency matrix of G having a weight $w_{i,j}$ on each edge (i, j). Thus, the weighted degree of node i is defined as:

$$wd(i) = \sum_{j \in \mathcal{N}(i)} w_{i,j}$$

In other terms, the weighted degree of a node is the sum of weights of the edges incident on that node. As for digraphs, the weighted in-degree and out-degree are defined by just replacing $\mathcal{N}(i)$ with $P(i)$ and $S(i)$ to the previous formula, respectively.

Density. Another fundamental metric giving an idea on the structure of the graph is the *density*. Let $G = (V, E)$ be a non-empty graph, where $V = \{1, \ldots, n\}$ is the node set and $E \subseteq V \times V$ is the edge set. The cardinality of V is $|V| = n$, while the total number of edges in E is $|E| = m$. We recall that the maximal number of edges of E is $\frac{n(n-1)}{2}$. Therefore, we define the density as the fraction of existing edges in E, namely m, and the maximal number of edges of the graph:

$$\lambda(G) = \frac{2m}{n(n-1)}$$

In the case of a digraph $D = (V, A)$, being $n(n-1)$ the maximal number of arcs in A, the density is:

$$\lambda(D) = \frac{m}{n(n-1)}$$

Clustering coefficient. The clustering measures the tendency of the neighbors of a node i to be connected to each other. In other terms, the *clustering coefficient* [199] is a measure of the degree to which nodes in a graph tend to cluster together. Let $G = (V, E)$ be a non-empty graph, where $V = \{1, \ldots, n\}$ is the node set and $E \subseteq V \times V$ is the edge set. The clustering coefficient $\vartheta(i)$ of a node i is computed as the ratio between the actual number of edges between its neighbors and the value of the maximum possible edges in the neighborhood $\mathcal{N}(i)$:

$$\vartheta(i) = \frac{2|\{e_{v,u} : v, u \in \mathcal{N}(i)\}|}{|\mathcal{N}(i)|(|\mathcal{N}(i)| - 1)}$$

This measure is meaningful only for $|\mathcal{N}(i)| > 1$. If $|\mathcal{N}(i)| = 1$ then we consider $\vartheta(i) = 0$. If we now look at a digraph $D = (V, A)$, since the number of the maximum possible arcs in the neighborhood $\mathcal{N}(i)$ is $|\mathcal{N}(i)|(|\mathcal{N}(i)| - 1)$, the clustering coefficient of node i is:

$$\vartheta(i) = \frac{|\{a_{v,u} : v, u \in \mathcal{N}(i)\}|}{|\mathcal{N}(i)|(|\mathcal{N}(i)| - 1)}$$

The clustering coefficient of the network, which measures the overall degree of clustering of the graph, is defined as the average on the local clustering coefficients of all the vertices:

$$\langle\vartheta\rangle = \frac{1}{n}\sum_{i=1}^{n}\vartheta(i)$$

where $n = |V|$ is the total number of nodes.

Modularity. The main difference between a completely random graph and a real-world social network is that the latter often exhibits interesting properties regarding its corresponding structure. One of these features is the community structure, namely the division of nodes into groups within which the network connections are dense, but between which they are sparser. This property is better known as *modularity* [161, 160]. Its name derives from the measurement of the division of the network into "modules" (or communities).

Consider a network divided in k communities. Let $\mathbf{W}$ be a $k \times k$ symmetric matrix whose element $w_{i,j}$ is the fraction of all edges in the network that link nodes in community i to nodes in community j. We can define the modularity as:

$$Q = \sum_i \left(w_{i,i} - \left(\sum_j w_{i,j} \right)^2 \right)$$

where Q lies in the range $[-0.5, 1)$. This quantity measures the fraction of the edges in the network that connect nodes in the same community minus the expected value of the same quantity in a network with the same community divisions but random connections between the nodes. A value of $Q \leqslant 0$ indicates a random behavior of the network, while values approaching $Q = 1$, which is the maximum, indicate strong community structure.

Centrality. One of the most important indicator in SNA is the *centrality*, that is a measure able to identify the most influent vertices within a social network. In other terms, centrality indices have a duty to answer the following question: "What characterizes an important vertex?". Actually, the word "important" has different meanings, depending on the considered context. For this reason, several centrality indices exist [103, 67, 68], such as degree centrality, closeness centrality, betweenness centrality and eigenvector centrality. Other two famous indices of centrality are Katz centrality [130] and Pagerank

[164].

The *degree centrality* is the historically first and conceptually simplest index belonging to the centrality class. It indicates how well a node is connected in terms of direct[1] connections. Let $G = (V, E)$ be a non-empty graph, where $V = \{1, \ldots, n\}$ is the node set with cardinality $|V| = n$ and $E \subseteq V \times V$ is the edge set. The degree centrality of a node i is simply:

$$C_d(i) = \frac{d(i)}{n-1}$$

Of course, the degree centrality misses a lot of other aspects of a network. For instance, it does not measure how well-located a node is in a network. In other words, a node with a high degree centrality could be located at the border of a graph and linked to a certain number of nodes already well-linked to other vertices, and thus having low relevance in terms of structure, while another node with low degree centrality could be located in the middle of the network linking two subgraphs, and so being more important for the connectivity of the whole graph.

As usual, in the case of digraphs, we must diversify between in-degree and out-degree centrality. However, the concept remains the same.

The *closeness centrality* measures how close a given node is to any other node. In other words, the closeness centrality of a node i is the reciprocal of the sum of the geodesics from i to all $n-1$ other nodes. Since the sum of distances depends on the number of nodes in the graph, closeness is normalized by the sum of the minimum possible distances $n-1$:

$$C_c(i) = \frac{n-1}{\sum_{j=1, i \neq j}^{n-1} \ell(i,j)}$$

where $\ell(i,j)$ is the number of links in the geodesic between i and j.

An index of centrality that better measures how well situated a node is in terms of the paths in which it acts as a bridge is the *betweenness centrality*. It quantifies the number of times a node lies two other nodes along geodesics. It was proposed by Freeman [102] as a measure for quantifying the control of an individual on the communication between other humans in a social network. Let $\sigma(i)_{k,j}$ denote the number of geodesics between k and j that i lies on, and let $\sigma_{k,j}$ be the total number of shortest paths between k and j. We can estimate how central a node i is in terms of connecting two other nodes by

[1] In this context, the word "direct", which is different from "directed", does not refer to the directionality of edges, but to all those links directly incident on a node.

computing the ratio $\frac{\sigma(i)_{k,j}}{\sigma_{k,j}}$. Averaging this ratio across all pairs of nodes, the betweenness centrality of a node i is:

$$C_B(i) = \sum_{k \neq j, i \notin k,j} \frac{\sigma(i)_{k,j} / \sigma_{k,j}}{\frac{(n-1)(n-2)}{2}}$$

where $n = |V|$ is the total number of nodes.

The *eigenvector centrality* is another measure of node's influence proposed by Bonacich [65]. The basic idea is that the centrality of a node is proportional to the sum of the centrality of the vertices in its neighborhood $\mathcal{N}(i)$. For a given graph $G = (V, E)$, let $\mathbf{M}$ be the adjacency matrix, so that $m_{i,j} = 1$ if node i is linked to node j, namely i and j are neighbors, and $m_{i,j} = 0$ otherwise. The eigenvector centrality $C_E(i)$ of node i is defined as:

$$C_E(i) = \frac{1}{\lambda} \sum_{j \in \mathcal{N}(i)} m_{i,j}\, C_E(j)$$

or equivalently, in matrix notation:

$$\lambda\, \mathbf{C_E} = \mathbf{M}\, \mathbf{C_E} \tag{1.2}$$

where eigenvalue λ works as a proportionality factor. (1.2) identifies the eigenvector equation (see Appendix A.1 for background on eigenvectors and eigenvalues), in which $\mathbf{C_E}$ is a right (or column) eigenvector. Generally, there could be many different eigenvalues λ for which an eigenvector solution exists. However, the requirement of all the entries in the eigenvector must be positive implies (by the Perron–Frobenius theorem [173, 107], Perron root) the existence of a real positive eigenvalue λ strictly greater in absolute value than all other eigenvalues, so it must be the largest eigenvalue of $\mathbf{M}$, and $\mathbf{C_E}$ the corresponding eigenvector. One method to compute the eigenvector centrality is *Power iteration* [155] (see Appendix A.2 for further information).

A variant of the eigenvector centrality is the *Katz centrality*, which quantifies the number of all nodes that can be connected through a path, penalizing the contribution given by distant nodes. It is a generalization of the degree centrality which instead measures the number of direct neighbors. For a given graph $G = (V, E)$ with $|V| = n$ the number of nodes in the network, let $\mathbf{M}$ be the adjacency matrix. The Katz centrality is defined as:

$$C_K(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k (M^k)_{j,i}$$

where α is an attenuation factor ranging between 0 and 1. The k-th power of $\mathbf{M}$ gives an adjacency matrix of a graph with the same vertex set and an edge between two vertices if and only if there is a path of length at most k between them [100].

PageRank. In recent years, one of the most important measure of nodes' centrality in static networks has certainly been *PageRank* [164, 140]. The basic metaphor behind PageRank [69] considers a *random surfer* that starts from a randomly chosen web page. At every page she visits, the random surfer either gets bored and quits navigation with a fixed probability, or she selects one of her outgoing neighbors uniformly at random. If the page has no outgoing links (i.e., it is a *sink*),the surfer jumps to a page selected uniformly at random in the network. The rank of a page according to the PageRank algorithm is the probability that a random surfer stops at that page. We denote by α the probability that the surfer continues her random walk (the *damping factor*). We must highlight that PageRank is usually used in digraphs, where the directionality of edges is fundamental for the algorithm. Vice versa, in undirected graphs, it can be shown that PageRank is statistically close to the degree distribution of the graph [172].

Let $\mathcal{N}(i)$ be the set of neighbors of a node i and $d(i) = |\mathcal{N}(i)|$ the degree of i. The *transition matrix* [179, p. 114] $\mathbf{Q}$ of G is defined as follows:

$$q_{i,j} = \begin{cases} \frac{1}{d(i)}, & \text{if } d(i) \neq 0 \text{ and } j \in \mathcal{N}(i); \\ 0, & \text{otherwise.} \end{cases}$$

In addition, we denote by $\mathbf{A}$ the *modified transition matrix* of G, corresponding to removal of sinks, namely the stochastic matrix such that $a_{i,j} = q_{i,j}$ if i is not a sink, $a_{i,j} = \frac{1}{n}$ otherwise.

Given these definitions, the Pagerank vector $\boldsymbol{\pi}$ of G is the stationary distribution (equivalently, the main left eigenvector) of the ergodic Markov chain [179, Theorem 4.2, p. 119] corresponding to the following stochastic matrix: $\mathbf{P} := \alpha\mathbf{A} + (1-\alpha)\left(\frac{1}{n}\mathbf{1}\mathbf{1}^T\right)$, where $\mathbf{1}$ denotes the 1-value column vector, while $\mathbf{1}^T$ is the row vector with unit components.

A more extensive discussion about PageRank is given in Chapter 3, in which we will also describe further issues.

1.2 Social networks in dynamic contexts

So far, we focused our attention on static networks, i.e., graphs that do not change over time. An advanced class of graphs is that of *evolving networks*. This kind of networks are networks that change as a function of time. For this reason, they are sometimes called *time-evolving networks* [146] or *temporal*

networks [118]. Evolving networks are very common in our daily life. Indeed, almost all real-world networks evolve over time. A good example exactly includes all those networks formed on social networking services, where people make and lose relationships over time, thus creating and destroying edges. Instances involving the biological domain are networks in which the state of each interaction between cell components, such as DNA and proteins, changes during the execution of biological processes [128, 138]. A further biological network that changes over time is the metabolism; here evolving networks can potentially capture its dynamics [74]. Regarding neural networks, Eguiluz et al. [89] studied the structure of brain functional networks using functional magnetic resonance imaging. They discovered that this network follows the power law degree distribution, with high clustering coefficient and small average path lengths. Evolving networks have applications in economics as well. A network representation of the stock market is analyzed in [63]. Vertices in this network are the instruments in the stock. Other examples are computer networks, where an edge is added every time a connection is established between two devices, and citations networks, namely the network of scientific papers' citations, in which a new edge is added when a new paper cites another one. Finally, as we will better see in the next chapter, proximity patterns of humans are emerging in recent years [87, 72]. This kind of evolving networks are important to understand and study the spread of diseases and word-of-mouth spreading of information. However, evolving networks are not all equals. The just cited examples suggest how those networks have different nature. Some networks are incremental (i.e., an edge or a node is never removed), while other networks allow node or link to be created and destroyed over time. A general way to characterize an evolving network is based on four fundamental actions occurring at every time-step, each of them with a given probability: a) add a link, b) add a node, c) remove a link, d) remove a node. Another common way to define an evolving networks, and that we will adopt throughout this thesis, is considering them as sequence of successive static graphs $\mathcal{G} = G_0, G_1, G_2, \ldots$ over the same vertex set V [48].

Definition 2. Let $G_0, G_1, G_2, \ldots$ be an infinite sequence of graphs over the same vertex set V and different edge sets E_i. We define *evolving network* a network $\mathcal{G} = \{G(t)\}_{t \geqslant 0}$, where $G(t) = (V, E(t))$ denotes the *snapshot* of the graph at time t. We call *history* the full evolving process.

The above definition implicitly forces the set of nodes to be the same over time. Although this characteristic may be a little binding, actually real-world scenarios usually involve finite evolving networks, where the node set V is well-known from the beginning.

A simple example of an evolving network $\mathcal{G}$, having a history of four time-steps and depicting physical interaction between users, is illustrated in Fig-

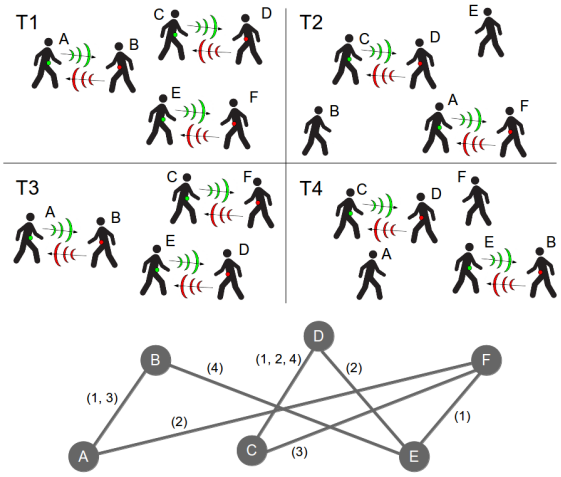


Figure 1.6: A simple instance of an evolving network

ure 1.6. At the top of the figure the full evolving process is shown, where each quadrant indicates the snapshot $G(t)$ at the corresponding time-step $t = 1, 2, 3, 4$. The size of the network is $|V| = 6$, while the vertex set is $V = \{A, B, C, D, E, F\}$. The edge set $E(t)$ changes at each time-step and it is never empty. We have $E(1) = \{(A, B), (C, D), (E, F)\}$, $E(2) = \{(C, D), (A, F)\}$, $E(3) = \{(A, B), (C, F), (E, D)\}$, $E(4) = \{(C, D), (E, B)\}$. Being an evolving network of undirected graphs, the order of the two nodes in the edge does not matter. At the bottom of the figure it is shown the corresponding "aggregate graph" where the labels on the edges indicate the time-steps each couple interacted. For instance, the edge linking nodes C and D occurred in $t = 1, 2, 4$. This kind of representation introduces the new concept of aggregation.

Definition 3. Given an evolving network G and a factor of aggregation Δ, we define *aggregate snapshot* at time t the weighted graph $G_\Delta(t)$ obtained by aggregating and averaging the Δ most recent snapshots, namely: $\frac{1}{\Delta}\sum_{\ell=0}^{\Delta-1} G(t-\ell)$. For every Δ the sequence $\mathcal{G}_\Delta = \{G_\Delta(\ell\Delta)\}_\ell$, where $\ell = 1, 2, \ldots$, defines a derived evolving network that we call *aggregate evolving network*.

The evolving network in Figure 1.6, having a total of four snapshots, could be aggregated by selecting a $\Delta \in \{1, 2, 3, 4\}$. Of course, for $\Delta = 1$ the aggregate evolving network is equivalent to the original evolving network. Oppositely, selecting a $\Delta = 4$, we obtain an aggregate evolving network $\mathcal{G}_\Delta$ coinciding with the unique snapshot G_Δ, where each edge is weighted by averaging the four consecutive snapshots. Thus, the edge (C, D) has a weight of $\frac{3}{4}$ since it occurs in three of four snapshots, the edge (A, B) has a weight of $\frac{1}{2}$, while the

other existing edges have a weight of $\frac{1}{4}$. Obviously, the aggregation of snapshots leads to a loss of information regarding the history, since the aggregate evolving network $\mathcal{G}_\Delta$ no longer includes information about the evolution of each snapshot it aggregated. This could be, in some applications, a matter to be reckoned with. However, in other kinds of scenario, the aggregation may be very useful, especially in networks too sparse to be analyzed at each single time-step.

1.2.1 Representing and measuring evolving networks

We have seen that static networks can be represented in different way using several network file-formats. A good tool to be adopted when the size of the matrix is not so big is the adjacency matrix in a CSV file. This allows to apply some of the most common metrics directly on the matrix. However, the representation of evolving networks is a little trickier, since for each time-step t we could have a different network structure over the same vertex set V. Assuming to use a similar approach to static networks, we would have the same number of files as the the total number of time-steps. Of course, this solution is not very efficient because we could have many repetitions of the same data, such as the set of nodes and recurrent edges. For this reason, evolving networks are usually represented in other different ways. A part of the research community focuses the attention on certain kinds of techniques of approximation and compression [81, 116, 146] to represent and analyze large evolving networks. However, since our real-world evolving networks are limited in size and time, we are interested in selecting an efficient network file-format to describe the full history of the network without loosing any kind of information. In the next subsection we compare the features given by GEXF to represent evolving network with DNF [6], a new network file-format realized during our research activity. Finally, in Section 1.2.1.2 we describe the most common metrics adopted in network analysis to measure dynamic evolving networks.

1.2.1.1 Evolving network file-formats: GEXF and DNF

In Section 1.1.1.1 we described several network file-formats to represent static networks. One of the most flexible, with a lot of additional features, is certainly GEXF. In this section, we resume the description of the format by introducing the markup `<spell />`, which is the element in charge of adding information of time to represent evolving networks. Specifically, `<spell />` can have a `start` or a `start/stop` attribute in which it is possible to specify a time-range an element exists. In other words, for each node or edge we can specify the whole list of time intervals (i.e., several `<spell />` grouped

in `<spells>...</spells>`) corresponding to their presence in the evolving graph. The data types allowed in `start` and `stop` attributes are *double*, *date* and *dateTime* [30]. Let's see an example:

```
<gexf ...>
...
<graph mode="dynamic" timeformat="double">
<edges>
<edge source = "1" target = "2">
<spells>
<spell start="1335090239" end="1335090241" />
<spell start="1335090242" end="1335090243" />
<spell start="1335090244" end="1335090249" />
</spells>
</edge>
</edges>
</graph>
</gexf>
```

In the code above the edge $(1, 2)$ appears in three different time ranges denoted by UNIX timestamps. Thus, it appears clear that listing spell times in a GEXF file allows to represent an evolving graph.

Unfortunately, we soon realized that our graphs, although they are not so large in size and time, resulted in GEXF files too big to be imported and computed in Gephi. Indeed, because of the verbosity of XML, even representing a short evolving history requires reasonable resources in terms of space. For this reason, we thought about a new way to represent evolving networks. The finding was *Dynamic Network Format (DNF)* [6]. Its efficiency has allowed us to solve all the issues regarding the representation of our real-world social networks. The idea behind DNF is very simple: it tries to reduce the file size thanks to *gaps* between time-steps. For instance, suppose to have the following dynamics and assume that the global initial time-step of the evolving graph is the UNIX timestamp 1335090220:

$$(1, 2) \in \{1335090242, 1335090243, 1335090246, 1335090247, 1335090249\}$$

then, DNF fixes the initial global timestamp in the header of the file and the initial edge gap will be the difference between the initial timestamp in which the edge occurred and the global initial timestamp, namely $22 = 1335090242 - 1335090220$. The next gap is calculated by subtracting the previous timestamp to the current timestamp, and so on. For instance, as for the second gap, we have $1 = 1335090243 - 1335090242$, while the last gap is $2 = 1335090249 - 1335090247$. The corresponding DNF string is:

```
1 [1,2] (22,1,3,1,2)
```

In addition, DNF manages continuous timestamps to avoid redundancy of gaps, further reducing the file size. Continuous timestamps are represented by the + symbol followed by the number of continuous instants (excluded the first one). As example, suppose to have the following history for the edge $(1, 2)$

$$(1,2) \in \{1335090249, 1335090251, 1335090252, 1335090253, 1335090254, 1335090259\}$$

where the sequence of continuous timestamps starts with 1335090251 and ends with 1335090254. Then, the corresponding DNF string is:

```
1 [1,2] (29,2,+3,5)
```

where 29 is the first gap between timestamp 1335090249 and the global initial timestamp, 2 is the gap between timestamp 1335090251 and 1335090249, +3 is a continuous gap for timestamps 1335090251 (excluded, since it is the starting point), 1335090252, 1335090253, 1335090254, and the last gap 5 is the difference between 1335090249 and 1335090254. The reason why timestamp 1335090251 is not included in the continuous gap is that it is already "taken" by the gap 2.

Graphs represented by DNF output text file of size at least one order of magnitude less than the ones built using the GEXF file-format. For instance, regarding one of our social experiments, described in the next chapter, the corresponding DNF file is 2 MB, while the GEXF file amounts to around 50 MB. The reduced size allowed us to import and analyze the dynamics of our evolving networks in Gephi. To do this, we wrote a Gephi plug-in [5] able to load DNF graphs. For the complete DNF syntax and further examples see Appendix A.3.

1.2.1.2 Measures of evolving networks

As already seen in previous sections, concepts, properties and measures in static networks are largely widespread and well-defined. In essence, most of them are based on connections between neighboring nodes or between specific sets of nodes. However, in evolving networks, because of the additional parameter of time, part of those measures must be revisited. Some measures can be directly applied on the aggregate evolving network $\mathcal{G}_\Delta$, while others need to consider the temporal-topological structure of the network. For instance, paths in evolving networks must necessarily follow the sequences of links activating one after the other in time. In literature, these kinds of paths are usually called *time-respecting paths* [131] or *journeys* [203]. That means

a path from i to k via j exists only if the first contact on (j,k) occurs immediately after the last contact on (i,j). In other words, the edge[2] $e_{i,j}^{(t)}$ must precede the edge $e_{j,k}^{(t+1)}$, namely they are in ascending order in terms of time.

We have introduced the term *contact* because, as we will better see in the next chapter, we often refer to an evolving graph as a real-world social network, namely a network of individuals interacting each other over a certain amount of time. Thus, links are often considered proximity contacts between humans. Other examples of revisited measures and further metrics on dynamic evolving networks are:

- the *connectivity*, where Nicosia et al. [162] differentiate between the term "strongly connected" (i.e., there is a directed, time-respecting path connecting i to j and vice versa) and the term "weakly connected" (i.e., there is an undirected, time-respecting path between i and j);
- the *latency* (or temporal distance) [166], which measures the shortest time within which i can reach j;
- the *temporal closeness centrality* [189], which measures how quickly a node may on average reach other vertices. It is defined as: $C_c^{(t)}(i) = \frac{n-1}{\sum_{i \neq j} \delta^{(t)}(i,j)}$, where $n = |V|$ is the total number of nodes and $\delta^{(t)}(i,j)$ is the latency between i and j at time t;
- the *temporal betweenness centrality*, which is redefined for temporal networks by Tang et al. [189]. They just add the dependence on time and count the fraction of fastest time-respecting paths that pass through a certain node;
- the *inter-contact time*, which measures the time interval between two continuous contacts of a same pair of nodes. Computing the inter-contact time distribution allows to observe what is the probability for a contact to be repeated after a certain amount of time.
- the *intra-contact time*, which measures the amount of time two nodes communicate. The resulting distribution is quite relevant to determinate the spreading capacity of a message;
- the *average temporal density*, which is defined as:

$$\Lambda = \frac{1}{T} \sum_{t=1}^{T} \lambda(t) = \frac{1}{T} \sum_{t=1}^{T} \frac{2|E(t)|}{|V|(|V|-1)}$$

[2] We indistinctly use $e_{i,j}^{(t)}$ or $e_{i,j}(t)$ to denote $e_{i,j}$ at time t.

where T is the total number of time-steps, $\lambda(t)$ is the network density at time-step t and $|E(t)|$ is the number of edges the evolving network has at time t.

Correlations and recurrence plot. One of the measures we have mostly adopted in evolving networks is the *correlation* between each pair of snapshots in the network $\mathcal{G}$. Specifically, we used some well-known correlation coefficients to compare the snapshot $G(t)$ to the snapshot $G(t+\Delta)$, such that $t \geqslant 0$ and $\Delta > t$. Such a measure allows to understand how much dependent the two single snapshots are. The two coefficients we have employed are the *Pearson's* ρ_P [170] and the *Spearman's* ρ_S [184].

The Pearson's correlation coefficient (also called linear or product-moment correlation) is one of the most used measure of correlation in statistical analysis. It gives the quality of a least-squares fitting (see Appendix A.4) to the original data. Specifically, it is defined as the covariance of the two variables X and Y divided by the product of their standard deviations:

$$\rho_P(X,Y) = \frac{cov(X,Y)}{\sigma(X)\sigma(Y)}$$

This ratio can range between -1 and $+1$. Values near to such borders indicate that the two variables are strongly correlated, negatively or positively. On the contrary, values near to 0 suggest that X and Y are not correlated.

The Spearman's rank correlation is a nonparametric measure of the monotonicity of the relationship between two variables. Unlike the Pearson's correlation, the Spearman's rank correlation does not assume that both variables are normally distributed. Thus, the raw variables X and Y, containing n values, are first converted to rank variables x and y [159], then the Spearman's rank correlation coefficient is computed applying the following formula:

$$\rho_S(x,y) = 1 - \frac{6\sum_{i=1}^{n}(x_i - y_i)^2}{n(n^2-1)}$$

Similarly to Pearson, the Spearman's rank correlation coefficient varies between -1 and $+1$ with 0 implying no correlation. Correlations of -1 or $+1$ imply an exact monotonic relationship. Positive correlations imply that as X increases, so does Y. Negative correlations imply that as X increases, Y decreases.

The Pearson's and the Spearman's correlation coefficients were widely used in the analysis of our data. Pearson was quite useful to understand how much two vectors of data are similar (see Chapter 4), whereas Spearman allowed us to analyze the level of correlation between rank vectors (see Chapter 3).

The last measure we discuss is the *Recurrence Plot* [88, 150]. Actually, the

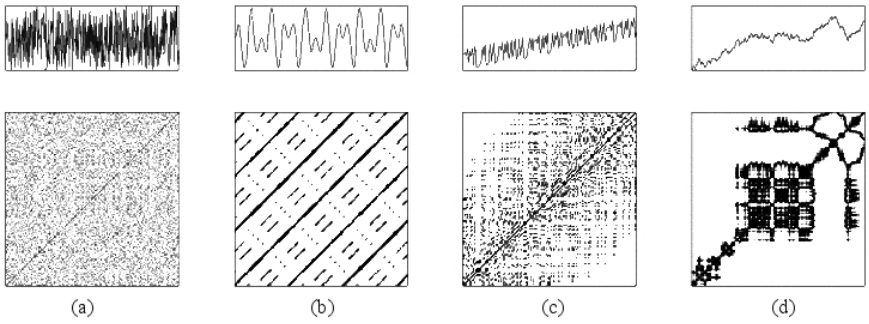


Figure 1.7: Examples of RPs. In the top row there are the time series plotted over time, while in the bottom row the corresponding recurrence plots are shown. (a) white noise, (b) harmonic oscillation with two frequencies, (c) chaotic data with linear trend, (d) data from an auto-regressive process.

Recurrence Plot (RP) is not a one-dimensional metric, rather it is an advanced technique of nonlinear data analysis able to visualize a corresponding square matrix in which each element coincides with the time at which a state of a dynamical system recurs. In other words, a RP shows all the times T at which a phase space trajectory visits roughly the same area in the phase space:

$$\mathbf{x}(i) \approx \mathbf{x}(j)$$

where $i, j \in \{1, \ldots, T\}$ and the symbol $\approx$ means equality up to an error (or distance) ε. The presence of ε is due to the fact that systems usually do not recur exactly to a formerly visited state but just approximately, therefore the error ε states a sort of range in which the two trajectories can differentiate. More formally, a recurrence $r_{i,j}$ can be defined by the following binary function:

$$r_{i,j} = \begin{cases} 1, & \text{if } \|\mathbf{x}(i) - \mathbf{x}(j)\| \leqslant \varepsilon; \\ 0, & \text{otherwise.} \end{cases}$$

where $r_{i,j}$ is the element of the full matrix $\mathbf{R}$ comparing the two states at time i and j. Thus, if the states i and j are similar, this is indicated by a 1 in the matrix $\mathbf{R}$; vice versa, if they are rather different the corresponding entry in the matrix is 0. To visually analyze $\mathbf{R}$, a RP is built by putting a black dot at all those coordinates (i, j) for which $r_{i,j} = 1$. Figure 1.7 shows some examples of RPs corresponding to typical signals.

Of course, in the case of evolving networks, the situation is a little different since we do not deal with two-dimensional signals, but we have n-dimensional vectors. Thus, the distance of two trajectories cannot be computed by just applying a subtraction between corresponding two vectors, rather we had to

rethink about a fair measure of distance to implement. More details are given in Chapter 3 where we made use of recurrence plots.

1.3 Random graph generative models

In this section we describe some models of static and growing random networks we have studied and analyzed during our research activity. Such models have become the fundamentals of random graphs due to the variety of properties that the corresponding randomly generated networks exhibit. The term *random graph* generally refers to probability distributions over graphs [64]. The main purpose of such random networks is that of serving as comparison against observed networks. Such a comparison allows to identify which properties of a social network are not the result of the pure randomness, but could be due to other factors, and so further analyzed.

1.3.1 Erdős-Rényi model

One of the first studies about random graphs for static networks was made by Erdős and Rényi in 1959 [94, 93]. The basic idea behind the Erdős-Rényi model is: a) consider a set of n nodes, b) each of the possible $\frac{n(n-1)}{2}$ links is formed with a given probability $0 \leqslant p \leqslant 1$, where the formation is independent. Thus, after iterating on all $\frac{n(n-1)}{2}$ links, the resulting graph will be a network having n nodes and $m \leqslant \frac{n(n-1)}{2}$ links, depending on the probability p. Of course, for $p = 1$ the resulting graph is a complete network, while for $p = 0$ the graph has no links.

Actually, the $G(n, p)$ model was first introduced by Edgar Gilbert [112] in 1959. In the same year, Erdős-Rényi introduced the $G(n, m)$ model, depending on a fix number of links m, where each link has the same probability to be created. However, both the models are often credited to Erdős and Rényi. The main difference between the two models is that the $G(n, m)$ model always outputs graphs all having a fixed number of links, while $G(n, p)$ outputs graphs with random links. Of course, for n sufficiently large, the probability p approximates the density $\lambda(G)$. Although the two models are clearly different, they have many properties in common.

Given a set containing all the possible networks that can be built on n nodes, the probability to have a network with exactly m links on those n nodes is:

$$p^m(1-p)^{\frac{n(n-1)}{2}-m}$$

For instance, if $n = 3$, then the probability to have a complete network with $m = 3$ is p^3. Vice versa, the probability to have any given network with

$m = 2$ is $p^2(1-p)$, while for any given network with one link the probability is $p(1-p)^2$. Finally, the probability to have an empty network with no links is $(1-p)^3$.

A fundamental statistic describing some of the properties of a random network is the degree distribution, which gives information about the probability of any given node i having a degree of k. So, the expression that provides the degree distribution is defined as:

$$P(d(i) = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

For large n and small p, it can be shown[3] that the above binomial expression can be approximated by a Poisson distribution, such as:

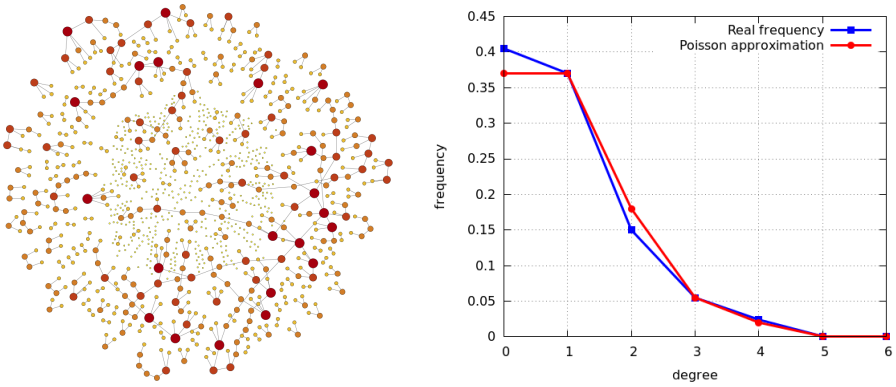
$$P(d(i) = k) \approx \frac{e^{-np}(np)^k}{k!}$$

Thus, random graphs, for which each link is formed independently with the same probability, are often called *Poisson random networks*. An example of a Poisson distribution, related to the random graph of Figure 1.8a, is shown in Figure 1.8b. It can be noticed that the real degree distribution is pretty near to the Poisson distribution. The graph is built using the $G(n,p)$ model, with $n = 1000$ and $p = 0.001$. Darker and bigger circles depict nodes with higher degrees. The resulting number of created edges is $m = 461$. This is a special case of a random network, where the expected degree for each node is 1. Indeed, considering a population of 1000 nodes, the probability $p = 0.001$ coincides, more or less, to the creation of 500 edges, namely an edge for each pair of nodes.

1.3.2 Wattz-Strogatz model

Although random graphs generated by the Erdős-Rényi model can show several properties related to social networks (e.g., a small diameter as the average degree grows sufficiently quickly), it appears clear that this kind of randomness lacks certain features quite common in social networks, such as a high clustering coefficient. To better figure out this issue, suppose that node i is linked to node j and j is linked to node k. What is the frequency with which nodes i and k will be linked in a Poisson random network? As we already said, the link formation is completely independent, so the frequency just corresponds to the probability p. Now, if n tends to infinity and the average degree grows

[3]Note that for large n and small p, $(1-p)^{n-1-k}$ can be approximated to $(1-p)^n$. If we now write $(1-p)^n = (1-\frac{np}{n})^n$, then it is approximately e^{-np} with $np = cost$. Similarly, for fixed k, large n and small p, $\binom{n-1}{k}$ is roughly $\frac{n^k}{k!}$.



(a) Poisson random graph (b) Degree distribution

Figure 1.8: A Poisson random graph (Fig. 1.8a) built using the $G(n,p)$ model with $n = 1000$ and $p = 0.001$, and the corresponding degree distribution with the Poisson approximation (Fig. 1.8b).

more slowly than the size of the network, p (and so the frequency of linking i and k) will tend to 0.

To overcome this effect, Watts and Strogatz presented an alternative model [199] to generate a random network. They show that only a small number of randomly placed links are required to obtain a small diameter in a random network. The algorithm starts constructing a regular ring lattice, namely a graph with n nodes, each of them connected to k neighbors, $k/2$ on each side. The parameter k is the average degree of each node and it is computed according to $n \gg k \gg ln(n) \gg 1$. More formally, assume to have nodes labeled $1, 2, \ldots, n$, then there is an edge (i,j) if and only if $0 < |i-j| \bmod(n - \frac{k}{2}) \leqslant \frac{k}{2}$. After building the regular ring lattice, the algorithm takes every edge (i,j) with $i < j$ and rewires it with probability β. Rewiring is done by replacing (i,j) with (i,k), where k is chosen uniformly at random from nodes that are not already neighbors of i. Of course, the more β increases, the more the network will be random. A random graph with $\beta = 1$ approaches the Erdős-Rényi random graph.

Thus, the underlying lattice structure of the model produces a locally clustered network and the random links dramatically reduce the average path length. This kind of networks are also called *small-world networks*. An example is shown in Figure 1.9. The figure shows that as β increases, does the randomness of the graph.

The main limitation of the model is that it produces an unrealistic degree distribution. A more realistic one is produced by *scale-free networks* introduced by the Barabási–Albert model (see Section 1.3.4). However, scale-free

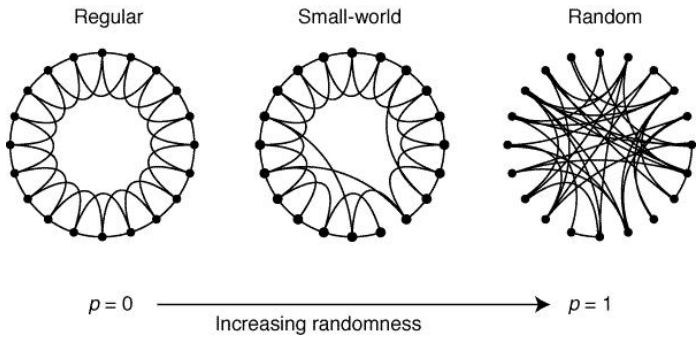


Figure 1.9: Examples of generated random networks using the Wattz-Strogatz model.

networks do not have high levels of clustering as in small-world networks.

1.3.3 Kleinberg model

Based on the work of Watts and Strogatz [199], Kleinberg proposed a class of small-world networks [133] to study decentralized algorithms by which humans, knowing only the locations of their direct acquaintances, attempt to transmit a message from a source to a target along a short path. The idea of transmitting a message from a user to another was firstly developed by Milgram in 1967, who proved and coined the famous property of "six degrees of separation" after deploying a social experiment in the United States [154].

Instead of using a regular ring lattice as in Wattz and Strogatz, the basic model proposed by Kleinberg uses a $n \times n$ grid network and allows for edges to be directed. A node i_{ab} in the network is identified by the coordinates a and b, such that $a \in \{1, \ldots, n\}$ and $b \in \{1, \ldots, n\}$. He also defines the *lattice distance* between two nodes i_{ab} and j_{xy} as the number of "lattice steps" separating them: $\Delta(i_{ab}, j_{xy}) = |x - a| + |y - b|$. The structure of the network is then ruled by two global constants: p and q. For $p \geqslant 1$, a node i has a directed edge to every other node within the lattice distance p, so forming its *local contacts*, and for $q \geqslant 0$ the model creates directed edges from i to q other nodes, which are called *long-range contacts*. The building policy of long-range contacts states that the m-th directed edge connecting i to j is formed with probability proportional to $[\Delta(i, j)]^{-r}$, with $r \geqslant 0$.

Setting p and q as fixed constants, the Kleinberg model generates a one-parameter family of networks, just tuning the value of r. Notice that for $r = 0$ the probability to connect i to any other node j is equal to 1, namely long-range contacts are chosen independently of their position on the grid. This means that the model builds long-range contacts according to the uniform distribution, the same used in the Wattz-Strogatz model. As r increases,

the long-range contacts of a node become more and more clustered in its neighborhood on the network.

The proposed decentralized algorithm is based on the Milgram's experiment. It starts with two randomly chosen nodes s and t (i.e., "source" and "target") and the goal is to transmit a message from s to t in as few steps as possible. The message is passed sequentially from a node to another of its local or long-range contacts using only local information. In this setting, Kleinberg shows that a simple decentralized greedy algorithm finds routes between any source and destination, coinciding with the expected delivery time, using only $O(log^2 n)$ expected links.

1.3.4 Barabási-Albert model

Beyond the Poisson random networks and small-world networks, another important class of networks is what is referred to as *scale-free networks*. A scale-free network is a network whose degree distribution asymptotically follows a power law. Some of the first work about such distributions was done by Pareto [167], who observed that wealth distributions had scale-free properties. The essential finding is that a rich person gets richer in a faster way than a poor person. Such features were also observed by Zipf [208] in the usage of frequent words.

Thus, a scale-free distribution $P(d)$ satisfies

$$P(d) \approx \alpha d^{-\gamma}$$

where $\alpha > 0$ is a scalar and γ is a parameter whose value is typically in the range $2 < \gamma < 4$. As the degree d increases, the frequency is reduced to $d^{-\gamma}$. However, scale-free networks have "fat tails" with respect to Poisson random networks. This means that there are many more nodes with small and large degrees than one would see in a Poisson distribution.

In the recent literature, Barabási and Albert [52] proposed a model, named *Preferential Attachment*, to explain such a link formation process and to generate random scale-free networks. The algorithm follows "the rich get richer" rule, namely it connects new Web pages to existing Web pages with a probability distribution proportional to the current in-degree of pages. Since the algorithm iteratively works over time, it can also model growing evolving networks, which are temporal networks that never delete nodes or edges. Of course, if the time is T-finite, then the graph at the time T will be the aggregate static network, result of all the previous steps.

In more detail, the algorithm starts with a network having a set of n_0 connected nodes. New nodes are added to the network one at a time and are connected to $n \leqslant n_0$ existing nodes with a probability that is proportional to

the degree of the existing nodes at that time. More formally, the probability $p^{(t)}(j,i)$ that a new node j is connected to the existing node i at time t is:

$$p^{(t)}(j,i) = \frac{d^{(t)}(i)}{\sum_{\tau=1}^{t} d^{(t)}(\tau)}$$

where $d^{(t)}(i)$ is the i's degree at time t, while $\sum_{\tau=1}^{t} d^{(t)}(\tau)$ is the sum of all the degrees of those nodes existing at time t. Therefore, nodes having higher degree, usually called "hubs", tend to accumulate more and more links, while nodes with only a few links are unlikely to be chosen as the destination for a new link. Hence the term preferential attachment.

1.4 Concluding remarks

In this chapter we have discussed about evolving social networks, first introducing social networks in the static case, then describing networks in dynamic contexts. We listed some general common properties among static networks and we extensively described the most popular representation and measures used in Social Network Analysis. One of the most important measure is doubtless PageRank, a centrality algorithm that has achieved a great success over the last decade. For this reason, we reserved a separate chapter in which we will discuss further issues. The same descriptive process was then used to illustrate dynamic evolving networks. We described GEXF, one of the most popular network file-format to represent evolving networks, then we introduced DNF, a new file-format able to represent static and evolving networks in a more compact way in terms of size. In the second part of the section we discussed about the corresponding measures. Finally, we introduced several models to generate random graphs. Such models are mainly useful to create synthetic datasets.

In the next chapter we carry on the discussion about social networks, mainly focusing on those networks formed by physical face-to-face interactions between humans in real-world social scenarios.

Chapter 2

F2F Social Networks

The participation in online social networking is continuously growing and more and more people use websites such as Twitter, Facebook or Google+. This new online phenomenon has led to a greater attention of the research community to social networks, also motivated by relevant application scenarios, such as reputation management [207], recommendation systems [44, 196] or information sharing platforms such as Quora. However, a new trend involving our "real life" relationships is progressively becoming contemporary due to the continuous development of new devices capable to track physical interactions. In this chapter, we discuss about Face-To-Face (F2F) social networks, namely networks made of evolving graphs in which nodes are humans and edges between nodes dynamically appear whenever a F2F interaction between humans takes place. Recently, some papers, such as [120], have focused on tracking physical proximities, however the size of those experiments is relatively small and some of them were deployed employing unsuitable technologies. So far, the main reason behind the lack of analysis of real-world interactions is mainly due to hardware limitations (e.g., bluetooth is too inaccurate to obtain an efficient F2F measure between people).

One of the first attempt to effectively track Face-To-Face interactions has been made by the SocioPatterns researchers [26, 54] using the Radio Frequency IDentification (RFID) technology. We believe SocioPatterns tags are among the most promising devices to track F2F interactions in real-world scenarios. For this reason, our experiments have been conducted using such devices with the main purpose of tuning the multiple access control (MAC) [202] protocol programmed into the tags to better suit heterogeneous application scenarios. Notice that, as we will better see in the next section, standard MAC protocols for Wireless Sensor Networks (WSNs) are not suitable in real-world social scenarios where the dynamics of interactions are extremely fast and unpredictable when compared to the WSNs ones.

This chapter is structured as following: in the next section we present protocols and applications in different kinds of scenario, such as Wireless Sensor Networks and Opportunistic Networks. Then, we introduce and discuss more formally F2F social networks presenting some of their applications and the technologies employed to track interactions. In the second part of the chapter we discuss the issues in tracking F2F interactions in heterogeneous application scenarios presenting two suitable MAC protocols. Finally, we show the results of the experiments conducted on F2F social networks and on a simulation environment to test the performance of the proposed MAC protocols.

2.1 Background on MAC protocols

MAC protocols in sensors applications
In recent years several works have concerned MAC protocols, mainly in the field of Wireless Sensor Networks (WSN). This is because WSN have become a leading solution in many important applications such as intrusion detection or environment monitoring. Typically, a WSN consists of a large number of small sensor devices that are distributed in the target area for collecting data of interest. All nodes are potential transmitters as well as receivers, consequently a MAC protocol is necessary to control multiple access to the channel [202].

One of the first contributions in this area was given by [42], where Abramson presents *ALOHA*, a protocol developed at the University of Hawaii, providing a fundamental demonstration of a wireless packet data network. In ALOHA, a node sends its packet as soon as it is available to be transmitted. If no other node is transmitting, the packet reaches the destination and the receiver sends an acknowledgment. Vice versa, if a collision occurs, the destination node does not receive any packet, consequently the sender does not get any acknowledgment. In that case, the sender will transmit again its packet after a random amount of time. Clearly, the simplicity of this protocol affects its performance, mainly compromising the maximum throughput. Better performance are achieved by an improved version of this protocol known as *Slotted ALOHA*, which introduces discrete time slots. Each node can only transmits at the beginning of a time slot, thus collisions are reduced and the maximum throughput increases.

Later on, Kleinrock and Tobagi thought about solving the problem at the root, rather than deal with it when occurring. In [134] they present *Carrier Sense Multiple Access* (CSMA), a contention-based mechanism recently used by many other MAC protocols. A wireless device adopting CSMA techniques that wishes to transmit a message, first senses the channel to determine if another device has already started transmitting. If the device detects activity on the channel, it waits a certain amount of time before attempting to transmit

again. The waiting interval may vary depending on which kind of CSMA it implements: in non-persistent CSMA, the device performs a backoff operation, while in *p*-persistent CSMA it continuously sense the channel. As the device senses no activity on the channel, it transmits the message. A further mechanism that extends the functionalities of CSMA is *Collision Avoidance* (CA). It limits the number of lost messages when nearby devices transmit at the same time thanks to the implementation of the *RTS-CTS* mechanism.

Other techniques relating to MAC are based on reservation over time [121]. The most representative protocol for such an approach is *TDMA*, where each single sensor node is assigned a time slot to transmit. This mechanism clearly reduces collisions, but requires the knowledge of the whole network topology to establish a schedule of communication.

Following studies based on contention or reservation methods have mainly related to energy efficiency techniques due to limited power supply of sensor nodes. In [205] Ye et al. presented *S-MAC*, one of the first low-power RTS-CTS protocols for WSN where nodes periodically sleep, wake up, listen to the channel and then return to sleep. Each active period is of fixed size of 115 ms, while sleep periods are variable determining the duty cycle of the protocol. At the beginning of each active period nodes exchange *SYNC* packets in order to be synchronized with others. S-MAC is considered to be not much scalable because, as the size of the network increases, it has to manage an increasing number of neighbors' schedules or to face an additional overhead due to continuous resynchronization.

T-MAC [191] is proposed to enhance the performance of S-MAC under conditions of variable traffic load. After the SYNC segment, there is a short window to send or receive RTS-CTS packets. If no activity occurs in that period, the node returns to sleep. Although T-MAC outperforms S-MAC in terms of energy efficiency whenever conditions of variable load occur, they perform equally well in homogeneous workloads. Nevertheless, T-MAC suffers from the same scalability problems of S-MAC.

Polastre et al. [174] proposed one of the first successful asynchronous low-power implementations knows as *Berkley MAC* (BMAC). Such a protocol uses *Low Power Listening* (LPL), a preamble-sampling strategy that enables radio to operate at low duty cycles. Every node implementing LPL periodically wakes up to sense activity in the wireless channel. Therefore, if the node detects any activity during channel sampling, then it stays awake to receive packets, otherwise it returns to sleep. Since BMAC uses only physical-layer information from the radio, all nodes in proximity of a sender wake up every time they detect activity, even if they are not the targeted recipient of the packet. After waking up, all receivers have to remain active in receiving mode consuming energy before acquiring the packet. This brings to a well-known overhearing problem and *X-MAC* [71], defined as short preamble MAC pro-

tocol for duty-cycled WSN, attempts to solve this issue by adding link-layer information into its mechanism. However, Moss and Levis [157] highlight that, because of the X-MAC policy, the packets within wake-up transmission consist only of 802.15.4 headers, so nodes are required to perform an additional handshake transmission to exchange the final payload within a separate packet. Consequently, this reduces its maximum possible throughput by almost half. Thus, in [157] they present *BoX-MAC-1* and *BoX-MAC-2* as alternative protocols to BMAC and X-MAC, in order to solve the aforementioned problems. BoX-MAC-1 improves upon BMAC by replacing its bit-stream preamble with a continuously packetized wake-up transmission at link-level, while BoX-MAC-2 outperforms X-MAC by replacing the X-MAC's packet-based receive check with a physical-layer energy-based receive check. As a result, no handshaking is required in BoX-MAC-2, so throughput can nearly double X-MAC's.

Other hybrid synchronous protocols, such as *WiseMAC* [90] or *Z-MAC* [177], attempt to overcome the TDMA's deficiency by combining TDMA with several types of asynchronous techniques. For instance, WiseMAC combines asynchronous channel sampling with scheduled transmissions, while Z-MAC works as a contention-based protocol for low-traffic levels, but it turns into TDMA mode for high levels. According to [139] WiseMAC is considered the most performing MAC protocol for low data rate applications, but, as in the case of BoX-MAC-1, its performances quickly degrades whenever broadcast communication pattern is required.

All MAC protocols we revised are usually employed in sensor applications where network topology remains quite the same over time and the amount of exchanged packets between nodes is relatively low. There is no mobility, so the communication pattern is well-known and does not change. The only phenomenon that can generate any change in such networks is node failure. However, even though some failures occur, the network topology still remains quite similar. Vice versa, the kinds of networks that we need to deal with are highly dynamic, where nodes continuously move and the corresponding topology changes every time. Although clustering may occur among people, the communication pattern is not fixed at all, nor predictable.

Opportunistic Networks and WBANs

In real-world scenarios, the kind of networks being more similar to F2F social networks is the opportunistic ones [80], an emerging paradigm of human-associated networks in which mobile users interact with each other based on their geographical proximity. Such networks, communications between humans are peer-to-peer using short-range technologies. However, their purpose is totally different with respect to F2F social networks since devices carried by humans aim to deliver messages on behalf of others in an intermittent com-

munication. In effect, opportunistic networks can be considered as a sub-class of Delay-Tolerant Networks [95], a particular architecture allowing communications among disconnected networks. Furthermore, the attention of the research community is mainly focused on how to define mobility models starting from real traces, such as ones discussed in [137]. Conversely, we want to shift the focus onto definition of ad-hoc MAC protocols suitable for social networks in real-world scenarios. Musolesi and Mascolo [158] classify human mobility models into synthetic mobility traces and real-world traces. They also introduce for the first time the concept of social networks into mobility models. Similarly, Aschenbruck et al. [46] provide a survey of available movement traces as well as synthetic mobility models for multi-hop wireless networks.

Another class of networks directly adopted on and by humans is Wireless Body Area Networks (WBANs) [143]. However, all networks produces by evolving real-world scenarios should be considered completely different with respect to WBANs, since the former refers to high-dynamic evolving networks belonging to real contexts, while the latter are usually employed in energy-saving and low-traffic communications. WBANs use protocols able to efficiently route traffic through several devices in order to save energy and to ensure an adequate level of reliability. They are often structured (e.g. in tree or star networks) so that designers can implement specific methodologies and functions able to exploit predicted paths for exchanging data as much as possible. This kind of networks is often used in medical applications [190] where people wear sensors capable to monitor their health condition and to forward data to a central collector. Vice versa, in our case, a network is totally unstructured and unpredictable since individuals interacting with each other can continuously change their relationships.

2.2 F2F social networks

A Face-To-Face (F2F) social network is a dynamic evolving network made by linking nodes (i.e., humans) that interact at short range (e.g., 1-1.5 meter of distance) for a sufficient amount of time to potentially exchange meaningful information. We recall from Chapter 1 that an evolving network $\mathcal{G}$ is made of a finite sequence $G_0, G_1, G_2, \ldots, G_t$ of t static networks over the same vertex set V and a variable edge set E_i, with $i \in \{1, \ldots, t\}$. A link connecting nodes $(u, v) \in E_i$ if and only if at least a F2F interaction between u and v occurred in the interval in time δ between G_i and $G_{i+\delta}$. We call δ the *resolution* of the network.

2.2.1 Technologies

At present, there is a limited number of solutions able to track interactions between individuals in a distributed way that include sensors and wireless technologies, such as Bluetooth, WiFi and RFID. One of the first experiments to collect information from a real group of people was made in [120], where 54 individuals participating in a conference were given Intel Imote devices, equipped with a Bluetooth radio and a flash memory. The Imotes were configured to perform a Bluetooth base-band layer "inquiry", discovering the MAC addresses of other Bluetooth nodes in range, and the results of inquiry were written to the flash memory. However, Bluetooth does not allow a fine-grained recording of social interactions because of two reasons: 1) the discovery process to identify potential F2F neighbors is slow and 2) since the radio range is $\approx$ 5-10 meters, it is possible to record as "social interaction" the simple fact of being in the same room, even if the users are not interacting in any way.

In [78] Choudhury et al. present initial results regarding physical interactions of several groups of people. The first group was composed by 8 subjects from the same research group, while the second set of the experiments included 23 individuals from four different research groups. Proximity was measured by sociometers, namely wearable sensors able to track people's interactions via the IR technology and speech recognition. Authors addressed a crucial problem of privacy related to speech recognition, highlighting how most people were wary about the final use of this information. Thus, to protect the user's privacy they only extracted speech features, such energy and spectral features, and never processed the content of the speech.

Later on, first Choudhury et al. [77] officially introduced their mobile sensing platform, then Jayagopi et al. [126] deployed some experiments involving 24 groups of 4 members each. Every participant was asked to wear a similar sociometric badges capable of recognizing speech activity and line-of-sight presence. Authors were interested in discriminating one conversational context against another, specifically brainstorming from decision-making interactions using easily computable nonverbal behavioral cues. However, all the experiments performed in [126] are based on relatively small groups in which interactions are limited inside the group, while we are interested in a technology to study the dynamics of larger groups in which members are free to move. Moreover, sensors based on speech recognition may be not efficient if there is too much noise.

SocioPatterns is an interdisciplinary research collaboration that supports the development of the *SocioPatterns Sensing Platform*, an infrastructure including new experimental RFID sensors that can be worn by humans in order to track their mobility and F2F interactions in real-world scenarios. The SocioPatterns platform is made of two main entities: active RFID tags (see

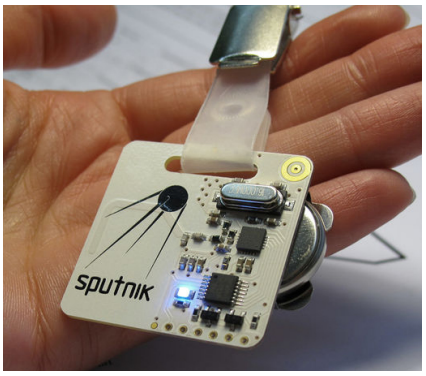


Figure 2.1: RFID Tag

Figure 2.2: RFID Reader

Figure 2.1) and RFID readers (see Figure 2.2). Specifically, OpenBeacon tags [22] consist of a PIC16F688 [35], which is a PIC16 micro-controller (MCU), and a nRF24L01 [21], namely a ultra low power 2 Mbps RF transceiver working on the 2.4 GHz ISM band. Similarly, the RFID reader has a nRF24L01 transceiver as well, but its schematics are not publicly available. The tags' MCU has a total SRAM of 256 byte and can work up to 8 MHz of frequency, while the transceiver has very low energy consumptions: 11.3 mAh in transmission at 0 dBm of output power and 12.3 mAh in reception at 2 Mbps of air data rate.

When two persons are in proximity within 1-1.5 meter (see Figures 2.3), their tags exchange proximity packets containing their IDs (process 1 of Figure 2.4) so that each tag is able to know who is talking to. Such tags have proved to be pretty accurate in measuring F2F proximity. For instance, if two people wear tags attached to lanyards and they are turn over, although in a close range, the communication of the packets sent by their tags is obstructed by their body mass. Eventually, tags send the received proximity packets to close-by readers (process 2 of Figure 2.4), which in turn will forward those messages to a central server running the OpenBeacon logger [98] (process 3 of Figure 2.4). Proximity ranges can be controlled via firmware by setting the transceiver's transmission at 4 different levels of powers: 0, -6, -12, -18 dBm. Low-power transmissions entail lower ranges of proximity. Usually, the two or three lower power levels are used to send proximity packets to sense spatial neighborhood, while the highest power is used to report proximity packets to close readers. We measured that tags can reach the readers up to 15 meters *line-of-sight* transmitting at the highest level. This empirical measure is fundamental to estimate the number of readers required in deploying a social experiment in a certain environment.

The SocioPatterns platform has proved to work well in a number of F2F

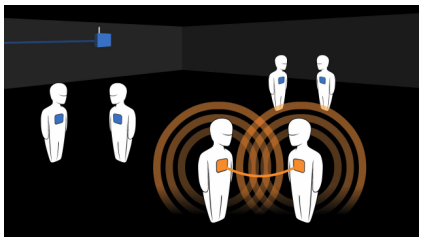


Figure 2.3: F2F interactions.

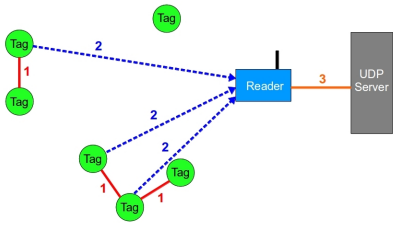


Figure 2.4: The communication process.

application scenarios that are discussed in the next section.

2.2.2 Applications

During our daily life we usually come into contact with other people, exchanging some words, talking to strangers and frequenting new places. So far, all these actions could not be measured or tracked with a high level of accuracy, but recently, thanks to new technologies such as RFID, F2F social networks have become a reality, allowing to analyze what happens in real-world scenarios. Therefore, capturing proximity between individuals allows to collect several type of data and study different behaviors of a community.

SocioPatterns performed several installations in different social contexts analyzing the obtained dynamics, such as the study of relationships between attendees in conferences [55, 85] or the spread of infectious disease in hospitals and schools [122, 195, 193, 185]. One of the first works concerns Live Social Semantics (LSS) [55], which is an applications that relates virtual and real interactions among individuals during conferences. Furthermore, since sensing human behaviors in real-world scenarios opens new frontiers in ubiquitous areas, SocioPatterns have started studying and analyzing characteristics in this kind of social networks [53, 111]. Analogous experiments to the SocioPatterns' ones were deployed by Chin et al. [76] giving each person an active RFID badge during the course of a conference. They were interested in realizing a system able to find and connect people to each other. Analyzing such an experiment, they discovered that more proximity interactions result in an increased probability for a person to add another as a social connection.

All these applications have in common the study of observing what happens in social networks formed in real contexts. Nevertheless, an application of such type may be totally different with respect to another one if we take into account the purpose of the study and the kind of interaction they are interested in collecting. So far, all deployed experiments have aimed to analyze the resulting social network formed by the occurred interactions, but

without the objective of understanding the nature of the single interactions or diversifying the relationships. If we consider a game of 20 minutes and an experiment deployed in a school lasting 1 week, is the interaction type the same or not? Probably, during the game interactions tend to be fast and short, while in the experiment of the school relationships are more durable and recurring. Managing this diversification requires various settings for the protocol of communication. In testbeds lasting long time we need a protocol able to save energy as much as possible, while in testbeds of a few minutes the protocol is allowed to consume more energy, but it must capture as many interactions as it can. For that reason, we analyzed and evaluated different settings of the communication protocol in order to observe its performance during *short-lasting* and *long-lasting* experiments.

2.3 Capture F2F interactions in real-world social scenarios

As we have seen in Section 2.1, nowadays a number of protocols exist for wireless sensor networks applications [84], opportunistic networks contexts [80] or mobile ad hoc networks [188]. Nevertheless, most of those protocols are not well-suited to track fast-changing F2F networks where interactions are very dynamic and potentially might change the structure of the network at every time-step. We stress that the main purpose of our investigation is to accurately track F2F interactions, rather than to allow users to exchange long messages. In this perspective, the proposed MAC protocols are primarily designed to quickly exchange small packets to build a neighborhood map rather than to exchange relatively big packets.

SocioPatterns tags were initially provided with a MAC protocol, dubbed `SOCIOMAC` [41] in the following, usually used in long-lasting applications and slow-changing interactions. `SOCIOMAC` allows to identify nodes in the proximity and to deliver the position and the list of encountered nodes to the readers. Communication is encrypted using the `XXTEA` encryption [201, 204] to support user privacy, and packets are broadcast to the readers in a *best-effort* fashion with no acknowledgment mechanism to minimize the overhead of the communication. Indeed, any additional reliability mechanism may be redundant and counter-productive in such networks. The three types of packets used by the protocol are: *contact* (or *proximity*), *beacon* (or *sighting*), *report*. Contact packets are used to identify close-by tags. This information, as well as the position of the node, is then delivered to the reader using the report and beacon packets, respectively.

Figure 2.5 illustrates an entire phase of the default `SOCIOMAC` protocol. Each phase, composed of 8 cycles, represents the main policy of the protocol.

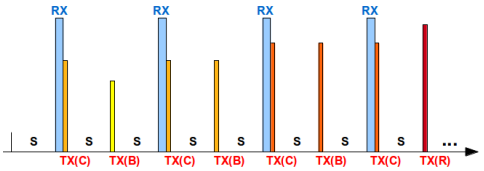


Figure 2.5: The SOCIOMAC protocol

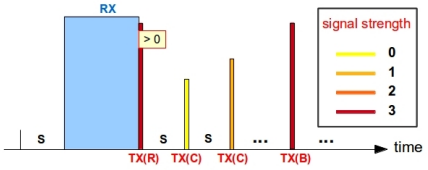


Figure 2.6: The PROXMAC protocol

Every cycle starts with a random sleeping period S ranging between 50 and 91 ms, to avoid periodic collisions, then it is concluded alternatively by the transmission of contact packets TX(C) or by the transmission of beacon packets TX(B). Before performing a TX(C), tags go in receiving mode RX for 5 ms so as to receive contact packets from possible neighbors. A phase is concluded with the 8th cycle, when a report packet TX(R) is sent to the reader. It can be computed that the average duty cycle of SOCIOMAC is less than 10%, making this protocol suitable for long-lasting deployments. However the probability P_{succ} of exchanging a contact packet between two neighbors in a phase "only" ranges between 10% and 17%.

According to the analysis and evaluation given in [72], a resolution of $\delta = 20$ seconds is considered sufficient to accurately track F2F interactions and to consequently assess proximity. More specifically, a contact between two individuals is established with a probability in excess of 99%, namely there is a link in G_i, as long as at least one of them receives a contact packet from the other every 20 seconds. However, such a time scale could be not sufficient in some cases. In particular, as discussed in Chapter 4, we consider a scenario to investigate the so called "Wisdom of Crowds" [187], where a group of people is asked to answer a set of questions in two separate phases.

In a first phase participants have to guess the answer without having any kind of relationship with others, while in the second phase they are allowed to interact and exchange their opinions. Since users are already aware of the questions asked in the first phase, in the second round fast interactions, such as "What did you answer the first question?", are likely to happen. In this case, a higher resolution (i.e., a lower time scale, such as 5 seconds) may be required in order to capture all those kinds of interactions, consequently it is crucial a fine-tuning of the SOCIOMAC parameters to increase P_{succ}.

Since in short-lasting deployments energy constraints are less demanding and interactions tend to be fast and dense, at the cost of a higher duty cycle, the first tuning that had to be made was the increase of the receiving interval so as to collect with a higher probability incoming packets from fast interactions. We recall that the SOCIOMAC protocol sets the RX interval to 5 ms; because the transmission of a packet approximately lasts the same order of magnitude,

the protocol is not always able to capture packets in their first exchanges. For that reason, we decided of redesigning the main policy of the protocol, named `PROXMAC` and shown in Figure 2.6, so that a fine-tuning of the parameters would have been achievable. In the next section we will show how P_{succ} increases, making this protocol more suitable for detecting fast dynamics. Values of the sleeping period and the receiving interval can be configured according to the scenario which is going to be deployed.

Notice that, in the `PROXMAC`'s new policy, the report packet can be sent after every receiving operation only if the tag has previously received some contact packets, otherwise this transmission is skipped. This features guarantees communications broadcasting only useful packets and reducing contingent collisions towards the readers. Furthermore, to better discriminate F2F links, `TX(C)` are transmitted with signal strengths 0 and 1, namely -18 and -12 dBm, while the default `SOCIOMAC` protocol uses values of 1 and 2, making the communication more prone to false positives. Finally, a tag running `PROXMAC` sends a *beacon* packet only every t phases, so as to avoid to incessantly flood the channel. The t parameter is usually configured to receive at least one beacon packet at each time-step. That ensures a continuous monitoring of the tag inside the network. Both the beacon and report packets are sent using a signal strength of 3 in order to ensure with a good probability the collection of the packet by close readers.

Of course, depending on the chosen parameters, we can experience with different duty cycles and probabilities P_{succ} of successfully receiving a packet. As sample of setting, also used in the experiments discussed in the next section, we pick out a sleeping period ranging in $[20, 30)$ ms and a receiving interval in a range between 30 and 39 ms. Such values, as well as other settings, were tested in our labs and, after a long time of measurements, we found that such a configuration was a good compromise in terms of performance for our purposes and social experiments in which a high number of iterations of the main phase in a time-step was required to capture fast interactions as much as possible. Other values of `S` and `RX` are clearly available and allowed; they only depend on the purpose of the deployment. However, we should take into account the fact that the minimum value of the receiving interval must always be equal or greater than the maximum value of the sleeping period, so that the `PROXMAC` policy is still guaranteed. If that condition is not fulfilled, then it may happen that a whole receiving interval completely falls in the middle of the sleeping period, thus compromising P_{succ}.

As we will better see in the next section, using those parameters, `PROXMAC` can often ensure the reception of at least one proximity packet every 2 or 3 time-steps when two tags face each other at 1-1.5 meter. Therefore, people having very fast interaction can be now tracked and logged. Unlike `SOCIOMAC`, where the recommended resolution is $\delta = 20$ seconds and the maximum signal

strength for collecting contacts is 2, PROXMAC can exploit a more fine-grained δ and better manage false positives thanks to a reduced signal strength. An improvement of the performance also depends on the new length of the phase being very short, so that the protocol can iterate the main cycle several times within a time-step.

2.4 Experiments and evaluation of the protocols

In this section we evaluate the SOCIOMAC and PROXMAC's performance under different conditions in real-world testbeds. First, a set of preliminary measurements on fixed small-scale networks are conducted to estimate the number of contact packets received by tags and their false positives. In a second step, we present results of all the six social experiments deployed during our research activity. In the end, we simulate the protocols performance in larger and denser graphs.

2.4.1 Preliminary experiments

The first experiment measures the number of contact packets exchanged in 10 minutes between two F2F tags 1 meter apart and then collected by a reader. This kind of setting is used to simulate an interaction as happens in real-world scenarios. Of course, in real-world testbeds, communications are more challenging with respect to this preliminary experiment because of overlapping transmissions and dense networks. However, this first experiment allows us to better understand how protocols perform in an ideal setting, so that we can know when to use determined settings in real-world social networks in which performance measures are too difficult to explore and analyze in detail. The plots depicted in Figure 2.7 show the results of the experiments. The x-axis represents the time of the testbed, while the y-axis the total number of distinct proximity packets collected by both the protocols. Notice that, the slope of both the plots is less than 45 degrees. This means that, for both the protocols, a resolution of 1 second (i.e., the minimum measurable time-step in real-world scenarios) cannot be achieved. However, as expected, PROXMAC provides better performance in terms of proximity collection. In this simple setting it is able to support an average resolution of $\delta = 1.8$ seconds, while SOCIOMAC supports a resolution of $\delta = 2.9$ seconds.

The results of the second experiment are shown in Figure 2.8. The experiment evaluates the number of false positives reported by both the protocols, namely all those contact packets exchanged by two tags *not* facing each other. Two pairs of nodes are placed at a distance of 2.5 meters. As in the previous experiment, the two tags in a couple are 1 meter apart. In principle, a tag

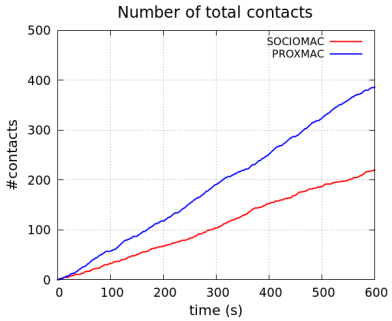


Figure 2.7: All the packets exchanged over time.

should detect only the other one in the couple (i.e., a F2F interaction). However, due to the proximity of the two couples, false positives may occur. Of course, similar behaviors occur in real-world scenarios whenever there are high network densities. For instance, it can happen that two independent groups of people at close range wrongly exchange contact packets.

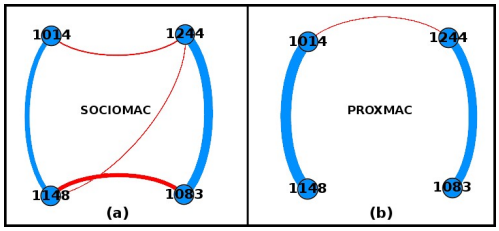


Figure 2.8: Correct contacts and false positives.

As shown in Figure 2.8(b), most of the `PROXMAC` contact packets are correctly exchanged (blue arcs) between the tags in the couple. On the contrary, that is not the same for `SOCIOMAC` (see Figure 2.8(a)), where a number of false positive occurs (red arcs). The thickness of the edges is proportional to the number of contact packets exchanged between the tags. A ticker edge indicates a longer interaction, while a thin arc shows a short-lasting interaction. This feature helps to visualize what kinds of contacts were common. In the `SOCIOMAC` case, the formed network is the result of 5 edges, including two regular interactions and three false positives. The two more thin false positives could be excluded using a filter that analyzes the structure of the network enforcing ti stay over a certain threshold, but the third arc has a thickness pretty similar to one of the two regular. Therefore, there is a significant problem in selecting which arc to keep and which arc to discard. The main cause of this issue in `SOCIOMAC` depends on the signal strength used by the protocol and the corresponding policy. The three levels of powers used by `SOCIOMAC` to

exchange proximity packets brings to a wider range of communication catching farther tags which cannot be considered face-to-face. This issue is rather reduced using PROXMAC. Indeed, the only false positive registered during the experiment was the thin edge between nodes 1014 and 1244. However, when the thickness of the false positive is quite thin with respect to other regular edges, then that arc can be easily rejected using some techniques of filtering in post-processing.

2.4.2 Real-world social experiments

During our research activity we have deployed six social experiments involving humans and using the SocioPatterns infrastructure. The first two experiments helped to start investigating this new technology and to study Population Protocols [45, 47] on real-world social networks [58], while the other four experiments were employed to analyze the wisdom of crowds phenomenon [187].

Before starting in describing each testbed in detail, we first introduce three different mechanisms of link establishment that we have considered to build connections between individuals in social graphs.

Continuous-steps mode. The first method we have analyzed simply establishes a link if two vertices u and v have a certain number of continuous contacts η over time. For instance, let us suppose a value of continuous contacts $\eta = 5$, then a link can be constituted if and only if at least one of the two vertices have logged an interaction of at least 5 continuous contacts. Since the finest-grained resolution is 1 time-step, then we can state that η continuous contacts are equal to an interaction of η continuous time-steps. The main drawback of this method is an unexpected interruption of the communication, mainly due to the physical limits of the technology used to track interactions. Wireless communications can generate a lot of collisions, so it is very likely that some packets could be lost, causing a setting-up break for the concerned link.

Interval mode. In this second technique a link exists between two vertices u and v if and only if they have exactly a certain number of contacts η over synchronous snapshots of τ time-steps. We call this quantity the frequency $f = \frac{\eta}{\tau}$. Unlike the previous method, the number of contacts are not required to be continuous. Therefore, the two parameters accepted by the method are the frequency $f \in [0, 1]$ and the window $\tau \in \{1, ..., T\}$. As simple example, suppose to have a frequency $f = 0.4$ and a snapshot $\tau = 10$ time-steps. A link is constituted if the two vertices collect at least 4 contacts in that snapshot. The main drawback of the time-interval method is the loss of contacts outside the snapshots. In other terms, if an interaction keeps on existing for other few

time-steps after a link was established and the corresponding contacts are not enough to be counted in the next window, then these last contacts will be lost.

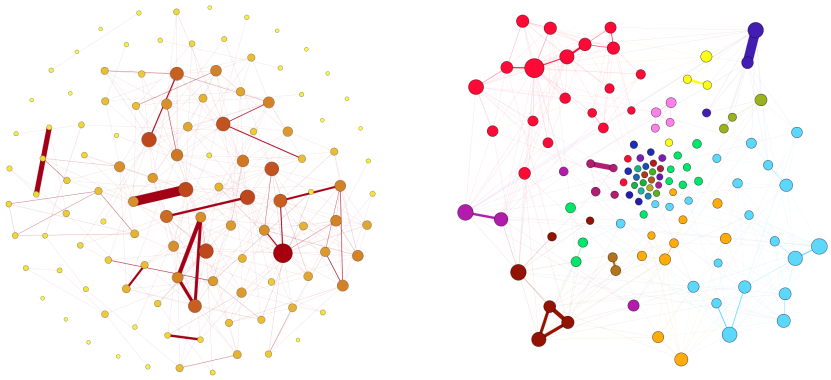
Incremental mode. The last method is likely the most fair. It is based on two parameters: the first one concerns the minimum number of contacts η (not necessarily to be continuous) to setup a link, while the latter is the expiration window Φ which kills the link establishing process if no contact corresponding to that link is collected in a given window. For instance, let us suppose to have a number of contacts $\eta = 5$ and an expiration window $\Phi = 3$. A link can be constituted if at least 5 contacts are collected and each of them has to occur before the window of 3 time-steps, starting every time a packet is collected, expires and kills the process. This method performs pretty well against technological constraints and never misses "orphan" contacts as happens in the interval mode. When $\eta = 1$, then Φ coincides with the resolution of the network δ.

2.4.2.1 Students at the department (DIAG0)

The DIAG0 experiment was deployed at the Department of Computer, Control, and Management Engineering "Antonio Ruberti" at Sapienza, University of Rome, monitoring several rooms and common spaces. It lasted 5 days, from October 17, 2011 to October 21, 2011, and involved 116 participants, including undergraduates, graduates and Ph.D. students. The main purpose of the experiment was to test the SocioPatterns infrastructure as we used it for the first time. Moreover, the social traces were then useful to simulate and analyze Population Protocols on a real-world social network [58].

First of all, we placed a reader, powered by a USB adapter, in each monitored room, corridors and relax areas. We used the network infrastructure already available across the whole building to connect all the readers in a dedicated local area network with static IP addresses. Then, we asked each of 116 students to wear an active OpenBeacon RFID tag which periodically sent information about the approximate location of the person and its social encounters. An instance of information we got is `tag 1274 met tag 1055 in room 3`, or `tag 1143 is close to room 5`. Furthermore, we stored additional information about each person participating to the experiment, such as age, gender, course of study and academic year. This allowed us to analyze how students cluster at the department, what kind of interactions they have and the common spaces they use. No other personal information was asked in respect of their privacy.

At the end of the experiment we collected around 200 MB of raw data log, which was then parsed using the OpenBeacon Parser [99], a software tool we have written to build networks from raw data, and made publicly available



(a) Fruchterman-Reingold layout (b) ForceAtlas2 layout

Figure 2.9: The social graph related to the DIAG0 experiment. Figure 2.9a shows the network drawn by using the Fruchterman-Reingold layout, while the network in Figure 2.9b is drawn by employing the ForceAtlas2 layout.

at [37] in several file-formats. Figure 2.9 shows the resulting aggregate social graph obtained from the interactions collected during the experiment and drawn using Gephi [8]. Since the protocol used to track F2F interactions was `SOCIOMAC`, we built the aggregate social network using the incremental mode with $\eta = 1$ and $\Phi = 20$ as previously suggested by [72]. The total number of single proximity packets collected by the server was almost 23000, among which we have 16300 contacts that lasted just 1 second, while around 3350 contacts lasted 2 consecutive seconds. Each single snapshot of the evolving network is so sparse that the average temporal density is $\Lambda = 1.205 \cdot 10^{-5}$.

More in detail, Figure 2.9a shows the social graph emphasizing the structure of the network in terms of degree of nodes and weight of edges. The size and the darkness of a node are directly proportional to its relevance (measured by the degree) in the network. Similarly, the thickness of an edge (i, j) is proportional to the strength of the interaction between the two nodes i and j. The layout adopted to draw the network was the one proposed by Fruchterman and Reingold [108]. The average degree of the network is 5.103, while its density[1] is $\lambda = 0.044$. The diameter is 7 and the average path length is 3.152. Figure 2.9b adds a further information regarding the modularity of the network (see Section 1.1.1.3), which is $Q = 0.836$. A total of 31 communities, each of them characterized by a different color, appear in the social graph, but half of them are due to isolated nodes. The community detection was

[1] Notice that the density λ computed on the aggregate graph is quite different from the average temporal density Λ computed on the evolving network. See Section 1.1.1.3 and 1.2.1.2 for further details.

obtained by using the method presented in [62], while the layout used to draw the graph was ForceAtlas2 [124].

A live demo showing the full evolving history is available at [38]. It draws, for each time-step, all the nodes and interactions they can have. Individuals are depicted by blue circles, while readers are represented by black circles. Each time two nodes have an interaction, they become red and a yellow edge connects them. The background image of the application is the map of the first floor of the department, where the experiment took place.

2.4.2.2 MACRO museum

We carried out a second social experiment at the MACRO museum in Rome during the NEON exhibition opening held on the 20th of June 2012. It lasted around 2.5 hours and involved 114 visitors. For the whole evening we tracked in a completely anonymous way, the movement and the interactions of people in the Enel Room. We asked each visitor to wear an active OpenBeacon RFID tag periodically transmitting packets to close-by readers, which were located in proximity of the artworks. This arrangement was thought to extract useful information for the museum, such as the artworks more popular or the spaces more common. As in the previous deployment at the department, we stored additional information about each visitor participating to the experiment, such as age, gender, education level and professional area. Specifically, we were placed at the enter of the museum where we asked each visitor to fill out an online form after receiving a RFID tag. This installation was more complex with respect to the previous one because no ethernet connection was available in the room where artworks were placed. We thus decided to set-up a mixed infrastructure using power-line adapters (D-Link AV 500) for cabled connection over the power line and wireless routers (TP-LINK TL-WR740N) to establish wireless bridges. Our main server was located at the "Area", a special open space of the MACRO museum where visitors can relax, meet other people and share ideas and thoughts. The person in charge of the "Area" was Miltos Manetas, an artist who collaborated with us for deploying the social experiment. In the "Area" we placed a screen where quasi-real time images (with a delay of about 15 seconds) of the movement of visitors inside the Enel Room were displayed. Such an application was an artwork as well, and its name was *300 Visitors* [36]. A further real-time application [40], showing the full evolving history of the interaction between visitors, was developed. The purpose of this second representation mainly was that of stimulating participation to our experiment. It shows the movement and the proximity interactions of visitors in an "artistic" fashion. We assigned each reader a different color and every node is colored by the same color of the reader that collected its movement. Then, every node drawn in the application is kept for a certain amount of

(a) Fruchterman-Reingold layout

(b) ForceAtlas2 layout

Figure 2.10: The social graph related to the MACRO experiment. Figure 2.10a shows the network drawn by using the Fruchterman-Reingold layout, while the network in Figure 2.10b is drawn by employing the ForceAtlas2 layout.

time in order to create a sort of "virtual impressionist painting". Thus, visitors coming to the entrance of the museum and watching our application were more interested and encouraged to participate to our social experiment.

The protocol used to capture F2F interactions was `SOCIOMAC`. At the end of the experiment we collected around 1800 proximity packets and saved all the raw data coming from the readers into a log file of 10 MB. The parsed data is publicly available in different network file-formats at [39]. Among all the proximity packets, around 1360 contacts lasted just 1 second, while 220 contacts lasted 2 continuous seconds. The average temporal density of the evolving network, equal to $\Lambda = 3.220 \cdot 10^{-5}$, has the same order of magnitude of the DIAG0 experiment.

Figure 2.10a shows the resulting social graph built by using a resolution $\delta = 20$ seconds. Specifically, Figure 2.10a shows the network arranged by the Fruchterman-Reingold layout, while Figure 2.10b depicts the network drawn using the ForceAtlas2 layout[2]. As can be easily noticed from the second figure, the size of each community is, on average, less than the size of the groups in the DIAG0 experiment (see Figure 2.9b). This is due to the fact that in the MACRO experiment groups of people visiting the exhibition usually did not know other visitors. On the contrary at the department, there was a more "sociable" environment, where students usually come into contact with other students. This is also confirmed by the other network metrics, such

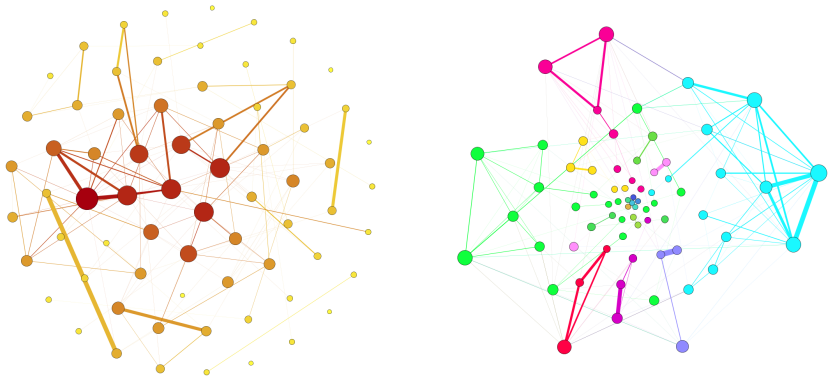
[2]Notice that thinner edges could not be visible in the printed version of this thesis.

as an average degree of 2.316, an average path length equals to 4.221 and a density of 0.02. However, the most surprising fact is that the modularity in the MACRO experiment is the same of the one calculated in the DIAG0 experiment, namely $Q = 0.836$. This result suggests how these kinds of F2F social networks exhibit a modularity structure quite simile.

2.4.2.3 WSDM 2013 conference

On February 2013 we deployed a real-world social experiment in Rome during the WSDM Conference [33] (see Figure 2.13a), where 69 attendees agreed to wear our tags running, for the first time in a real-world context, the `PROXMAC` protocol. The main purpose of the experiment was to collect data to study how F2F interactions can possibly influence the wisdom of a group of people, which is usually called the "wisdom of crowds" phenomenon [187] (see Chapter 4 for a detailed description). Compared to the two previous experiments, here the scope of the interaction was quite different. While in the DIAG0 and MACRO experiments people talked to others without having a task to solve, in the WSDM experiment people were asked to share their own opinions with other participants according to the wisdom of crowds game. However, people were not constrained to talk with a restricted group of other participants, but they were allowed to interact with anyone they wanted. The experiment was deployed during the lunch break, where people are usually more prone to socialize. The total experiment lasted around 1.5 hour, but 50 minutes were allocated for the social interaction, thus collecting data from a large area, including several rooms, the corridor and common spaces of the Auditorium Antonianum [1], the place hosting the conference. As already happened in the MACRO deployment, we installed a dedicated network infrastructure to connect all the readers to a main server.

At the end of the experiment we collected more than 23000 single proximity packets. Notice that, a similar number of packets was obtained in the DIAG0 experiment, but in that case the employed protocol was `SOCIOMAC` and the experiment lasted 5 days. This gives a clear evidence of the different policies adopted by the two MAC protocols. While `SOCIOMAC` is preferred to deploy long-lasting experiments where interactions are quite durable in time, `PROXMAC` is used in short-lasting experiments having fast-changing interactions. Another aspect highlighting the different features of the protocols is the distribution of contacts. We have seen that in the DIAG0 and MACRO experiments the maximum contact duration was 2 seconds; vice versa in WSDM, the distribution ranges from 1-second contacts to 50-second contacts, following a power-law trend. As a consequence, the average temporal density is higher of two orders of magnitude, approaching a value of $\Lambda = 0.003$. Although a higher average temporal density may be due to the different scope of the experiment (col-



(a) Fruchterman-Reingold layout (b) ForceAtlas2 layout

Figure 2.11: The social graph related to the WSDM experiment. Figure 2.11a shows the network drawn by using the Fruchterman-Reingold layout, while the network in Figure 2.11b is drawn by employing the ForceAtlas2 layout.

laborative versus natural), `PROXMAC` was able to capture many more packets thanks to its different policy optimized for fast interactions.

Figure 2.11 shows the social network built by aggregating all the evolving history over time. The process of aggregation was carried out by using the incremental mechanism with $\eta = 5$ and $\Phi = 3$. This means that an edge was built only if at least 5 proximity packets were collected, each of which had to be captured within a window of 3 seconds. Such a resolution was chosen according to the results obtained in Section 2.4.1, while that η empirically gives a good chance to record a real interaction in a wisdom of crowds context. Indeed, assuming to choose a lower η, an edge may be established by the simple fact that two users can bump into each other just for a moment.

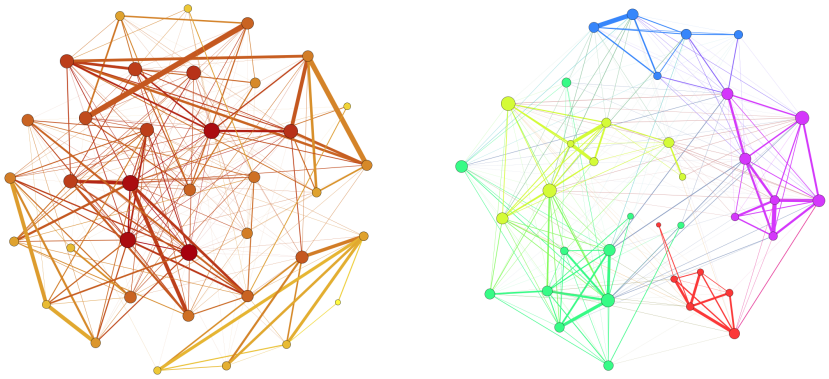
As in previous graphs, darker and bigger nodes of the network in Figure 2.11a represent people that in the whole experiment accumulated the higher number of F2F interactions (i.e., the degree), while the thickness of an edge is proportional to the number of interactions observed between two nodes over the whole time. The graph is made of 69 nodes and 133 undirected edges. The average degree is 3.855, while the density has a value of 0.057. The network diameter and the average path length are 7 and 3.222, respectively. Only 6 nodes are isolated, while the maximum degree is 13. Analogously, Figure 2.11b shows the same network but emphasizing its communities. In this case, the total number of communities is 17, and the value of the modularity is 0.72. Even though still pretty high, the modularity of the WSDM experiment is lower than the DIAG0 and MACRO ones. This is explained by the fact that, being a social experiment where people collaborate to achieve a goal,

the sociality and the interconnections among different communities are more stimulated, thus decreasing the modularity.

2.4.2.4 Students at the department (DIAG1)

A second experiment at our department was deployed on May, 16 2013 to keep studying the wisdom of crowds phenomenon. We recruited 37 students from the Master's degree in Computer Engineering. The choice of recruiting students with the same education level and studying the same university course was dictated by the fact of observing how individuals having similar educational characteristics perform on the wisdom of crowds phenomenon with respect to an heterogeneous population. Furthermore, a smaller population of humans allowed us to compare the corresponding features with the ones studied in the previous larger networks. The social interaction of the experiment lasted 20 minutes, where more than 30000 single proximity packets were exchanged between tags running `PROXMAC` and then collected by the readers. Such a high number of exchanged packets suggests how there was a very intensive social activity in the students' network. As already observed in the WSDM experiment, the distribution of contacts follows a power-law trend, here starting from contacts lasting just 1 second and ending to contacts lasting 40 consecutive seconds. The corresponding average temporal density is $\Lambda = 0.035$, which is even higher than the WSDM's one. This gives a clear evidence about the higher social exchanges between students with respect to participants of the WSDM conference. This is explained by the fact that the students selected for our experiment knew each other and, due to its young age and strong enthusiasm for the proposed game, they collaborated as much as they could.

Figure 2.12 shows the social graph of the DIAG1 experiment. The parameters of the incremental technique chosen to aggregate the evolving history were $\eta = 10$ contacts and $\Phi = 3$ seconds. With respect to the WSDM experiment, where $\eta = 5$, in this deployment we have doubled such a value to further filter false positives (i.e., interactions that actually did not occur). In other terms, while participants in WSDM were more widespread than in DIAG1 just interacting when really necessary, in the DIAG1 experiment students tended to join larger groups of individuals, so the condition on the link establishment in DIAG1 must be stricter. To better clarify the difference of context we observed, suppose to be in a large area and want to talk with just somebody. This scenario is what we had in the WSDM experiment. Now, suppose to be in a smaller area and know most of the other few participants. You probably want to join the discussion in larger groups where two or three people interact, while others just listen to the discussion. This is what we observed in the DIAG experiment. For this reason, the number of contacts to establish a



(a) Fruchterman-Reingold layout (b) ForceAtlas2 layout

Figure 2.12: The social graph related to the DIAG1 experiment. Figure 2.12a shows the network drawn by using the Fruchterman-Reingold layout, while the network in Figure 2.12b is drawn by employing the ForceAtlas2 layout.

contact as real as possible should be bigger.

The different kind of interaction we observed in this experiment leads to a "flatter" network structure (see Figure 2.12a) than in the previous wisdom-of-crowds deployment. In other words, while in WSDM some individuals have a degree much higher than others, in DIAG1 students tend to have a similar level of relevance in the network. This indicates that, in general, each user interacted with most of the other students. This is confirmed by a lower value of a modularity $Q = 0.515$ and by the network diameter equals to 4. This two metrics suggest how the social network formed in the DIAG1 experiment is rather connected, and so the population collaborated in a quasi-uniform way. The total number of communities, highlighted in Figure 2.12b, is 5, while the average path length is 1.736. A further proof about the highly cohesiveness of the network is given by the average degree, which is 12.649 and by the graph density, equals to 0.351. Considering that the size of the network is 37, an average degree of such a value is pretty high.

2.4.2.5 Priverno's country fair

Similarly to the previous experiments and adopting the same modalities, on May 11, 2014 we deployed another real-world social experiment in Priverno (LT), Italy, during a country fair and recruiting 60 volunteers (see Figure 2.13b). As usual, each participant wore a tag running `PROXMAC` in order to track their interactions. They were free to move and interact within a delimited monitored area of around 15×15 meters in a green park. The total

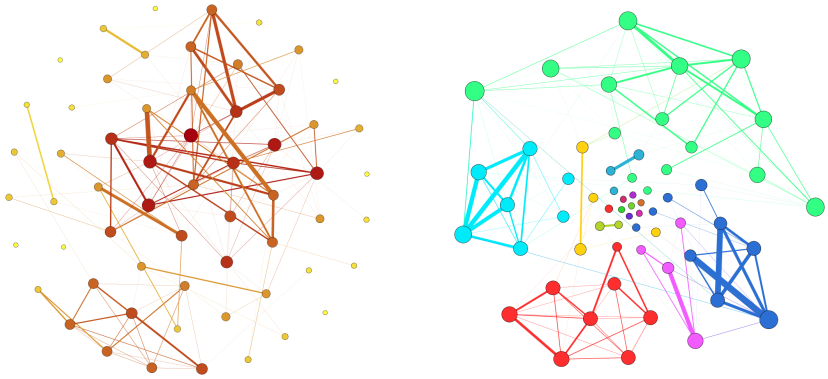
(a) (b)

Figure 2.13: (a) WSDM experiment, (b) Priverno experiment

experiment lasted less than 30 minutes and the main purpose was to study the same social phenomenon of WSDM and DIAG1. The interaction part of the experiment lasted around 10 minutes, time in which we collected more than 11000 single proximity packets, distributed according to a power law with the same maximum value found in DIAG1. The average temporal density is $\Lambda = 0.012$, lower than in DIAG1, but having a same order of magnitude. This value proves what we observed during the experiment, namely a good collaboration among participants to try and solve the questions we gave them. Even though people in Priverno did not know most of the other participants, as instead happened in DIAG1, they probably were more stimulated to interact with strangers thanks to the limited area just equipped for the experiment. Conversely, in WSDM, the experiment overlapped with the lunch break and people could wander throughout the location.

The graph in Figure 2.14 depicts the aggregated result of the full social interaction. As in the DIAG1 experiment, we used the incremental mode with $\eta = 10$ contacts and $\Phi = 3$ seconds to build the social graph from the evolving history. It is formed by 60 nodes (i.e., the number of participants) and 128 undirected edges, namely the number of distinct interactions between them.

The resulting network structure obtained in this experiment (see Figure 2.14a) is quite different with respect to the DIAG1's one. While in DIAG1 students tended to cluster in a big community, in Priverno participants had a more natural behavior forming a heterogeneous social network such as in WSDM. The average degree is 4.267, the network diameter is 9 and the average path length is 3.713. 7 users are isolated, while the maximum degree is 10. Finally, the density of the graph is $\lambda = 0.072$, which is a little higher than the WSDM's one, but still belonging to the same order of magnitude. Figure 2.14b shows the same graph emphasizing all the 15 communities of the network, but take into account that 7 of them are due to the isolated nodes. The modularity of the network, equals to 0.743, approaches the modularity of WSDM, which was 0.72.



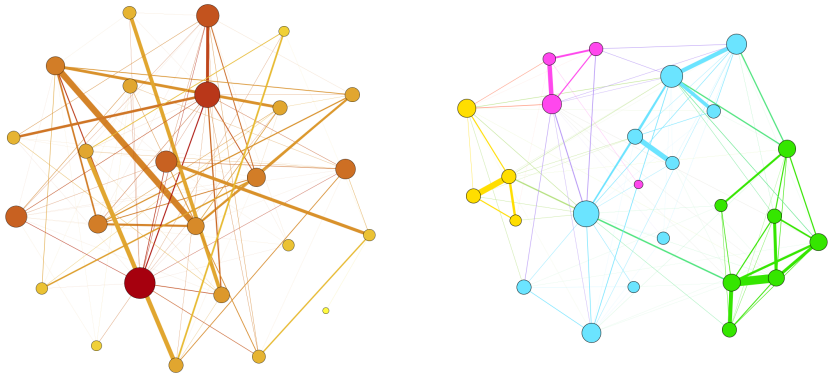
(a) Fruchterman-Reingold layout (b) ForceAtlas2 layout

Figure 2.14: The social graph related to the experiment deployed in Priverno. Figure 2.14a shows the network drawn by using the Fruchterman-Reingold layout, while the network in Figure 2.14b is drawn by employing the ForceAtlas2 layout.

2.4.2.6 Students at the department (DIAG2)

The latest experiment deployed during our research activity and studying the wisdom of crowds phenomenon was again at our department. We involved a total of 25 students following a same course of study and wearing tags running `PROXMAC`. The experiment was arranged on May 24, 2014 at the open area of the department, where students usually study, relax and have lunch. The social part of the experiment lasted 10 minutes. At the end of the experiment we collected around 8350 single proximity packets constituting a distribution of contacts which follows a power-law trend as in the other deployments. The average temporal density is $\Lambda = 0.040$. This value is quite near to the average temporal density computed on DIAG1, which was 0.035. This suggests how, once again, small groups of individuals knowing each other are much more active in terms of sociability than in other kind of networks, at least when participating to social games.

Figure 2.15 shows the social graph, related to this experiment, which is composed by 25 nodes and 103 distinct undirected edges. As usual, the technique used to build the aggregate network starting from the evolving history was the incremental method with $\eta = 10$ contacts and $\Phi = 3$ seconds. More in detail, Figure 2.15a emphasizes the structure of the network, which appears alike to the DIAG1's one. Most of the nodes have a similar degree, excepting a couple of individuals standing out. On the other hand, Figure 2.15b gives a clear evidence of the 4 communities in the network. The value of the



(a) Fruchterman-Reingold layout (b) ForceAtlas2 layout

Figure 2.15: The social graph related to the DIAG2 experiment. Figure 2.15a shows the network drawn by using the Fruchterman-Reingold layout, while the network in Figure 2.15b is drawn by employing the ForceAtlas2 layout.

modularity, equals to 0.471, is the lowest among all the experiments. This is due to the high number of interconnections among the 4 modules. Other metrics with similar values to DIAG1 are the diameter, which is equals to 4, the average path length having a value of 1.773 and the network density of 0.343. In this case, the average degree of 8.24 is lower, but it strictly depends on the size of the network. Indeed, while in DIAG1 the size was 37, here it is 25. Nevertheless, if we compute the ratio between the average degree and the size of the network, we get an equivalent result in both the two experiments.

2.4.3 Simulation on larger and denser graphs

A simulation on graphs having many more nodes and higher densities allows us to analyze what kind of situation we may encounter in future and bigger real-world scenarios. First of all, we remind that `SOCIOMAC` sends report packets to the readers only after 8 cycles, while `PROXMAC` can report after every receiving operation if new contact packets have been received. Due to this policy, in dense networks `SOCIOMAC` can more easily experience a buffer overflow (tags have room for only 4 packets) and consequently discards new incoming contacts. We set up a simulation for different temporal densities and graph sizes for a total of 1200 time-steps. For each step, we generate a random graph based on the Erdős-Rényi model [93], with $n = \{60, 100, 200, 500\}$ and $p = \{0.01, 0.02, 0.05, 0.1\}$, where n is the number of nodes and p is the probability of including an edge in each graph over the evolving history. Notice that for a sufficient large amount of time-steps, the probability p approximates the

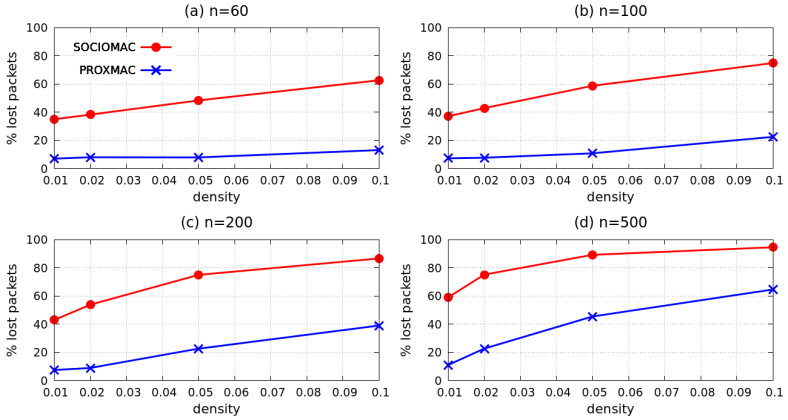


Figure 2.16: The percentage of lost packets for different graph densities and number of nodes.

average temporal density of the whole evolving network. Figure 2.16 shows the results of our simulation.

The equivalent case to the real-world experiment deployed in Priverno is depicted in Figure 2.16(a), where the number of nodes is 60 and the average temporal density is 0.01. `PROXMAC` turned out to be very efficient in this case, where the percentage of lost packets is around 7%. Vice versa, `SOCIOMAC` was not able to collect almost the 40% of the total possible packets. A similar behavior can be observed also in the other cases considered in the simulation where `PROXMAC` is able to better support denser graphs. Along with an improved policy of packets forwarding, we recall that the whole `PROXMAC`'s phase lasts less than the `SOCIOMAC`'s one. This brings to a faster actions cycle in `PROXMAC`, which is able to execute each operation several times in every time-step. Still considering the plot with $n = 60$, `PROXMAC` has a similar percentage of lost packets for higher densities as well, while the `SOCIOMAC`'s trend slightly increases. Their behavior changes little for the graph having 100 nodes, but increases in graphs with $n = 200$ and $n = 500$. However, for lower, but more realistic densities (e.g., 0.01 and 0.02), `PROXMAC` was able to maintain a very low level of loss. On the contrary, `SOCIOMAC` starts with a level of loss equals to 40% and 60%, respectively. Then, their trends increase and reach more than 80% for `SOCIOMAC` with a density 0.1, while `PROXMAC` amounts to 40% in $n = 200$ and 60% in $n = 500$. We recall, however, that very high densities are unlikely in real-world scenarios, as already observed in our social experiments.

2.5 Concluding Remarks

In this chapter we introduced the concept of F2F social network, a network made by individuals physically interacting each other in real-world scenarios. We then described some applications that can be deployed using the SocioPatterns Sensing Platform, an infrastructure composed by active RFID tags and RFID readers. The former are worn by humans to track their interactions, while the latter are required to collect interaction packets. The first practical step in our study concerned the analysis of `SOCIOMAC`, the default MAC protocol provided with the tags. Afterwards, we designed a more flexible and suitable MAC protocol, named `PROXMAC`, to be used in social experiments lasting a limited amount of time and having fast-changing interactions between users. Employing this infrastructure we then deployed several real-world social experiments. Preliminary experiments in ideal conditions were needed to better analyze the protocols' behaviors and performances. Afterwards, we deployed two first long-lasting social experiments using `SOCIOMAC` and other four more dynamic social experiments using `PROXMAC`. Specifically, these last four experiments were used to study the wisdom of crowds phenomenon, an application having an increase of interest in the research community and that will be extensively discussed in Chapter 4.

The next chapter opens the second part of the thesis discussing about decentralized computation on evolving networks. This, along with the wisdom of crowds phenomenon, represent two major applications on social networks we extensively investigated during our research activity.

Part II

Computation over Evolving Social Networks

Chapter 3

Decentralized Computation of Centrality Scores: the case of PageRank

Link analysis has become a fundamental tool in Web information retrieval over the past 15 years, completely reshaping this area [141]. Moreover, link analysis has proved extremely effective in other domains, virtually in any scenario in which a suitably defined network captures important aspects of a system. Examples include proximity graphs, in which links connect mobile nodes when they are sufficiently close, the graph of phone calls (or messages) spanning users of a cellular network, the myriad of static or dynamic networks that can be extracted from user activity logs in social networking platforms. The ability of computing accurate measures of centrality in the aforementioned scenarios can benefit a number of advanced services such as recommender systems, information spreading, data collection and advertising to name a few. The problem is that many of the above networks exhibit extremely dynamic behavior, their link structure changing frequently over time, whereas many algorithms designed for Web information retrieval apply to relatively static networks. These considerations also apply to PageRank, a fundamental building block to assign centrality scores to Web pages [164, 140, 60]. In fact, the notion itself of PageRank over an evolving network is not clear. The approach usually adopted in the literature for dynamic networks is to design algorithms that at every step maintain an accurate estimation of the PageRank vector computed over the current snapshot of the network (e.g., [51, 50]), which entails a definition of centrality for evolving networks. While this approach is clearly viable since all necessary information is typically available to the service provider, it is not clear that PageRank computed on the current snapshot provides an accurate picture of nodes' relative importance in an evolving

network.

Furthermore, maintaining Pagerank under continuous updates can be computationally very expensive in a centralized setting, even if efficient, incremental algorithms [50] are used.

An alternative approach that can help to reduce the computational burden, demands the computation to distributed lightweight applications running at the nodes of the network. In general, fully decentralized centrality computation in social communities has been considered in the past [168, 169] and also more recently [200]. In [178], the algorithm proposed in [49] based on Monte Carlo statistical sampling, is used to achieve a fully distributed version of the PageRank algorithm.

Our contribution. In this chapter, we consider two notions of centrality for evolving networks, which go beyond the standard approach of defining centrality at time t as simply PageRank (or an alternative measure) computed over the t-th snapshot. The notions we propose are consistent with PageRank, in the sense that they correspond to computing PageRank over a *static*, weighted, directed network. Considered any time t, this network may either reflect the "expected" network topology at time t, or it may reflect the "average" topology over a suitable window of the most recent Δ snapshots. In the first case, expectation is defined with respect to the generally unknown process describing the evolving network. We also show that, when the evolving network satisfies additional statistical properties the former notion we consider corresponds to the score vector dynamically and continuously maintained by a decentralized heuristic for Monte Carlo PageRank sampling previously proposed in [178]. While static networks are considered in [178], the algorithm they propose can be run continuously over an evolving, directed network. However, since real-life evolving networks may exhibit significant non-homogeneous properties, we propose a modified heuristic which, intuitively, assigns each node a score that mostly reflects nodes' centrality over the evolving network's recent past and also better handles sparse networks by suitably adapting the sampling process. While this heuristic still provides the same guarantees on homogeneous networks, it addresses some issues that may negatively affect the basic one. We perform an extensive experimental analysis on both synthetic and real, publicly available, evolving network datasets. Results support the validity and feasibility of the approach we propose. In particular, in case of *slow evolving networks*, our modified heuristic closely approximates the centrality score defined according to the second notion we propose even when only the current or few recent snapshots are considered. This, in our opinion, supports the validity of Monte Carlo sampling for authority computation in slow evolving networks. When the network evolves fast and is strongly non-stationary,

satisfactory results can be obtained only observing the network with larger aggregation windows.

Roadmap. We discuss related work in Section 3.1. In Section 3.2, we formalize the notion of evolving networks and we discuss the design of suitable centrality measures for them. We provide experimental evidence showing that Pagerank computed over single snapshots of an evolving network may not provide useful information about node centrality: at least in some cases, this information is better captured by considering aggregate behavior over consecutive windows of suitable size. In Section 3.4 we present the basic heuristics we propose and analyze their properties in the case of homogeneous evolving networks. We also discuss potential optimizations and some issues arising from the underlying assumption of a synchronous scenario. We further address the issue of non-homogeneous evolving networks, and we present a heuristic to address these cases. Section 3.5 presents experimental results for the heuristics we propose, both on synthetic and real datasets.

3.1 Related work

Over the past decade, a considerable amount of work has considered the problem of computing link related scores in a network, the most prominent example being PageRank [69, 164]. In this section, we revised work that is more closely related to the problem we study.

Main approaches to the PageRank computation. A number of methods to compute PageRank and related measures have been proposed in the literature. Most approaches exploit the relationship between these importance measures and the eigenvectors of suitably defined matrices to exploit algebraic methods. Such was also the original approach to PageRank computation [164]. Very nice references discuss the use of algebraic methods in the computation of PageRank and other Eigenvector based measures [140, 60, 141]. An orthogonal approach to PageRank (and other related measures) computation exploits relationships between the score vectors under consideration and the stationary distributions of suitably defined random walks. Such methods use a possibly small number of simulated random walks (per node) to accurately approximate the value of PageRank at every node of the network [101, 49]. This approach naturally lends itself to estimate PageRank (and variants thereof) upon incremental updates, as we discuss in the next paragraph.

PageRank over evolving networks. Maintaining PageRank under network updates is an important and non trivial task. A first issue is to decide

what the importance measure should reflect. Should it at any point reflect centrality in the current snapshot of the evolving network, or should the centrality score depend on the past history of the evolving network? A common approach followed in the literature is to require the PageRank to reflect node centralities on the current snapshot of the evolving network. This is for example the scenario considered in [50, 51]. In [50], the authors propose a Monte Carlo method to maintain PageRank with provable accuracy over a sequence of incremental updates. In particular, the authors prove that for updates that involve m edges, a total work at most $O(n \log n \log m/\epsilon^2)$ is needed, where n is the number of vertices in the network and ϵ is the desired accuracy. In [51], the authors do not impose any constraints on the computational costs. The focus is actually on strategies to probe nodes of the graph over time, so that PageRank computed with respect to the current (approximate) image of the graph closely approximates the true PageRank vector.

It should be emphasized that both these approaches assume a setting in which the real or approximate (in the case of [51]) network on which PageRank has to be computed is fully available to the algorithm that can gain a complete knowledge of it by, for example, crawling it or probing its pages. In contrast, in the scenario for which we design our algorithms the network is not fully available but rather each vertex owns the its outgoing links that might constitute private information about the node itself.

Distributed PageRank computation. Computing PageRank on massive Web graphs can be computationally expensive, so the problem of distributing the computational load has been considered in the literature [197, 168]. In particular, [168] addresses PageRank computation over a P2P network. This paper assumes a pretty different distributed model than the one we consider in our work. In particular, each node has knowledge of a portion of the graph over which PageRank must be computed, which in general differs from the P2P network. On the other hand, nodes can select peers in the P2P network uniformly at random or according to some other distribution. Also, sinks are (albeit indirectly) removed from the graph.

The recent paper [178] is more closely related to our line of work. They consider the distributed computation of PageRank on static networks, in which each nodes locally computes its own rank score. The basic algorithm they propose corresponds to Algorithm `FDSAMPLE` presented in Section 3.2. We remark that in this work we analyze this algorithm in the much more general scenario of evolving networks and also propose a refined version in Section 3.4.3, which performs better on the real dataset we consider in Section 3.5.

3.2 Preliminaries

We are interested in the problem of computing *centrality* scores for the nodes of an *evolving network* in a fully distributed way. This section introduces and discusses basic aspects and issues of this problem.

3.2.1 Recap on evolving networks

We have seen in Chapter 1 that in its simplest formulation, a deterministic evolving network is simply an infinite sequence $G_0, G_1, G_2, \ldots$ of graphs over the same vertex set V and dynamical edge sets E_t, with $t = \{0, 1, 2, \ldots\}$.

Here we are interested in a general framework in which, for every t, the graph observed at time t is a realization of some stochastic process. For this reason, we adopt a definition that follows and extends [48],

Definition 4. Assume an underlying family $\mathbf{G}$ of directed graphs over the same vertex set V of cardinality n. An *evolving network distribution* $\mathcal{G}$ over the sample space $\mathbf{G}$ is an infinite sequence $\mathcal{G} = \{\mathcal{G}(t)\}_{t \geqslant 0}$ of probability distributions over $\mathbf{G}$. We call *evolving network* a realization $G = \{G(t)\}_{t \geqslant 0}$ of $\mathcal{G}$, i.e., a temporal sequence of graphs from $\mathbf{G}$.[1]

To simplify notation, we denote by $\mathcal{G}(t)$ both the probability distribution at time t and the corresponding random variable. Given a realization G of $\mathcal{G}$, $G(t)$ is called the t-th *snapshot* of G.

Stationary and homogeneous evolving networks. We call an evolving network distribution $\mathcal{G}$ *stationary* if $\mathcal{G}(t)$ does not vary with t and *homogeneous* if, for every k and for every $t_1, \ldots, t_k$, distributions $\mathcal{G}(t_i)$'s are *identically* and *independently* distributed.[2] A homogeneous network, for example, reasonably describes a network in which edges are subject to a rapidly mixing ergodic Markovian process [79], so that the network has enough time to approximately achieve stationarity between consecutive probes.

[1] One might adopt a simpler definition in which the underlying network is the complete graph over vertex set V and, for every time t and for every arc (i, j), we consider the probability that (i, j) exists at time t. Though possible, this definition is less general, since it does not suitably model spatial dependencies among arcs.

[2] Note that this is a stronger notion than stationarity. Consider for example a simple 2-vertex Edge-Markovian graph [79] consisting of vertices a and b and arcs (a, b) or (b, a) according to the following deterministic rule: (a, b) exists at time $t + 1$ if (b, a) existed at time t and vice versa. If the network is initialized in one of the two states with probability 0.5 at time 0, it is easy to see that we have a stationary process according to the standard definition, but clearly this is no homogeneous evolving network distribution.

Induced matrices. Let $G = (V, E)$ any network. The *adjacency matrix* of the network is defined as the $n \times n$ matrix $\mathbf{L}(G)$, such that $\mathbf{L}_{ij}(G) = 1$ if arc (i, j) exists, $\mathbf{L}_{ij}(G) = 0$ otherwise. Denote by $\mathcal{N}_G(i)$ the set of neighbors of node i (that is, if $j \in \mathcal{N}_G(i)$ then (i, j) is an arc of G) and set $d_G(i) = |\mathcal{N}_G(i)|$ the degree of node i in G. The *transition matrix* $\mathbf{Q}(G)$ of G is defined as follows:

$$\mathbf{Q}_{ij}(G) = \begin{cases} \frac{1}{d_G(i)}, & \text{if } d_G(i) \neq 0 \text{ and } j \in \mathcal{N}_G(i); \\ 0, & \text{otherwise.} \end{cases}$$

Considered any matrix associated to a graph (e.g., the transition matrix), an evolving network $G = \{G(t)\}_{t \geqslant 0}$ induces a sequence of matrices, each associated to a particular network observed at time t. So, for example, $\mathbf{L}(t)$ and $\mathbf{Q}(t)$ respectively denote the adjacency and the transition matrices of the network observed at time t. Obviously, for every $t \geqslant 0$, $\mathcal{G}(t)$ induces distributions $\mathcal{L}(t)$ and $\mathcal{Q}(t)$. More precisely, $\mathcal{L}(t)$ is the probability distribution that assigns to a matrix $\mathbf{L}$ the probability that $\mathbf{L}$ is the adjacency matrix of the evolving network at time t. Note that $(\mathbf{E}_{\mathcal{G}}[\mathcal{L}(t)])_{ij}$ is the (unconditional) probability that arc (i, j) exists in $G(t)$. Analogously, $\mathcal{Q}(t)$ is the probability distribution that assigns to a matrix $\mathbf{Q}$ the probability that $\mathbf{Q}$ is the transition matrix of the evolving network at time t.

In general, for a sequence $\mathcal{X}(t)$ of probability distributions over matrices, we denote by $\mathbf{E}_{\mathcal{G}}[\mathcal{X}(t)]$ the matrix whose (i, j)-entry is $\mathbf{E}_{\mathcal{G}}[\mathcal{X}_{ij}(t)]$. The average is w.r.t. distribution $\mathcal{G}(t)$. Also note that the adjacency matrix of a homogeneous evolving network at time t and t' have the same average; that is, $\mathbf{E}_{\mathcal{G}}[\mathcal{L}(t)] = \mathbf{E}_{\mathcal{G}}[\mathcal{L}(t')]$.

Expected and aggregate networks. We next introduce two definitions that will be used throughout the chapter:

Definition 5. Given an evolving network distribution $\mathcal{G}$, for every t, we call *expected network* at t the weighted graph whose adjacency matrix is $\mathbf{E}_{\mathcal{G}}[\mathcal{L}(t)]$.

Note that the definition above makes no assumption about $\mathcal{G}$.

Definition 6. Given an evolving network G and an integer Δ, we define *aggregate snapshot* at time t the weighted graph $G_\Delta(t)$ with adjacency matrix obtained by collapsing the Δ-the most recent snapshots, namely: $\frac{1}{\Delta} \sum_{\ell=0}^{\Delta-1} \mathbf{L}(t - \ell)$. For every Δ the sequence $G_\Delta = \{G_\Delta(\ell\Delta)\}_\ell$, where $\ell = 1, 2, \ldots$, defines a derived evolving network that we call *aggregate network*.

3.2.2 PageRank

A pervasive measure of nodes' centrality in static networks is PageRank [164, 140]. As already disclosed in Chapter 1, one of the ways to define PageRank [69] is to consider random walks on a graph. Intuitively, this can be thought of as modeling the behavior of a *random surfer* that starts from a randomly chosen web page. Every page the random surfer visits, she either gets bored and quits navigation with a fixed probability, or she selects one of her outgoing neighbors uniformly at random. If the page has no outgoing links (i.e., it is a *sink*), the surfer jumps to a page selected uniformly at random in the network. The rank of a page according to the PageRank algorithm is the probability that a random surfer stops at that page. As previously said, the probability that the surfer continues her random walk is given by the *damping factor* α.

We denote by $\mathbf{A}(G)$ the *modified transition matrix* of G, corresponding to removal of sinks, namely the stochastic matrix such that $\mathbf{A}_{ij}(G) = \mathbf{Q}_{ij}(G)$ if i is not a sink, $\mathbf{A}_{ij}(G) = \frac{1}{n}$ otherwise.[3]

Given these definitions, the Pagerank vector $\boldsymbol{\pi}$ of G is the stationary distribution (equivalently, the main left eigenvector) of the ergodic Markov chain corresponding to the following stochastic matrix: $\mathbf{P} := \alpha\mathbf{A} + \frac{1-\alpha}{n}\mathbf{1}\mathbf{1}^T$. While algebraic methods are a standard approach to PageRank computation, some contributions have proposed to directly use the random walk definition briefly outlined above to perform Monte Carlo sampling of the PageRank distribution [127, 101, 49]. In particular, these contributions rely on an alternative and equivalent definition of Pagerank, whereby[4]

$$\boldsymbol{\pi}_i = \frac{1-\alpha}{n} \sum_{k=0}^{\infty} \alpha^k \sum_j \mathbf{A}^k_{ji}$$

or, in matrix notation

$$\boldsymbol{\pi} = \frac{1-\alpha}{n}\mathbf{1}^T \sum_{k=0}^{\infty} \alpha^k \mathbf{A}^k = \frac{1-\alpha}{n}\mathbf{1}^T(\mathbf{I} - \alpha\mathbf{A})^{-1}.$$

Specifically, they use the fact that the alternative definition above amounts to defininining $\boldsymbol{\pi}_i$ as the probability that the random walk defined by the following process ends at node i (see also [127]):

Process 1. Start a random walk at a node chosen uniformly at random with probability $1/n$; at each step, the random walk terminates with probability $1-\alpha$, while with probability α the transition occurs according to matrix $\mathbf{A}$.

[3] We write $\mathbf{Q}$ and $\mathbf{A}$ whenever G is clear from context.

[4] In the remainder, $\mathbf{1}$ denotes the column vector with unit components.

Remark. Though probably obvious, we emphasize that the definition(s) of Pagerank given above naturally extend to more general Markov chains, i.e., when $\mathbf{A}$ is a stochastic matrix, but not necessarily the modified transition matrix of an unweighted graph. We use this fact extensively in the next Section 3.3, when defining Pagerank over the expected network or aggregate snapshots.

This naturally translates to a first, obvious, distributed algorithm `IDEALSAMPLE`, to statistically sample $\boldsymbol{\pi}$ [49], given in Figure 3.1.

At every step, every node generates r *tokens* on average[5] and receives a certain number of tokens from its neighbors. Then, each token is propagated (and thus performs a random walk) in the network according to process 1. We say that a certain token *dies* at a node i whenever the corresponding random walk terminated and the token is removed from the network. In `IDEALSAMPLE`, each node i keeps a counter $\mathbf{C}_i$ of the number of tokens that died at i. The algorithm assumes that vertices have a full view of the network. Indeed, a sink sends each token received and that does not die to a randomly chosen vertex of the network (notice that if i is a sink then $A_{ij} = 1/n$ for all j).

```
IDEALSAMPLE(i, A, α, r)
Require: node-id i, matrix A, damping factor α, rate r
 1: C_i = 0
 2: for every step do
 3:   T = incoming() ⋃ generate(r)
 4:   for every token in T do
 5:     dies = rnd(0, 1)
 6:     if dies ⩾ α then
 7:       C_i = C_i + 1
 8:     else
 9:       send token to vertex j with prob A_ij
10:     remove token from T
```

Figure 3.1: Ideal distributed sampling algorithm.

The main limitation of `IDEALSAMPLE` is the assumption that each node has knowledge of all the nodes in the network. This assumption may be unfeasible in fully decentralized scenarios, especially in the case of real-world evolving networks.

[5] r is not necessarily an integer in which case we intend that the node generates at each step an average of r tokens. For example, $r = 1/n$ corresponds to a node that generates on average one token every n iterations.

3.3 Defining Pagerank on evolving networks

Real life networks typically evolve over time. Sometimes, the frequency of updates is low enough that an incremental update algorithm can dynamically maintain accurate approximations of the PageRank vector as the network evolves. In other cases, the network may undergo major changes over relatively short intervals.

In general, defining centrality scores on evolving networks presents some conceptual difficulties. The approach usually followed for PageRank in the literature (e.g., [51, 50]) is to require that the estimated PageRank vector be at any point an accurate estimation of PageRank computed on the current snapshot of the evolving graph. This requirement implicitly entails a definition of centrality. Namely, the centrality vector at time t is simply the PageRank computed over the t-th snapshot of the evolving network. While this definition appears natural, it is not clear that it allows an adequate characterization of nodes' centralities in an evolving network in all cases. In some cases, centrality[6] may be better appreciated by varying the temporal scale of observation. For example, for an evolving network representing user on-line activity within a social networking platform, the PageRank of the current snapshot might provide little information about the relative centralities of nodes in the network in the longer term.

3.3.1 An experimental outlook

In order to test the intuition above, we first study the evolution of PageRank on aggregate networks obtained from original, real ones for different values of the aggregation window Δ.

We conducted experiments on two real evolving networks: the CAIDA dataset [144], tracking the evolution of a network of autonomous systems, and a network of Facebook wall posts [194], tracking wall posts of a sample of Facebook users over a period of more than one year. More details about these datasets are given in Section 3.5.1. Assuming G as the original network resulting from any of the two datasets, the first question we addressed was the following: *What is the (rank) correlation between the Pagerank vectors computed over two randomly chosen, consecutive aggregate snapshots $G_\Delta(t)$ and $G_\Delta(t+\Delta)$? How does it vary as Δ increases?* Notice that $G_\Delta(t)$ and $G_\Delta(t+\Delta)$ correspond to the aggregation of two consecutive (but disjoint) windows of snapshots from the original network. Intuitively, a correlation growing with Δ implies that centrality of the nodes emerges by increasing the temporal scale at which the network is observed.

[6] And possibly other properties.

For each value of Δ, we considered the average value of Spearman's rank correlation coefficient ρ_S (see Chapter 1) between the PageRank of $G_\Delta(t)$ and $G_\Delta(t+\Delta)$ for 100 values of t, sampled uniformly at random over the original snapshots. We considered Δ varying between 1 (original evolving network) and 50. Figure 3.2b highlights that the behaviors of these two datasets are quite different. In the case of CAIDA, Pagerank correlation is rather high and (slightly) decreases with Δ. Conversely, in the case of Facebook, correlation is very small over pairs of consecutive snapshots, while it increases as aggregate networks corresponding to larger window sizes are considered[7]. These results are also confirmed by the analysis of the Person's correlation coefficient ρ_P between the adjacency matrices of $G_\Delta(t)$ and $G_\Delta(t+\Delta)$ (see Figure 3.2a).

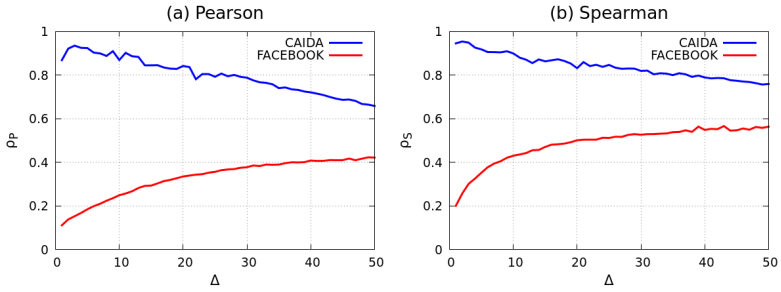


Figure 3.2: Pearson correlation between the adjacency matrices (a) and Spearman correlation between the $\boldsymbol{\pi}$ vectors (b). $\Delta = [1, 50]$

These results represent behaviors at roughly opposite sides of the spectrum. To better understand what kind of dynamic the two networks have, we computed the fraction of changing edges between t and $t+1$, namely

$$\Psi\left(E(t+1), E(t)\right) = \frac{|E(t+1) - E(t)| + |E(t) - E(t+1)|}{|E(t+1) \bigcup E(t)|}$$

where $E(t)$ is the edge set at the time-step t. As shown in Figure 3.3a, the CAIDA dataset represents a network whose topology evolves slowly over time (apart from a few peaks). This implies that the PageRank computed on the current snapshot of the network adequately represents existing centrality relationships over a temporal window including the recent past and the next future. Vice versa, the Facebook dataset represents a fast evolving network, in which topology exhibits little short-term correlation, as also reflected by Figure 3.3b (the vast majority of the edges change between consecutive snapshots). Not surprisingly, trends in node centrality are better captured by increasing the temporal resolution at which the network is observed.

[7] Observe that a minimum amount of correlation is present due to the $(1-\alpha)/n$ term in PageRank definition.

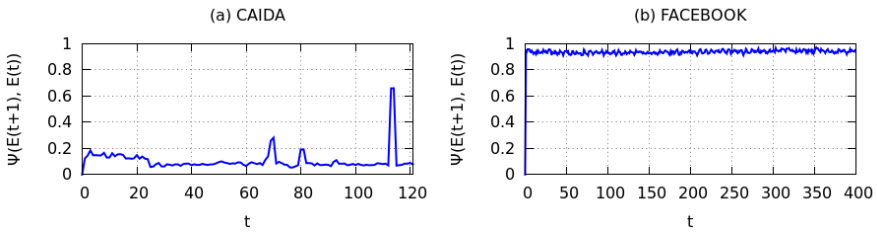


Figure 3.3: Fraction of changing edges over time.

A further analysis is given by using the Recurrence Plot (RP) (see Chapter 1). Although RP is one of the most powerful tool to analyze phase space trajectories in descriptive statistics and chaos theory, it can be also used on evolving networks after selecting an appropriate metric of distance able to somehow correlate separate snapshots. Since we apply the RP on rank vectors we need to deal with n-dimensional trajectories, so a simple subtraction between them is not feasible. Using the Spearman's rank correlation coefficient, a natural expression of distance may be $d(x, y) = 1 - \rho_S(x, y)$, where x and y are two general variables to which the distance metric d has to be applied. However, as discussed in [192], the dissimilarity thus obtained does not guarantee the triangle inequality. The authors propose two distance alternatives:

$$d_1(x, y) = \sqrt{\frac{1}{2}(1 - \rho_S(x, y))}, \quad d_2(x, y) = \sqrt{(1 - \rho_S^2(x, y))}$$

The two functions are plotted in Figure 3.4. Since $-1 \leqslant \rho_S \leqslant 1$, we ranged the x-axis coordinate of the figure between -1 and 1. As can be easily noticed, the two distances are always positive in that range. However for our purpose, since we needed a symmetric behavior, we decided to employ d_2. In other words, a correlation coefficient of $\rho_S = -1$ has the same meaning of $\rho_S = 1$ in this kind of analysis, consequently d_2 is the most appropriate distance metric to be used.

Once we determine a proper distance metric, opposite to the Spearman's rank correlation coefficient, we need to find a distance threshold ε beyond which the RP can be drawn. Actually, this kind of study is often empirically accomplished by trying reasonable values over and over again until something significant appears. However, a first rough estimation of a suitable value can be achieved from the plots of Δ-consecutive correlations shown in Figure 3.5. In this case, for each value of Δ, we measured the Spearman's rank correlation coefficient and the related distance between consecutive pairs of $\boldsymbol{\pi}_\Delta(t)$ and $\boldsymbol{\pi}_\Delta(t + \Delta)$, for $t = 1, 2, \ldots$. We can notice that, as for the CAIDA correlation curve (see Figure 3.5a), a reasonable (neither too much relaxed, nor too much

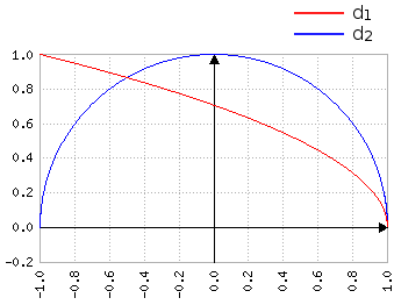
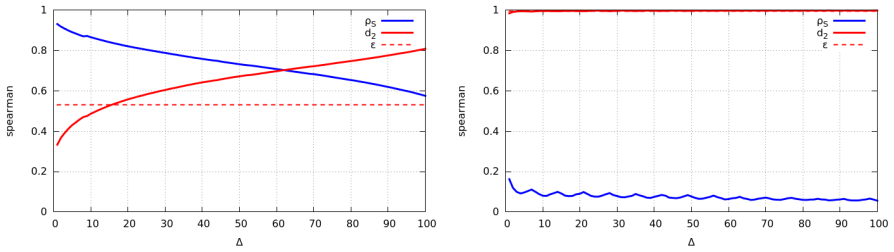


Figure 3.4: The two distance functions related to the Spearman's rank correlation coefficient.

strict) correlation coefficient could be 0.85, consequently $\varepsilon = \sqrt{(1 - 0.85^2)} = 0.53$. Such a value of ε intersects the distance curve for $\Delta \approx 15$. Regarding Facebook, as shown in Figure 3.5b, the Δ-consecutive correlations plot shows that the level of the Spearman's rank correlation coefficient is rather low for all the values of Δ. However, selecting a correlation threshold of 0.1 gives an $\varepsilon = 0.994$, which is a distance threshold value very high. This reveals a first symptom of a noisy RP, such as the white noise in Figure 1.7a or a chaotic trend in Figure 1.7c of Chapter 1.



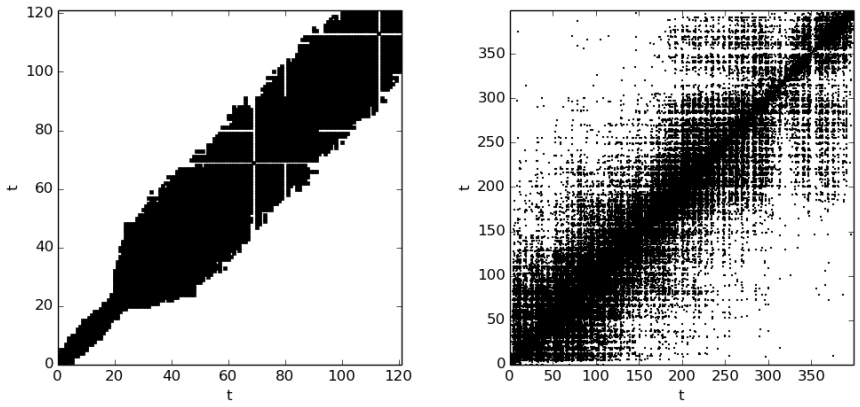
(a) CAIDA (b) FACEBOOK

Figure 3.5: Δ-consecutive correlations and distances for the CAIDA (3.5a) and Facebook (3.5b) dataset.

Thus, all the pairs of rank vectors $(\boldsymbol{\pi}_\Delta(t), \boldsymbol{\pi}_\Delta(t + \Delta))$ having a distance $d_2 < \varepsilon$ can be plotted in the RP. The results are shown in Figure 3.6a for the CAIDA dataset and in Figure 3.6b for the Facebook dataset. In the first case, the RP is totally outlined, whereas in the second case it is pretty sparse. The value of $\Delta \approx 15$ found in the distance plot of CAIDA coincides with approximately half the thickness[8] of the RP after the stabilization. Vice versa regarding Facebook, the RP showing the correlations of rank vectors

[8] Due to the symmetry.

over time gives a plot following a trend similar to the chaotic signal of Figure 1.7c. This suggests what we previously observed, namely the Facebook dataset represents a fast evolving network, in which topology continuously changes without having any correlation with the past.



(a) CAIDA (b) FACEBOOK

Figure 3.6: Recurrence Plots for the CAIDA (3.6a) and Facebook (3.6b) dataset.

Recurrence Plot is a powerful tool able to show in a deeper way what is the degree of correlation between all the single snapshots in the evolving network. The most interesting property is the possibility to analyze, for different values of ε, the corresponding trends of correlation. Indeed, while in the simple Δ-consecutive correlations plot we have just an "average" idea of the degree of correlation, the RP exactly reveals, in a very efficient way, the $t \times t$ correlations space.

3.3.2 PageRank of the expected network

The experimental results of the previous section show that computing the Pagerank of aggregate snapshots can capture correlations that are otherwise missed. On the other hand, considering $G_\Delta(t)$ amounts to estimating the expected network at time t by "averaging" over the Δ most recent snapshots. We next discuss how it is possible to define Pagerank directly on the expected network. To this purpose, note that for every $t \geqslant 0$, $\mathcal{G}(t)$ also induces a probability distribution $\mathcal{Q}(t)$, assigning to a matrix $\mathbf{Q}$ the probability that $\mathbf{Q}$ is the transition matrix of the evolving network at time t. Similarly, $\mathcal{G}(t)$ induces a distribution $\mathcal{A}(t)$, assigning to a matrix $\mathbf{A}$ the probability that it is the modified transition matrix of the evolving network at time t (see Section 3.2.2).

More in detail, denote by $\mathcal{P}(t)$ the PageRank Markov chain associated with $\mathcal{A}(t)$; namely [140], $\mathcal{P}(t) := \alpha\mathcal{A}(t) + \frac{1-\alpha}{n}\mathbf{1}\mathbf{1}^T$. By extending the argument of [49, Section 4], we can naturally define the Pagerank of the expected network.

Lemma 1. For every $t \geqslant 0$, matrix $\mathbf{E}_{\mathcal{G}}[\mathcal{P}(t)]$ is the transition matrix of an ergodic Markov chain and its main left eigenvector is:

$$\boldsymbol{\pi}^T(t) = \left(\frac{1-\alpha}{n} + \gamma\right)\mathbf{1}^T\left(\mathbf{I} - \alpha\mathbf{E}_{\mathcal{G}}[\mathcal{Q}(t)]\right)^{-1},$$

where $\gamma = \frac{\alpha}{n}\sum_i \boldsymbol{\pi}_i \mathbf{P}_{\mathcal{G}}[i \text{ is a sink at time } t]$.

Proof. The fact that $\mathbf{E}_{\mathcal{G}}[\mathcal{P}(t)]$ is the transition matrix of an ergodic Markov chain, for every $t \geqslant 0$, follows by its definition, since this is true for every realization of the process. Furthermore, from the definition of $\mathcal{P}$ we have:

$$\begin{aligned}\mathbf{E}_{\mathcal{G}}[\mathcal{P}(t)] &= \alpha\mathbf{E}_{\mathcal{G}}[\mathcal{A}(t)] + \frac{1-\alpha}{n}\mathbf{1}\mathbf{1}^T \\ &= \alpha\mathbf{E}_{\mathcal{G}}[\mathcal{Q}(t)] + \frac{\alpha}{n}\mathbf{E}_{\mathcal{G}}\left[\mathbf{1}_{\mathcal{D}(t)}\mathbf{1}^T\right] + \frac{1-\alpha}{n}\mathbf{1}\mathbf{1}^T,\end{aligned}$$

where $\mathcal{D}(t)$ denotes the set of sinks at time t and $\mathbf{1}_{\mathcal{D}(t)}$ is the corresponding indicator vector. If we impose $\boldsymbol{\pi}^T(t) = \boldsymbol{\pi}^T(t)\mathbf{E}_{\mathcal{G}}[\mathcal{P}(t)]$ we obtain:

$$\boldsymbol{\pi}^T(t) = \alpha\boldsymbol{\pi}^T(t)\mathbf{E}_{\mathcal{G}}[\mathcal{Q}(t)] + \frac{\alpha}{n}\boldsymbol{\pi}^T(t)\mathbf{E}_{\mathcal{G}}\left[\mathbf{1}_{\mathcal{D}(t)}\right]\mathbf{1}^T + \frac{1-\alpha}{n}\mathbf{1}^T.$$

The lemma then follows by observing that $\mathbf{E}_{\mathcal{G}}\left[\mathbf{1}_{\mathcal{D}(t)}\right]$ is the probability that i is a sink at time t. □

Lemma 1 suggests to use the main left eigenvector of $\mathbf{E}_{\mathcal{G}}[\mathcal{P}(t)]$ as authority score vector. For every t, this might provide a more robust authority score as it reflects the probability distribution at t and not just a single realization. Still, as we also discuss in the next paragraph, this approach may not be feasible in all cases.

Remarks. Defining Pagerank with respect to the expected network may be mathematically appealing, but the underlying distribution $\mathcal{G}(t)$ is unknown in general. To this purpose, we note that considering $G_\Delta(t)$ amounts to estimating the expected network at time t by "averaging" over the Δ most recent snapshots, so that the Pagerank vector computed over $G_\Delta(t)$ can be a reasonable approximation of $\boldsymbol{\pi}(t)$, as long as the evolving network is "almost" stationary. In particular, the former is an increasingly (with Δ) accurate estimation of the latter in the case of homogeneous evolving network distributions.

3.4 Fully decentralized Pagerank algorithms

In this section, we present algorithms that at each time t maintain an estimate of the Pagerank of the current aggregate snapshot, for a given value of the aggregation window Δ. We discuss Monte Carlo algorithms for tracking Pagerank of the aggregate network. It turns out that the proposed algorithms are amenable for a fully decentralized implementation, in which each node computes its own score by exchanging tokens with its neighbors. More specifically, the algorithms we consider in this section work in the scenario briefly defined below.

Computational setting. We assume a synchronous setting, so that computation occurs over discrete time-steps, one for each different snapshot of the evolving network $G(t)$. Each node of the network is a computing element. At each step t, every node i can only exchange information with its neighbors in the current snapshot $G(t)$.

Goal of the computation. Considered an aggregation window size Δ, for every step t, the goal is for every node i to keep an estimate of the i-th component of $G_\Delta(t)$'s Pagerank vector, up to a common multiplicative constant.

Remark. We show in Section 3.4.2 how the assumption about synchronicity of the system can be mitigated.

A basic Monte Carlo algorithm. Monte Carlo methods naturally lend themselves to maintaining approximations of the PageRank vector under continuous updates. Still, as observed in [50], how to achieve this exactly is not clear, nor are the accuracy and computational costs clear.

On the other hand, Lemma 1 suggests using the distributed algorithm proposed in [178] and derived from [49, Section 2, Algorithm 4], to track (up to normalization) the main eigenvector of $\mathbf{E}_{\mathcal{G}}[\mathcal{Q}(t)]$ under continuous updates. This is presented in Figure 3.7 as Algorithm `FDSAMPLE` [178].

`FDSAMPLE`$(i, \mathbf{A}, \alpha, r)$

Require: `node` i`, matrix` $\mathbf{A}$`, damping factor` α`, rate` r

```
1: C_i = 0
2: for every step do
3:   T = incoming() ⋃ generate(r)
4:   for every token in T do
5:     C_i = C_i + 1
6:     dies = rnd(0, 1)
7:     if dies < α AND node i is not a sink then
8:       send token to neighbor j with probability A_ij
9:     remove token from T
```

Figure 3.7: Fully decentralized version of Algorithm `FDSAMPLE`.

In a nutshell, FDSAMPLE follows the random walk described by Process 1 in Section 3.2.2, but building on Lemma 1 and [49, Section 4], it keeps track of the overall number of tokens that visit each node. More in detail, at the beginning of each step the generic node i has a set T of tokens (generated in the current step or received from other nodes at the end of the previous step). Each token causes an increase of the local counter. Furthermore, each token either dies with probability $1 - \alpha$, or it is forwarded to a randomly chosen neighbor. If the node is a sink at time t, all tokens are terminated.

3.4.1 Analysis of FDSAMPLE on homogeneous networks

In the next section, we prove that when the evolving network is homogeneous, Algorithm FDSAMPLE maintains an accurate estimation of the main left eigenvector of $\mathbf{E}_{\mathcal{G}}[\mathcal{P}]$, up to a factor that is constant over all eigenvectors' components[9]. Experimental evidence presented in Section 3.5 further supports these claims. Though assuming a homogeneous evolving network is in general unrealistic, performing this analysis is beneficial for a number of reasons: i) analyzing the behavior of the algoritm on a simplified but tractable model gives us a first grasp of the main factors affecting performance; ii) as experimental evidence in Section 3.5.3 shows, real networks (at least the ones we consider) exhibit a certain degree of stationarity, when not homogeneity; iii) assuming a homogeneous network allows us to theoretically ground changes to the basic FDSAMPLE algorithm and to quantitatively estimate their effects (e.g., on the number of circulating tokens); iv) last but not least, this idealized model provides some theoretical justification for maintaining Pagerank estimates under continuous updates using Monte Carlo sampling methods.

Preliminaries. In the rest of this section, we denote by $\mathcal{D}_{t_0,j}(i,t)$ the event that a token generated at time t_0 at vertex j visits node i at time $t \geqslant t_0$, thus causing an increase of $\mathbf{C}_i$.[10] We further denote by $\mathbf{C}_i(t)$ the value of $\mathbf{C}_i$ at time t. We denote by $\tau_{i,[a,b]}$ the overall number of visits paid to i by tokens that were released between (and including) times a and b. We write $\tau_{i,a}$ when $a = b$ and $\tau_{i,\leqslant b}$ when $a = 0$. Finally, $\boldsymbol{\tau}_{[a,b]}$ denotes the vector whose i-th component is $\tau_{i,[a,b]}$. It should be noted that, in general, $\mathbf{C}_i(t) \neq \tau_{i,\leqslant t}$, since the latter includes the visits paid after time t by tokens that were released at or before time t. Henceforth, we denote by $\mathbf{P}_{A,\mathcal{G}}[\cdot]$ a probability taken with respect to both the distribution $\mathcal{G}$ and the random choices of algorithm FDSAMPLE. The analysis proceeds as follows: i) we first show that $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\boldsymbol{\tau}_{\leqslant t}]$ is proportional to the Pagerank of the expected network at any time t (Theorem

[9] We drop t, since $\mathbf{E}_{\mathcal{G}}[\mathcal{P}]$ is constant for homogeneous evolving networks.

[10] Note that in general, the probability that a token generated by node j visits i and thus contributes to $\mathbf{C}_i$ of node i depends on the time t_0 at which the token was generated.

1); ii) we further show that at any time t, the difference between $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\mathbf{C}(t)]$ and $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\boldsymbol{\tau}_{\leqslant t}]$ is sufficiently "small" (Theorem 2), so that the former is also a good approximation of Pagerank.

The next lemma shows a useful property of homogeneous networks:

Lemma 2. For a homogeneous evolving network $\mathcal{G}$, for every t_0 we have that

$$\mathbf{P}_{A,\mathcal{G}}[\mathcal{D}_{t_0,j}(i,t)] = \mathbf{P}_{A,\mathcal{G}}[\mathcal{D}_{0,j}(i,t-t_0)]\,,$$

i.e., $\mathbf{P}_{A,\mathcal{G}}[\mathcal{D}_{t_0,j}(i,t)]$ does not depend on the time t_0 at which the token was generated.

Proof. The claim follows for the following reasons: i) homogeneity implies that the probability of observing any given subsequence of graphs is the same if the token starts at time t_0 or at time 0 and ii) the random choices of the algorithm only depend on the evolving network. □

On $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\boldsymbol{\tau}_{\leqslant t}]$ and Pagerank of the expected network. We next show that, for every i and for every t, the value $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,\leqslant t}]$ is proportional to the i-th component of the PageRank vector associated to matrix $\mathcal{A}$ (recall that $\mathcal{A} = \mathbf{E}_{\mathcal{G}}[\mathcal{A}(t)]$).

Theorem 1. If the evolving network $\mathcal{G}$ is homogeneous and r new tokens are generated at every node in each step we have:

$$\mathbf{E}_{\mathcal{G}}[\boldsymbol{\tau}_{a,b}] = r(b-a+1)\mathbf{1}^T(\mathbf{I}-\alpha\mathcal{Q})^{-1}.$$

Proof. We first note that $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\boldsymbol{\tau}_{a,b}] = \sum_{s=a}^{b}\mathbf{E}_{\mathbf{A},\mathcal{G}}[\boldsymbol{\tau}_s]$ from linearity of expectation. Furthermore, Lemma 2 implies that $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,s}] = \mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,0}]$, for every s and for every i, so that $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\boldsymbol{\tau}_{a,b}] = (b-a+1)\mathbf{E}_{\mathbf{A},\mathcal{G}}[\boldsymbol{\tau}_0]$. We therefore only need to compute $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\boldsymbol{\tau}_0]$.

Let $G = \langle G(0),\ldots,G(t)\rangle$ be a sequence of graphs (and let $Q = \langle Q(0),\ldots Q(t)\rangle$ be the associated sequence of transition matrices). Then, switching to matrix notation and assuming each node injects r tokens in every step, we have:

$$\mathbf{E}_{\mathbf{A}}[\boldsymbol{\tau}_0 \mid \mathcal{G} = G] = r\mathbf{1}^T\sum_{t=0}^{\infty}\alpha^t\prod_{k=0}^{t}Q(k),$$

where we set $Q(0) = \mathbf{I}$. Assuming $r = 1$ in the rest of the proof to simplify notation, the equality above implies:

$$\mathbf{E}_{\mathbf{A},\mathcal{G}}[\boldsymbol{\tau}_0] = \mathbf{1}^T \sum_{t=0}^{\infty} \alpha^t \mathbf{E}_{\mathcal{G}}\left[\prod_{k=0}^{t} Q(k)\right].$$

The i-th component of this vector is the expected, overall number of visits to i from tokens that are released at time 0, where expectation is taken over the random choices of $\mathcal{G}$ and of Algorithm `FDSAMPLE`.

We next prove the following:

Lemma 3. If the evolving network distribution $\mathcal{G}$ is homogeneous we have:

$$\mathbf{E}_{\mathcal{G}}\left[\prod_{k=0}^{t} \mathcal{Q}(k)\right] = \mathcal{Q}^t,$$

where $\mathcal{Q} = \mathbf{E}_{\mathcal{G}}[\mathcal{Q}(0)]$.

Proof. The proof is by induction on t. The claim is trivially true when $t = 0$. For $t > 0$, assume it holds for $t - 1$. We have:

$$\mathbf{E}_{\mathcal{G}}\left[\prod_{k=0}^{t} \mathcal{Q}(k)\right] = \mathbf{E}_{\mathcal{G}}[\mathbf{M}\mathcal{Q}(t)]\,,$$

where we set $\mathbf{M} = \prod_{k=0}^{t-1} \mathcal{Q}(k)$. We next consider the (i, j) component of $\mathbf{E}_{\mathcal{G}}\big[\prod_{k=0}^{t} \mathcal{Q}(k)\big]$. We have:

$$\begin{aligned}
\left(\mathbf{E}_{\mathcal{G}}\left[\prod_{k=0}^{t} \mathcal{Q}(k)\right]\right)_{ij} &= \mathbf{E}_{\mathcal{G}}\left[\sum_{\ell=1}^{n} \mathbf{M}_{i\ell}\mathcal{Q}_{\ell j}(t)\right] \\
&= \sum_{\ell=1}^{n} \mathbf{E}_{\mathcal{G}}[\mathbf{M}_{i\ell}\mathcal{Q}_{\ell j}(t)] \\
&= \sum_{\ell=1}^{n} \mathbf{E}_{\mathcal{G}}[\mathbf{M}_{i\ell}]\,\mathbf{E}_{\mathcal{G}}[\mathcal{Q}_{\ell j}(t)] \\
&= \sum_{\ell=1}^{n} \mathbf{E}_{\mathcal{G}}[\mathbf{M}_{i\ell}]\,\mathcal{Q}_{\ell j} = \mathcal{Q}^t,
\end{aligned}$$

where the third equality follows since $\mathbf{M} = \prod_{k=0}^{t-1} \mathcal{Q}(k)$ and $\mathcal{Q}(t)$ are independent (by definition of homogeneous evolving network), while the fifth equality follows since the induction hypothesis implies that $\mathbf{E}_{\mathcal{G}}[\mathbf{M}] = \mathcal{Q}^{t-1}$. □

Lemma 3 implies that $\sum_{t=0}^{\infty} \alpha^t \mathbf{E}_{\mathcal{G}}\left[\prod_{k=0}^{t} Q(k)\right]$ is the Neumann series of a matrix $(\alpha\mathcal{Q})$ with spectral radius strictly less than 1, so that the series converges to $(\mathbf{I} - \alpha\mathcal{Q})^{-1}$. This concludes the proof of Theorem 1. □

Note that, up to the factor $\frac{1-\alpha}{n} + \gamma$ (< 1), this is exactly the expression of the main left eigenvector of the stochastic matrix $\mathcal{Q} = \mathbf{E}_{\mathcal{G}}[\mathcal{A}(t)]$ found in Lemma 1.

Bounding the error. At each time t, for each node i, $\mathbf{C}_i(t)$ is the current estimate of i's centrality at time t. We have seen that, in general, $\mathbf{C}_i(t)$ and $\tau_{i,\leqslant t}$ may differ[11]. Fortunately, it is possible to show that $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\mathbf{C}_i(t)]$ and $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,\leqslant t}]$ differ by a quantity that is a small fraction of $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_i(t)]$ as soon as t is sufficiently large.

Theorem 2. If the evolving network $\mathcal{G}$ is homogeneous we have for every i:

$$\mathbf{E}_{\mathbf{A},\mathcal{G}}[\mathbf{C}_i(t)] \leqslant \mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,\leqslant t}]\,,$$
$$\mathbf{E}_{\mathbf{A},\mathcal{G}}[\mathbf{C}_i(t)] > \mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,\leqslant t}]\left(1 - \frac{2c\ln n + 1}{t+1} - \frac{\alpha}{(1-\alpha)n(t+1)}\right),$$

where $c = 1/\ln(1/\alpha)$.[12]

Proof. From Lemma 1 and since the network is homogeneous, we know that the Pagerank over the expected network at any time t is

$$\boldsymbol{\pi} = \left(\frac{1-\alpha}{n} + \gamma\right)\mathbf{1}^T\left(\mathbf{I} - \alpha\mathcal{Q}\right)^{-1},$$

independently of t. We next denote by q_i the i-th component of the vector $\mathbf{1}^T\left(\mathbf{I} - \alpha\mathcal{Q}\right)^{-1}$ and set $\phi = (\frac{1-\alpha}{n} + \gamma)^{-1}$. Theorem 1 then implies, for integers $a, b, a \leqslant b$:

Fact 1.

$$\mathbf{E}_{\mathbf{A},\mathcal{G}}\left[\tau_{i,[a,b]}\right] = \phi r(b - a + 1)q_i.$$

Next, observe that $\mathbf{C}_i(t) \leqslant \tau_{i,\leqslant t}$ is deterministically true from the definition of $\tau_{i,\leqslant t}$, which implies $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\mathbf{C}_i(t)] \leqslant \mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,\leqslant t}]$.

We now introduce the following random variables, exclusively for the rest of this proof: considered an (integer) interval $I = [a, b]$ and an integer $x \geqslant b$, we denote by $T^x_{i,I}$ the overall number of visits paid to node i *after* time x by tokens released during interval I. As usual, we write $T^x_{i,a}$ whenever $a = b$ and

[11] Indeed, it is possible to give toy examples in which $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\mathbf{C}_i(t)]$ and $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,\leqslant t}]$ differ substantially.

[12] c is slightly larger than 6 when $\alpha = 0.85$.

$T^x_{i,\leqslant b}$ or $T^x_{i,<b}$ whenever $a = 0$. Then, fixed a constant c ($1/\ln(1/\alpha)$ in our case) the following holds deterministically from the definitions of $\tau_{i,\leqslant t}$ and $T^x_{i,I}$:

$$\begin{aligned}\tau_{i,\leqslant t} &= \mathbf{C}_i(t) + T^t_{i,\leqslant t} = \mathbf{C}_i(t) + T^t_{i,<t-2c\ln n} + T^t_{i,[t-2c\ln n,t]}\\ &\leqslant \mathbf{C}_i(t) + T^t_{i,<t-2c\ln n} + \tau_{i,[t-2c\ln n,t]}.\end{aligned}$$

The first inequality essentially states that the difference between $\tau_{i,\leqslant t}$ and $\mathbf{C}_i(t)$ is due to the visits to i paid after time t by tokens released within time t, while the third follows since $T^t_{i,[t-2c\ln n,t]}$ refers to a subset of the visits paid to i by tokens released in the interval $[t - 2c\ln n, t]$. Taking expectations we obtain:

$$\begin{aligned}&\mathbf{E}_{\mathbf{A},\mathcal{G}}[\mathbf{C}_i(t)] \geqslant \mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,\leqslant t}] - \mathbf{E}_{\mathbf{A},\mathcal{G}}\left[T^t_{i,<t-2c\ln n}\right]\\ &-\mathbf{E}_{\mathbf{A},\mathcal{G}}\left[\tau_{i,[t-2c\ln n,t]}\right]\\ &= \phi r(t+1)q_i - \phi r(2c\ln n+1)q_i - \mathbf{E}_{\mathbf{A},\mathcal{G}}\left[T^t_{i,<t-2c\ln n}\right], \qquad (3.1)\end{aligned}$$

where the second inequality follows from Fact 1. Now, assume n tokens are released, one per node, at time 0. We are interested in the distribution of such tokens among the nodes of the network at time $\ell \geqslant 0$. More precisely, for every integer ℓ, let $\mathbf{X}(\ell)$ denote the vector whose i-th components denotes the number of tokens at node i at the end of step ℓ. We prove the following

Fact 2. If $\mathcal{G}$ is homogeneous then we have:

$$\mathbf{E}_{\mathbf{A},\mathcal{G}}\left[\mathbf{X}(\ell)^T\right] = \alpha^\ell \mathbf{1}^T \mathcal{Q}^\ell,$$

Proof. Recall that $\mathbf{E}_{\mathbf{A},\mathcal{G}}\left[\mathbf{X}(\ell)^T\right] = \{\mathbf{E}_{\mathbf{A},\mathcal{G}}[\mathbf{X}_1(\ell)], \ldots, \mathbf{E}_{\mathbf{A},\mathcal{G}}[\mathbf{X}_n(\ell)]\}$ by definition. The proof then follows by observing that the probability that the token released at node j is at node i at the end of step ℓ is exactly $\alpha^\ell \mathbf{e}_j^T \mathcal{Q}^\ell$, where $\mathbf{e}_j$ is the j-th canonical vector. The claim immediately follows. □

Next, we need the following

Lemma 4. For every $x \geqslant 1$ we have:

$$\mathbf{E}_{\mathbf{A},\mathcal{G}}\left[T^\ell_{i,0}\right] \leqslant n\alpha^\ell \mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,0}]$$

Proof. Denote by Ω the set of possible values for the vector $\mathbf{X}(\ell)$. For $\mathbf{x} \in \Omega$ we have:

$$\mathbf{E}_{\mathbf{A},\mathcal{G}}\Big[T^{\ell}_{i,0} \,|\, \mathbf{X}(\ell) = \mathbf{x}\Big] = \mathbf{x}^T \sum_{t=0}^{\infty} \alpha^t \mathbf{E}_{\mathcal{G}}\left[\prod_{k=0}^{t} Q(k)\right]$$
$$= \mathbf{x}^T \sum_{t=0}^{\infty} \alpha^t \mathcal{Q}^t.$$

As a consequence:

$$\mathbf{E}_{\mathbf{A},\mathcal{G}}\big[T^{\ell}_{i,0}\big] = \sum_{\mathbf{x}\in\Omega} \left(\mathbf{x}^T \sum_{t=0}^{\infty} \alpha^t \mathbf{E}_{\mathcal{G}}\left[\prod_{k=0}^{t} Q(k)\right]\right) \mathbf{P}_{A,\mathcal{G}}[\mathbf{X}(\ell) = \mathbf{x}]$$
$$= \sum_{\mathbf{x}\in\Omega} \mathbf{x}^T \mathbf{P}_{A,\mathcal{G}}[\mathbf{X}(\ell) = \mathbf{x}] \sum_{t=0}^{\infty} \alpha^t \mathcal{Q}^t = \mathbf{E}_{\mathbf{A},\mathcal{G}}\big[\mathbf{X}(\ell)^T\big] \sum_{t=0}^{\infty} \alpha^t \mathcal{Q}^t$$
$$= \alpha^{\ell} \mathbf{1}^T \mathcal{Q}^{\ell} \sum_{t=0}^{\infty} \alpha^t \mathcal{Q}^t,$$

where the second equality follows from exchanging summations, while the third follows by simply observing that the generic term of the sum $\sum_{\mathbf{x}\in\Omega} \mathbf{x}^T \mathbf{P}_{A,\mathcal{G}}[\mathbf{X}(\ell) = \mathbf{x}]$ is a vector whose i-th component is $\mathbf{x}_i \mathbf{P}_{A,\mathcal{G}}[\mathbf{X}(\ell) = \mathbf{x}]$ and that $\sum_{\mathbf{x}\in\Omega} \mathbf{x}_i \mathbf{P}_{A,\mathcal{G}}[\mathbf{X}(\ell) = \mathbf{x}] = \mathbf{E}_{\mathbf{A},\mathcal{G}}[\mathbf{X}_i(\ell)]$. □

Now, we continue the proof of Theorem 2. We have from Lemma 4:

$$\mathbf{E}_{\mathbf{A},\mathcal{G}}\big[T^{t}_{i,<t-2c\ln n}\big] = \sum_{s=0}^{t-2c\ln n-1} \mathbf{E}_{\mathbf{A},\mathcal{G}}\big[T^{t}_{i,s}\big]$$
$$= \sum_{s=0}^{t-2c\ln n-1} \mathbf{E}_{\mathbf{A},\mathcal{G}}\big[T^{t-s}_{i,0}\big] \leqslant \sum_{s=0}^{t-2c\ln n-1} n\alpha^{t-s} \mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,0}]$$
$$= \mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,0}] \sum_{s=2c\ln n+1}^{t} n\alpha^s$$
$$= n\alpha^{2c\ln n+1} \mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,0}] \sum_{s=0}^{t-2c\ln n-1} \alpha^s < \frac{n\alpha^{2c\ln n+1}}{1-\alpha} \mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,0}]$$
$$= \frac{n\alpha^{2c\ln n+1}}{1-\alpha} \phi r q_i.$$

Recalling Equation (3.1), we finally have:

$$\mathbf{E}_{\mathbf{A},\mathcal{G}}[\mathbf{C}_i(t)] \geqslant \phi r(t+1)q_i - \phi r(2c\ln n+1)q_i - \frac{n\alpha^{2c\ln n+1}}{1-\alpha}\phi r q_i.$$

Recalling that $\mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,\leqslant t}] = \phi r(t+1)q_i$ we have:

$$\mathbf{E}_{\mathbf{A},\mathcal{G}}[\mathbf{C}_i(t)] \geqslant \mathbf{E}_{\mathbf{A},\mathcal{G}}[\tau_{i,\leqslant t}]\left(1 - \frac{2c\ln n+1}{t+1} - \frac{n\alpha^{2c\ln n+1}}{(1-\alpha)(t+1)}\right).$$

Recalling that $c = 1/\ln(1/\alpha)$ yields the thesis. □

3.4.2 Discussion

In this section, we discuss some properties of `FDSAMPLE` and highlight issues that at least in part motivate the modified heuristic proposed and discussed in Section 3.4.3.

Accuracy and convergence. The notion of convergence should be handled with care when referred to an algorithm that performs a continuous computation over a time evolving network. In fact, the notion might be ill-posed, e.g., when the underlying network possesses strong non-stationary properties. On the other hand, Theorem 1 shows that, in the case of homogeneous evolving networks, the expected vector computed by Algorithm `FDSAMPLE` is (up to normalization) exactly PageRank computed on the matrix $\mathcal{A}$. Techniques similar to those presented in [49, 178] allow to show that, in the case of homogeneous networks, after a logarithmic number of iterations (with respect to n and the inverse of the minimum Pagerank value), the vector computed by `FDSAMPLE` stabilizes to a value that, up to normalization, is close to PageRank computed over the stochastic matrix $\mathcal{A} = \mathbf{E}_{\mathcal{G}}[\mathcal{A}(t)]$.

Resource efficiency. We consider computational and communication costs. As for computational costs, the cost of Algorithm `FDSAMPLE` is essentially measured, within a given iteration t and for a given node v, by the number of tokens v processes during the i-th iteration. As a result, we can approximately measure the expected total work done during the t-th iteration by any of the two algorithms by the expected number of tokens that are in the network in the same iteration. It is very easy to prove that this is $O(rn)$ (with high probability). This is shown in the following lemma.

Lemma 5. Let $T(t)$ denote the overall number of tokens at the end of iteration t of Algorithm `FDSAMPLE`. Then, regardless of the underlying evolving network $\mathcal{G}$, we have:

$$\mathbf{E_A}[T(t)] \leqslant \frac{rn}{1-\alpha}.$$

Furthermore, $T(t)$ is within $1+\epsilon$ from its expectation with high probability.

Proof. Consider a generic token released at time $t_0 < t$. Define by $\mathcal{L}_s$ the event that the token still lives at time s, with $s = t_0 + 1, \ldots, t$. Then we have:

$$\mathbf{P}_A[\mathcal{L}_s] = \alpha^{\Delta},$$

where $\Delta = t - t_0 + 1$ and where the probability above does not depend on the evolving network. Hence, the expected number of tokens released at time t_0 that are still alive at time t is $rn\alpha^{\Delta}$. Summing over all time instants we obtain:

$$\mathbf{E}[T(t)] = rn \sum_{\Delta=0}^{t} \alpha^{\Delta} < rn \sum_{\Delta=0}^{\infty} \alpha^{\Delta} = \frac{rn}{1-\alpha}.$$

Finally, the probabilities that tokens released within the t-th iteration are still alive at the end of t are clearly independent. As a consequence, simple application of a standard Chernoff bound allows to conclude that $T(t) < (1+\epsilon)\frac{rn}{1-\alpha}$ with high probability. □

Remark. Note that this lemma does not depend on the distribution $\mathcal{G}$ of the evolving network.

Link congestion. So far, we measured communication efficiency (and computational cost) by bounding the total number of tokens that are alive in the network at any given step t. In fact, congestion over networks' links can also be an issue, as pointed out in [178], where some possible optimization is also discussed. For example, it is possible to transmit fewer bits per step along each link, by transmitting the total number of tokens that should reach a neighbor in a given step, instead of the token themselves [178].

Memory. The amount of memory necessary at every node is essentially the one necessary for token bookkeeping and for the local counter. An issue with algorithm `FDSAMPLE` (shared by its variants) is that counters grow unboundedly

over time, which means they are eventually going to overflow/reset. This issue is addressed by the modified version of the algorithm presented in Section 3.4.3.

Asynchronous settings. We have so far assumed that i) computation proceeds along a series of parallel and synchronous steps; ii) computation at the nodes of the network starts at some time 0 common to all nodes and proceeds continuously thereafter. This is in general not the case and we next discuss some potential consequences. The scenario does not change if local clocks have same periods but different phases. This case is not really an issue and essentially all results we have shown so far carry over, though with some more technicalities. In fact, we were implicitly assuming this slightly more general framework as we presented `IDEALSAMPLE` in the introduction. On the other hand, things can change substantially when i) does not hold because nodes have different clock periods. In this case, nodes inject tokens at different rates. In the special case in which rates are constant (albeit different) and the network is homogeneous, the techniques we consider can be extended to show the intuitive fact that the score vector `FDSAMPLE` tends to a personalized PageRank computed over the expected transition matrix. In general, these scores tend to be biased towards the personalized pagerank vectors of nodes that generate tokens more frequently. A similar effect arises if ii) does not hold, in the sense that nodes start their local computations at different times and/or present inactivity periods. In this case, the consequent bias reflects nodes' activity in the network and may be informative after all.

We finally remark that the modified algorithm presented in the next section partially mitigates issues of synchronicity, by giving more importance to recent snapshots of the network.

3.4.3 Addressing non homogeneous networks

In this section, we consider cases in which the evolving network is not homogeneous (possibly, not even stationary), or is only so in part. This is likely to occur pretty frequently in practice. The Facebook dataset introduced in Section 3.2 is an example. In fact, $\mathcal{G}$ obviously depends on time in many important cases. For example, the evolving network of phone calls (where we have a link associated to the t-th time interval if a call occurred between its endpoints during t) is likely to be affected by the interplay between geography and local time. Also, the evolving network is unlikely to be memoryless.

In particular, two phenomena can negatively affect accuracy in these cases. First, some networks are very dynamic (like the Facebook dataset) and extremely sparse in most snapshots. This was, for example, the case with a small, real mobile dataset [87]. Under such circumstances, `FDSAMPLE` was performing poorly, the main reason being that the vast majority of tokens would

traverse one link at most, or no link at all, so that the estimation of the Pagerank on the aggregate network is close to uniform[13]. Second, non-stationarity entails that it may be reasonable to require that Pagerank scores should at least reflect the recent history of the network. Consider for example the case of a network in which some node i has extremely high in-degree over each time-step until some time $\hat{t}$, after which it is no longer reachable. Even in this scenario, i's relative authority would slowly decrease, as more and more tokens visit other vertices. Still, this process might be too slow to be useful in practice.

To address these issues, we present below algorithm `PATIENTSAMPLE` in Figure 3.8. This algorithm i) randomly delays tokens, whenever the nodes that are in the current step are sinks and ii) it uses an exponential weighted moving average to demote older counter updates.

`PATIENTSAMPLE`$(i, \mathbf{A}, \alpha, r, \gamma)$

Require: `node` i`, matrix` $\mathbf{A}$`, damping factor` α`, rate` r`, patience parameter` γ

```
1:  C_i = 0; T = ∅
2:  for every step t do
3:    S = incoming() ⋃ generate(r)
4:    C_i = βC_i + |S|
5:    T = T ⋃ S
6:    if node i is a sink then
7:      for every token in T do
8:        if rnd(0, 1) > γ then
9:          remove token from T
10:   else
11:     for every token in T do
12:       dies = rnd(0, 1)
13:       if dies < α then
14:         send token to neighbor j with probability A_ij(t)
15:       remove token from T
```

Figure 3.8: "Patient" version of Algorithm `FDSAMPLE`.

The algorithm follows the very same lines as `FDSAMPLE`. The main difference is that, if node i is a sink in the current step, all tokens in the node are given a chance survive to the next step with probability γ, for some $\gamma \in (0, 1)$. Also, counters are updated, so that tokens received (or generated) in the current

[13]Note that these effects arise because the network is strongly non-homogeneous. Results on synthetic, homogeneous networks with the same degree of sparsity were very accurate, similar to those presented in Section 3.5.1.

time-step count 1, while the overall contribution of token visits received in the past is reduced by a factor $\beta < 1$ in each step.

It is worth noting that this algorithm reduces to `FDSAMPLE` when $\beta = 1$ and $\gamma = 0$.

Homogeneous networks. Algorithm `PATIENTSAMPLE` is consistent with Algorithm `FDSAMPLE`, in the sense that it estimates the same Pagerank scores in important cases. In particular, we remark that the technical arguments presented in Section 3.4 can be extended to show that Algorithm `PATIENTSAMPLE` computes a vector that, in expectation and up to normalization, is an excellent approximation of PageRank in the case of homogeneous networks. In particular, Theorem 1 still applies, with minor modifications.

Counter space. As we mentioned earlier, Algorithm `PATIENTSAMPLE` ensures that counters are extremely unlikely to overflow. In particular, recall from Lemma 5, that with high probability there are at most $\frac{rn}{1-\alpha}$ tokens in the network at every step. We present below the argument for $\gamma = 0$, for the sake of simplicity.

In the worst possible case, a node i receives n tokens at every step. After t steps, the overall value of $\mathbf{C}_i$ is thus upper bounded by $\frac{rn}{1-\alpha}\sum_{s=0}^{t}\beta^s = \frac{1-\beta^{t+1}}{1-\beta}\frac{rn}{1-\alpha}$ with high probability.[14]

3.5 Experimental analysis

The main goal of this section is to complement the results presented in the previous sections by putting the theoretical findings of Section 3.4 to the test and by assessing the suitability of Monte Carlo methods for PageRank computation under continuous updates.

More in detail, for the homogeneous case, we experimentally reproduce the conditions under which our theoretical analysis predicts that, over time, the centrality vector computed by `FDSAMPLE` converges to the PageRank of the expected transition matrix of the evolving network. In particular, we are interested in experimentally quantifying the speed of convergence and the efficiency in resource usage, in particular as regards the overall amount of tokens in the network.

Furthermore, we test the performance of Monte Carlo methods in the case of real evolving networks. As also discussed in Section 3.3, assessing the performance of a heuristic presents some problems in this case, since it

[14] Note that a simple Chernoff bound application allows to conclude that the number of tokens in the network at any time t is $O\left(\frac{rn}{1-\alpha}\right)$ with probability exponentially small in n.

is not obvious what would be a meaningful benchmark for comparison. So, we tested the performance of `FDSAMPLE` and `PATIENTSAMPLE` by considering, at any time t, a suitable centrality score defined over the window of the last Δ snapshots (see further). All the code written to carry out this analysis is available at [2].

Performance measures. Let $\boldsymbol{\pi}$ be the benchmark Pagerank vector and let $\mathbf{p}$ be the vector computed by our algorithm. We define the following measures of accuracy. For a component i, the *absolute error* is $a_i = |\boldsymbol{\pi}_i - p_i|$ and the *relative error* is $e_i = \frac{a_i}{\boldsymbol{\pi}_i}$. The *average relative error* is defined as $E_1 = \frac{1}{n}\sum_{i=1}^{n} e_i$, while the *maximum relative error* is $E_\infty = \max_{i\in\{1,\ldots,n\}} e_i$. We further define the *1-norm of error*, namely, $L_1 = \sum_{i=1}^{n} a_i$, and the *infinite norm of error*, i.e., $L_\infty = \max_{i\in\{1,\ldots,n\}} a_i$. Another important measure of accuracy is *precision*. In particular, we are interested in $p@k$, namely, the *precision@k*, i.e., the fraction of the top k nodes (ordered according to non increasing $\boldsymbol{\pi}_i$'s) that are also among the top k when nodes are ordered according to non increasing p_i's. A measure that provides a quick glimpse of the computational and communication overhead, as remarked in Section 3.4.2, is the *token number* $nt(i)$; i.e., the overall number of tokens in the system at the end of the i-th iteration.

3.5.1 Evolving networks datasets

In this section we briefly describe the evolving networks we used to evaluate the performance of our algorithms.

Synthetic evolving networks. We synthetically create a homogeneous evolving network by considering the edge set of the Arxiv High Energy Physics paper citation network [183, 144]. Specifically, for probability $p = 0.05, 0.1, 0.2, 0.5, 0.8$, we define $\mathcal{G}_p(t)$ as the distribution that assigns probability p of appearance to each edge of the network. The resulting graph has 421578 edges and 34546 nodes.

Evolving networks from real datasets. We considered the CAIDA[182, 144] and Facebook[136, 194] datasets. The first one consists of $N = 122$ graphs describing links between autonomous systems in the period from January 2004 to November 2007. The total number of nodes is 31379, while the edges are 6096640. The second one is the directed network describing the sequence of wall posts by a sample of Facebook users. The total number of nodes is 46952 while the arcs are 876993. Nevertheless, since the growth rate (i.e., edges per year) follows an exponential trend, mainly in the last two years, we decided to use a restricted portion of the dataset in which the number of nodes is more

stable. Specifically, we considered a period from February 2007 to March 2008, for a total of 400 days, 22267 nodes and 277449 edges.

For the CAIDA dataset, each snapshot of the evolving network contains a list of edges being present in the corresponding month, while for the Facebook dataset each snapshot corresponds to a single day and a link (a, b) exists in a snapshot if a posted at least once on b's wall on the corresponding day.

3.5.2 Experiments on synthetic networks

Experimental setting. In this case, the vector $\mathbf{p}$ is obtained running `FDSAMPLE`. We use damping factor $\alpha = .85$ for both algorithms. All the metrics are computed over 5 independent runs on homogeneous evolving networks obtained with $p = 0.05, 0.1, 0.2, 0.5, 0.8$.

Results. In this paragraph we discuss results on $\mathcal{G}$ generated with probability $p = 0.1$. Similar results holds also for the other values of $p > 0.1$; in particular the precision slightly improves. The experimental results shown in Figure 3.9 confirm the theoretical ones in case of stationary evolving networks, namely the expected score vector computed by `FDSAMPLE` converges to PageRank. Indeed, both the L_1-norm and the average relative error E_1 are about 10% after just 50 iterations and the corresponding max relative error E_∞ is less than 1%. Remarkably, the precision computed over the first k elements in the PageRank (see Figure 3.12 (a,b,c)) converges to 1 (i.e., best accuracy) after few iterations irrespectively from the value of k. Finally, Figure 3.10 shows that the number of tokens remains bounded as expected from the theoretical results.

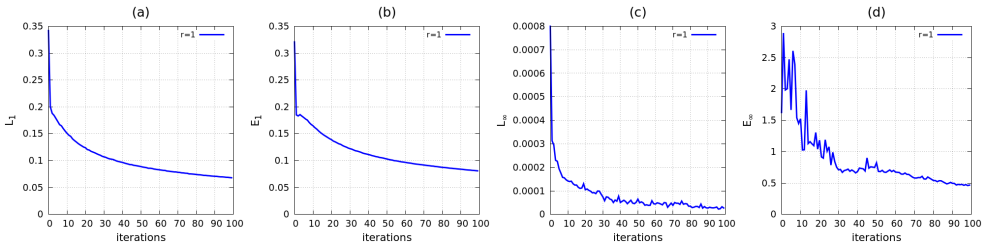


Figure 3.9: L_1-norm (a) and average relative error E_1 (b) and L_∞-norm (c) and max relative error E_∞ (d). The relative error is small (<10%) after few iterations (50). The token rate is $r = 1$ and the probability is $p = 0.1$.

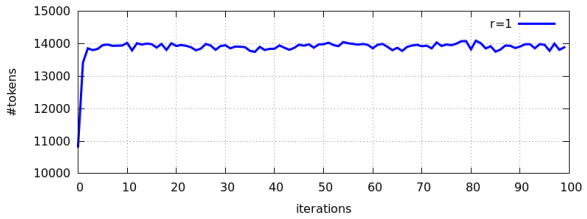


Figure 3.10: The number of tokens in the system is limited and converges to a value that is considerably smaller than the bound given by Lemma 5. The token rate is $r = 1$.

3.5.3 Experiments on real evolving networks

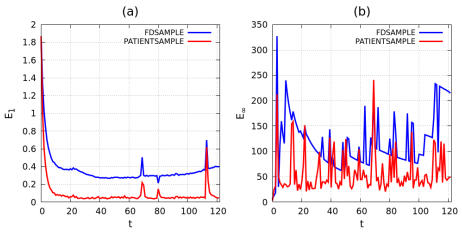
Experimental setting. In our experiment we considered several aggregate networks. In particular for each of the CAIDA and Facebook networks, we derived aggregate networks with aggregation window sizes $\Delta = 1, 5, 10, 20$. The PageRank $\boldsymbol{\pi}$ was computed for each aggregate window with damping factor $\alpha = .85$. Metrics have been computed for both `FDSAMPLE` with $\alpha = .85$ and `PATIENTSAMPLE` with $\alpha = .85$, $\gamma = \frac{\Delta}{\Delta+1}$ and $\beta = e^{-(\frac{\ln(10)}{\Delta})}$.[15] In what follows, we show the results for $\Delta = 1$ in the case of CAIDA and $\Delta = 10$ in the case of Facebook. Similar behaviors are observed for other values of Δ.

Results. Regarding CAIDA, Figure 3.11a shows that the average relative error E_1 converges very quickly, while, in the case of Facebook (see Figure 3.11b), E_1 is pretty low since the beginning. This dissimilarity is due to the fact that the Facebook dataset has a lower density than the CAIDA one, consequently most nodes in Facebook have rank values close to $\frac{1}{n}$ for both Pagerank and `FDSAMPLE` (or `PATIENTSAMPLE`) algorithms. Vice versa, CAIDA is pretty dense, therefore it needs time to converge to a lower error. Furthermore, there is a significant difference between `FDSAMPLE` and `PATIENTSAMPLE`; not only does `PATIENTSAMPLE` provides a lower E_1, but also a more stable behavior. Indeed, in both datasets, the E_1 of `FDSAMPLE` shows a slightly increase as the iterations increase. As for E_∞, `PATIENTSAMPLE` outperforms `FDSAMPLE` in the CAIDA dataset, even if the values for both the algorithms are relatively high. Contrary, E_∞ issued by `PATIENTSAMPLE` is higher in the case of the Facebook dataset. This can be explained by the fact that some nodes having a lot of incoming edges, affect other nodes' score by forwarding all their tokens before they are able to "unload" most of them. Indeed, if we set $\gamma = 0$, the probabil-

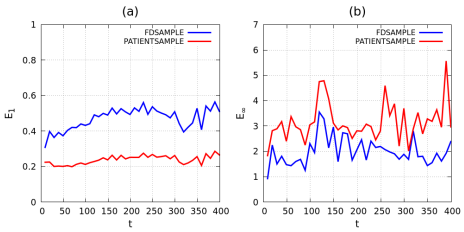
[15] The value of γ for `PATIENTSAMPLE` corresponds to each token having an expected number of Δ of chances before dying at some sink. The value of β corresponds to demoting the contributions of by token visits performed Δ or more time-steps earlier by a factor at least $1/10$.

ity that a sink can delete a token is 100%, therefore the E_∞ steeply decreases to 1. Nevertheless, the E_1 increases because sinks are allowed to continuously delete all their tokens, so we decide to keep $\gamma > 0$.

Regarding the *precision@k*, in the case of CAIDA, it converges to 1 (i.e., best accuracy) after few iterations (see Figure 3.12 (d,e,f)), while, in the Facebook dataset, it reaches at most 60% for $k = 1000$ (Figure 3.12 (g,h,i)). The poor result achieved in the Facebook dataset is mainly due to the high network dynamics and graph sparsity. Since most of the nodes exhibit similar behaviors by appearing and disappearing over time, the rank distribution is pretty "flat", consequently part of the first "k" nodes are simply random nodes not coinciding between the two computed ranks. However, in both the datasets, `PATIENTSAMPLE` outperforms `FDSAMPLE`.



(a) [CAIDA] $\Delta = 1$ in `FDSAMPLE`, $\Delta = 1$, $\gamma = 0.5$ and $\beta = 0.091$ in `PATIENTSAMPLE`.



(b) [FACEBOOK] $\Delta = 10$ in `FDSAMPLE`, $\Delta = 10$, $\gamma = 0.91$ and $\beta = 0.787$ in `PATIENTSAMPLE`.

Figure 3.11: The average relative error E_1 (a) and max relative error E_∞ (b) for `FDSAMPLE` and `PATIENTSAMPLE` in both the datasets. L_1 and L_∞ show similar and consistent behaviors.

3.6 Concluding remarks

In this chapter we have better analyzed PageRank, one of the most popular measures of centrality already introduced in Chapter 1, and possible notions to be applied in dynamic evolving networks. So far, most work on this field has considered a standard approach of defining centrality at time t as simply PageRank computed over the t-th snapshot. While this definition appears natural, it is not clear that it allows an adequate characterization of nodes' centralities in any evolving network. We have proposed two alternative notions that are still consistent with PageRank in the static case. The first notion takes into account the expected network topology at time t, while the second one considers the average topology over a suitable window of the most recent Δ snapshots. Then, we showed that, when an evolving network satisfies additional statistical properties, the first notion corresponds to the rank

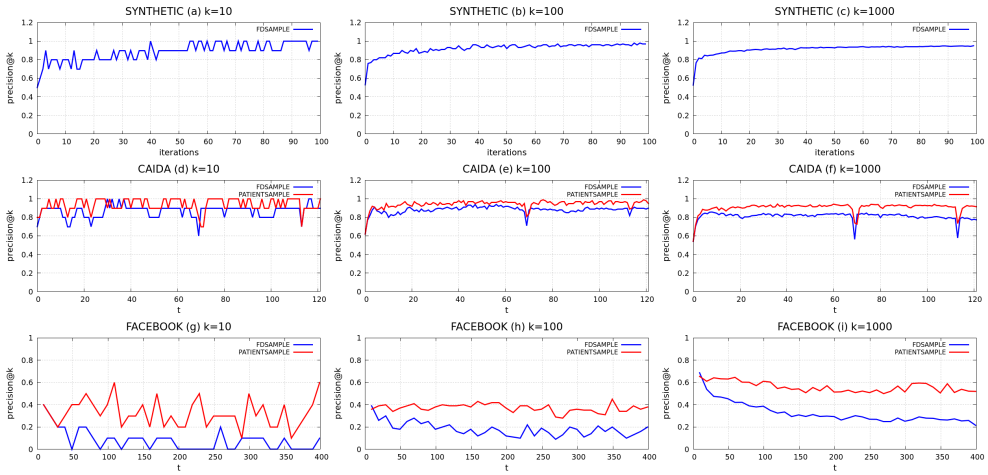


Figure 3.12: The precision computed over the first k elements in the PageRank for the synthetic network (a,b,c), the CAIDA evolving network (d,e,f) and the Facebook evolving network (g,h,i). The token rate is $r = 1$.

vector dynamically computed by a decentralized heuristic for Monte Carlo PageRank sampling proposed in a previous work. However, real-life evolving networks, such as F2F social networks discussed in the previous chapter, may exhibit significant non-homogeneous properties, so we have presented a modified heuristic which assigns each node a score that mostly reflects nodes' centrality over the evolving network's recent past and also better handles all those networks not satisfying homogeneous conditions. Finally, we have presented and discussed experimental results for the heuristics we have proposed, both on synthetic and real datasets. Results support the validity and feasibility of our approach.

In the next chapter we discuss about the wisdom of crowds effect, a phenomenon of collective intelligence directly following from the most recent evolving social networks applications.

Chapter 4

The Wisdom of Crowds effect

In recent years, researchers has started focusing their attention on the "Wisdom of Crowds" phenomenon, a well-known process of taking into account the collective opinion of a group of individuals rather than single experts to answer a certain question. The fundamental reason is that independent opinions of a crowd of people may be relatively accurate, even when most of the individuals are not expert. Such a hypothesis is derived from the fact that a collective opinion includes both the signal and the noise, therefore averaging opinions will cancel out the noise and extract the signal [117, 149].

This phenomenon has started being widespread after the success of the book [187] with the same title written by James Surowiecki and published in 2004. The author argues about one important condition, namely the aggregation of information in groups, resulting in solutions that a decision made by a crowd is often better than decisions made by any single member of the group, even if she is an expert user. An intuitive explanation for this phenomenon is that there is noise associated with each individual judgment, and taking the average over a large number of responses will go some way towards canceling the effect of this noise. The first observation on the wisdom of crowd effect was made by Sir Francis Galton as he discovered, much to his surprise, that a crowd attending a country fair was collectively able to estimate the weight of an ox with high accuracy [109, 110]. It is important to point out that this effect is statistical in nature and arises when the errors of individual estimates are such and aggregated in such a way that they "cancel out". This can for example be the case when individual errors can be large but are unbiased. This effect has been successfully used in many practical contexts, such as predictive markets [59]. Surowiecki also provides other three important conditions required to form wise crowds: diversity of opinion, independence and decentralization. While the latter is a bit harder to define formally, the first two properties roughly correspond to an experimental setting in which a set of

randomly chosen people are asked to individually answer a question, with no prior knowledge about the subject of the question and without being able to interact with other participants before providing their answers. Such an ideal setting is clearly not satisfied in everyday life, in which people's opinion are affected by a number of factors that are hard to anticipate and, above all, quantify. Two important aspects, for instance, are the social influence and the role played by expert people.

4.1 Background and related work

In a relevant paper appeared in 2011, Lorenz et al. [147] argue that "even mild social influence can undermine the wisdom of crowd effect in simple estimation tasks". This theory is supported by results of iterated experiments, conducted over small groups of individuals that were requested to answer quantitative questions and were exposed to varying degrees of information about other participants' responses. More in detail, the authors considered 12 groups of 12 individuals each. Every group participated in 6 sessions, one per question. In every session, each member of each group was asked to answer the same question 5 times, under 3 different information regimes: in particular, in sessions other than the first, participants were allowed to revise their previous estimates based on either no, or aggregate, or non-aggregate information about other participants' estimates. In a nutshell, experimental evidence reported in the paper suggests that social influence reduces group diversity under a variety of estimators, without increasing accuracy of the crowd. It should be noted that a potentially negative impact of social influence was known before this rigorous and quantitative approach, as already reported by Surowiecki [187] or Lanier [142], which focuses on Wikipedia's example to address a more general framework than the one considered here.

Similarly to previous work, King et al. [132] conducted some experiments involving a total of 429 people. All participants were invited to guess the number of sweets in a jar at one of their five voting booths equipped with a computer and a keypad. Each booth was subjected to different kinds of conditions. People voting at the booth A answered independently, without receiving any information about other participants' answers, while other booths provided participants with information about other answers. Specifically, the booth B provided the following additional information "The last person's guess was N", the booth C "A random previous guess is N", the booth D "The average of previous guesses is N" and the booth E "The best guess so far is N". At the end of their experiments authors showed that individuals with access to the previous guess, mean guess or a randomly chosen guess, tended to overestimate their answers and this undermined the wisdom, as already observed in

[147]. However, when people were provided with the current best guess, this prevented very large inaccurate guesses, resulting in convergence of guesses towards the true value. This suggests how people tend to trust and follow experts in a crowd. Another important result achieved in this work is the fact that there is a strong wisdom of the crowd effect when people had no public information as the group size increases (i.e., greater than 70 people). Nevertheless, the wisdom of crowds is not a pure statistical regularity, namely more does not always means better if all conditions argued by Surowiecki [187] are not respected.

In 2012 Mavrodiev et al. [151] recruited 144 students at ETH Zurich and studied how all the subjects, split in 12 groups of 12 people, answered a set of 6 quantitative questions regarding geographical facts and crime statistics. The participants had to answer all the questions over 5 separated rounds. Similarly to the Lorenz's experiments, different information conditions were tested regarding the information that participants got from answers of others in the previous time periods: "no information", "aggregated information" (i.e., the arithmetic average of everyone else's answers in the previous round) and "full information" (i.e., all opinions from all previous rounds). The authors also motivated students to do their best offering a reward at each round. In their analysis they measures the "collective error", defined as the squared deviation of the average opinion in the round t from the true value, the "group diversity", namely the variance of the opinion distribution, and the wisdom of crowds indicator, which measures how much deviation from the most central estimate is needed to encompass the true value. On the basis of those indicators, Mavrodiev et al. found that the social influence had a negative effect on the wisdom of crowds, thus confirming Lorenz's results.

Other papers have appeared in recent years studying and covering several topics related to the wisdom of crowds phenomenon. In [206] Yi et al. investigate whether the effect can also be observed for combinatorial problems (e.g., the planar Euclidean traveling salesperson problem), where the answer requires the combination of multiple pieces of information. Inspired by the work [96] in the financial domain, Hsieh et al. [119] examine the wisdom of the crowd effect in the domain of news recommendation by conducting experiments on Twitter. They conclude that they could not identify an expert group whose news recommendation performance was consistently better than that of the crowd. Afterwards, they study a mechanism to further improve the wisdom of the crowd performance giving more importance to experts when there is sufficient evidence and reducing noise in the crowd by removing "overly talkative" users. This task is also performed by [70] where Budescu and Chen seek to improve the quality of the aggregation by eliminating poorly performing individuals from the crowd. In doing so, they then use positive contributors to build a weighted model for aggregating forecasts and show how

this model outperforms the unweighted version. This process is equivalent to directly delete outliers, but in some cases determining if an outlying opinion is bad data is not possible. Indeed, outliers may be due to random variation or may indicate something scientifically interesting. There has been much debate in the literature regarding what to do with outliers; several papers tend to explain and summarize when and whether they should be removed or not [163, 97]. We will deal with this problem in Section 4.3.1. Studies by Harris [115] have examined how a crowd of individuals was able to judge and rank images and textual documents against their own perception of the estimate for a consensus decision. Contrary to previous work, the author shows that the group asked to make ranking decisions based on their estimate of consensus significantly outperforms the group making decisions based on their own opinions when judging relevance for a set of documents and images. This is a signal that consensus should be reconsidered as a good effect in group decision-making for specific contexts. Finally, a paper appeared in 2014 [176] examines the effect of a social network on prediction markets using a controlled laboratory experiment. This work is one of the first attempts to correlate the dynamics of a social network to the wisdom of crowds effect. The authors try to identify possible relationships between the social network and the performance of participants. Contrary to previous work, their study shows that a social-network-embedded prediction market outperforms a prediction market without social interactions in terms of prediction accuracy.

Our contribution

Most previous experimental work has focused on highlighting the impact of social influence on the wisdom of crowd effect, by considering predefined or supervised social network structures. In our research we are interested in discovering if any particular aspect of a spontaneous social network can implicitly affect the crowds in taking a decision or answering a question. For that purpose, we have deployed several experiments to study how people interact and change their opinions. As deeply described in Chapter 2, we considered a more natural setting, not a lab and no rewards that force people to work better. We just used active RFID tags capturing communication patterns during regular discussions of the corresponding participants. Though limited in size, the results of our experiments are all consistent. In particular, a first result is that, at least in the scenario under consideration, the social network plays a positive role in aggregating information and the reduction of diversity in this case seems to reduce the effect of poor initial aggregate estimation of the correct value. This result is consistent over most of the questions considered in our experiments and in line with the conclusions drawn in previous work, such as [151], which uses a model and simulation based approach, and [132]

highlighting how guesses converge towards the true value when people were provided with the current best guess. Indeed, relaxing constraints in communication and leaving people the freedom of forming their own network entail participants to talk with trusted neighbors, or at least to follow individuals they believe are expert. In the second part of this chapter we evaluate several models of opinion formation focusing the attention on how well they perform in estimating the final users' beliefs compared to the true ones. This is a critical point since understanding how beliefs and behaviors evolve over time allows to study and analyze social characteristics of several kinds of phenomena and, eventually, predict future behaviors.

4.2 The experiments in real-world scenarios

The experiments we deployed aims to observe and analyze how the wisdom of a crowd changes after a social interaction in real-world scenarios. In other words, the main goal is to discover if there is any correlation between the network dynamics and the accuracy of the answers given by participants. To do that, we designed a process to follow for each experiment.

4.2.1 The experiment process

The study in which people were asked to participate was totally anonymous and it was only designed to investigate human attitudes and abilities in performing certain tasks. We like naming this kind of experiments "social game" because participants cooperate to improve their answers as much as possible and, as the game ends, they discover if their answers are right or not. The game is composed by two rounds (or phases) and, as soon as it starts, each participant receives a RFID tag to wear and a questionnaire including four questions to be answered later.

Round 1. Every participant has exactly "x" minutes to think on possible answers, but without interacting with other people. After "x" minutes have elapsed, they will receive a sheet of paper to provide their answers to the proposed questions. They have to answer all four questions and return the sheet we gave them; at this point, the second round can start.

Round 2. During the second round, participants can interact with other attendees comparing their own answers. They are completely free to share their ideas, talk to others, trying to improve their answers using other participants' expertise. After "y" minutes they will be asked to answer again the same set of questions as happened in the first round. They are completely

free to change their previous answers if they believe they were able to improve them after the social interaction. None of the questions contained confidential requests or personal information, but they regarded totally general facts and curiosities, just according to the wisdom of crowds phenomenon.

4.2.2 Questions categories

Three classes of questions were basically proposed in our experiments:

1. *innate and learned abilities*,
2. *knowledge and reasoning*,
3. *prediction ability*.

Giving participants questions of different types allows us to observe which kind of questions works better in decision-making environments and which ones are less appropriate for the wisdom of crowds phenomenon. Essentially, in our research context, innate and learned abilities include all those questions where participants are asked to observe and answer about a counting problem (e.g., number of beans in a jar). Knowledge and reasoning require to think about a well-know and existent fact (e.g., the average population of a country over a precise period). In the end, prediction includes problems where participants try to guess a future fact.

4.2.3 The WSDM conference

The first experiment was deployed at the Antonianum Auditorium [1] in Rome during the WSDM Conference [33] taken place on February 2013. The experiment lasted around 1.5 hour, where 50 minutes were allocated for the social interaction thus collecting data from several rooms, the corridor and the lunch area. We recruited 94 volunteers between the attendees of the conference, but only 69 properly participated to our experiment. As already explained in Section 4.2.1, we first gave participants a RFID tag to be worn and a questionnaire to be answered without interacting with other individuals, then, after the conclusion of this first round, we allowed people to interact with each other and, in the end, answer again the same set of questions.

The four questions participants had to answer were the following:

1. *What was the total value in euros of all the coins thrown at the Trevi fountain in* 2011*?*
2. *What is the total length (in meters) of the corridor of the Auditorium Antonianum?*
3. *What is the average number of journal papers among the WSDM 2013 participants according to DBLP?*
4. *What was the number of Internet users in New Zealand by the end of* 2011*?*

The corresponding correct answers were: *1.* 951000*, 2.* 52.59*, 3.* 10.35*, 4.* 3796000.

In this experiment we selected questions all belonging to the *knowledge and reasoning* class. The main reason in this sort of selection lies in the fact that, as first attempt of such deployments, we principally aimed to explore the effectiveness of the participation in the experiment. Therefore, the easiest way to do that was proposing participants questions able to stimulate their curiosity as much as possible. Since all attendees were researchers from every part of the world we proposed some questions about real facts related to places in Rome.

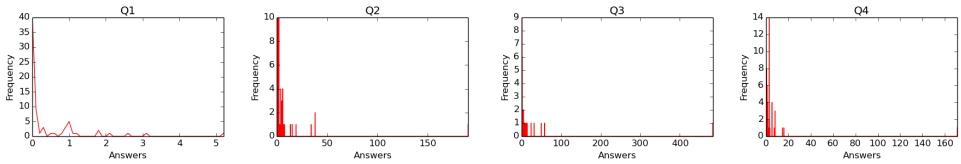


Figure 4.1: [WSDM] Answers distribution in the first round

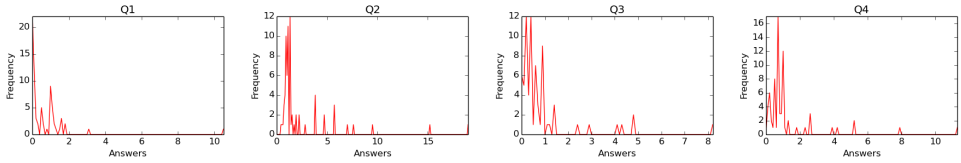


Figure 4.2: [WSDM] Answers distribution in the second round

Figures 4.1 and 4.2 show the answers distributions regarding the four questions in the first and the second round, respectively. The x-axis depicts the answers' values, while the y-axis the frequency. All the answers are normalized by the ground truth. This means that the right answer is represented by value 1 and all the answers given by participants should ideally be near to 1 as much as possible. Although in this experiment most values are pretty far from the ground truth, it can be noticed that in the second round (see Figure 4.2) the scale of values (x-axis) is rather lowered than the first round and the corresponding values are more centered on 1 in all the four questions. As we will better see in Section 4.3, this coincides with a higher accuracy after the social interaction.

4.2.4 Students at DIAG (first act), aka DIAG1

A few days after the WSDM Conference, we deployed another social experiment in our department of Computer Engineering. This time we involved 37 students from the master's degree. As already done in the previous experi-

ment, we gave each participant a RFID tag to be worn. The social interaction of the experiment lasted 20 minutes, in which students proved to be very active. The modalities of the experiment were the same, where we gave participants the following four questions to be answered:

1. *What was the total value in euros of all the coins thrown into the Trevi fountain in* 2011*?*
2. *How many beans are in the package on the table?*
3. *What is the distance in meters between the two security staircases in the department's yard?*
4. *What is the price of the smart tag you are wearing?*

The corresponding correct answers were: *1.* 951000, *2.* 792, *3.* 71.94, *4.* 21.01.

In this case we selected a question belonging to *innate and learned abilities* asking participants to guess the number of beans inside a package (number 2). Regarding the other three questions, they all belong to the *knowledge and reasoning* class. As first question we "recycled" the one about the total value in euros thrown into the Trevi fountain so as to compare the results with the WSDM experiment. Then, we selected other two questions about some facts related to reality of that moment: we asked to guess the distance between the two security staircases in the department's yard and to guess the price of the tag they were wearing.

Participation was very good and most of the students tried to speak with as many other participants as possible, so forming a very dense social graph (see Figure 2.12a in Chapter 2). No student in this experiment remained isolated, indeed all participants talked with at least other two individuals. The most active student talked with other 22 individuals.

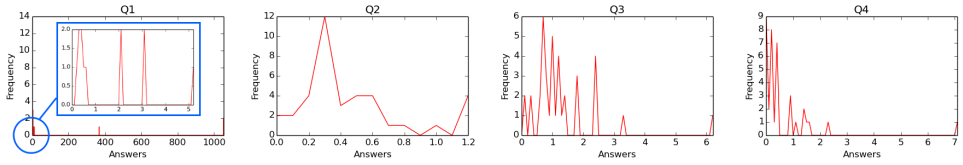


Figure 4.3: [DIAG1] Answers distribution in the first round

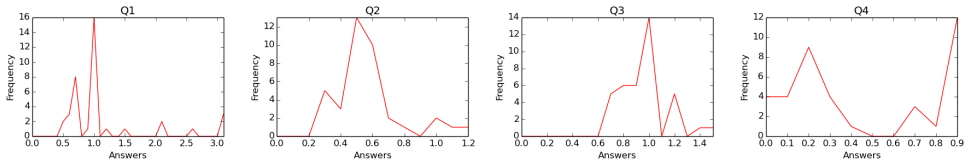


Figure 4.4: [DIAG1] Answers distribution in the second round

In Figures 4.3 and 4.4 we plotted the answers distributions in both the two rounds. As for the first question (Q1) in the first round, users distributed their answers on a very scattered scale. 8 out of 37 users answered values between 0 and 5. This means that a user giving a value of 5 makes an error of 500%. Other users answered even higher values. A cluster of individuals gave answers near to 400, while another group answered values greater than 1000. This proves to be a distribution affected by spurious data. We tackle this issue in Section 4.3.1. On the contrary, the distribution of answers related to the same question turned out to be more cohesive in the second round. 16 users answered values near to the ground truth and the farthest was just 3. With regard to the other three questions, the scale of values is less scattered with respect to the first question. However, answers given after the social interaction are always better in terms of focus on the ground truth. For instance, observing the question 3, 14 users gave an answer near to 1 in the second round. Vice versa, only one user answered a value near to 1 before the social interaction, while most of the other users answered values in the range 0 and 2.5.

4.2.5 Country fair in Priverno

On May 11, 2014 we arranged a third experiment in the park of San Martino (Priverno, Italy) during a country fair. We recruited 60 volunteers giving each of them a RFID tag to wear. Then, we delimited a large monitored area of around 15 × 15 meters in which all participants were free to move. The total experiment lasted less than 30 minutes, where around 10 minutes were assigned for the social interaction. As opposed to the WSDM experiment, we decided to heavily decrease the time of the experiment since in the previous deployment we noticed that already after 10-15 minutes some attendees wanted to answer the questionnaire for the final part of the experiment, while others tended to talk about other stuff after a certain amount of time. In order to reduce signal noise as much as possible, we set a period of around 10 minutes, a reasonable time to talk with some other participants in a social network composed by 60 individuals.

The four questions we proposed in such an experiment were the following:

1. *What was the average female population of Italy over the years* 1960-1970*?*
2. *How many meters long is the main side of the Castle of San Martino?*
3. *What was the temperature (expressed in Celsius) this morning, at 9.00 am, in Priverno?*
4. *How many dots are contained in the following figure?*

The corresponding correct answers were: *1.* 27.65 *M, 2.* 40*, 3.* 16*, 4.* 600.

Similarly to the DIAG1 experiment, we selected a question belonging to the first class of abilities (*innate and learned abilities*) asking participants to

guess the number of dots contained in a figure (number 4). Regarding the other three questions we chose questions belonging to real facts and evidences, but hardly searchable on Google.

Even in this case, participants were pretty amused by our experiment and their participation was very good.

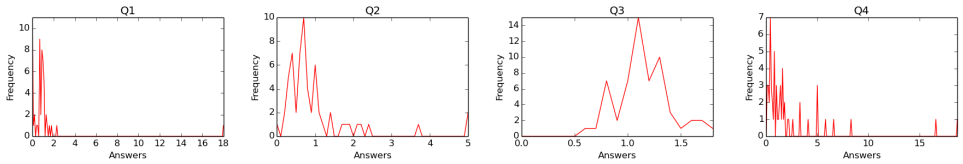


Figure 4.5: [Priverno] Answers distribution in the first round

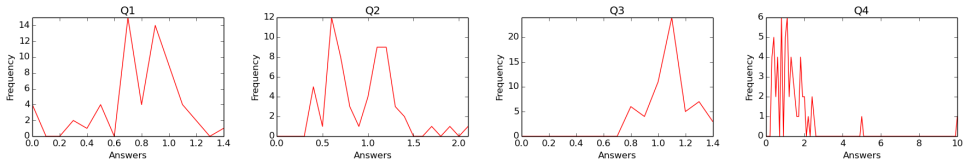


Figure 4.6: [Priverno] Answers distribution in the second round

Figures 4.5 and 4.6 depict the two distributions of the answers given by participants in the first and second round. Excepting the case of the third question (Q3), where the distributions are quite similar in both the two rounds, in the other three questions the answers given in the second round are more dense and centered on the ground truth. Specifically Q1 has most of their values spread between 0 and 2, and the farthest one is around 18. Vice versa, the values in the second round are pretty condensed between 0.7 and 1. The answers distributions of Q2 are quite similar in both the rounds if we limit the range of observation from 0 to 2. However, some answers in the first round are beyond that range. A similar behavior is observed in the last question, as well.

4.2.6 Students at DIAG (second act), aka DIAG2

The last real-world experiment was deployed at our department on May 21, 2014 where 25 students in Computer Engineering were recruited. We gave each participant a tag to wear and four questions to be answered in two separated rounds. The experiment took place in a large square behind the department, where usually students relax and eat. The time allocated for the social interaction was around 10 minutes.

The four questions we proposed in this experiment were the following:

1. *How many kilometers long is the part of the Tiber river (Tevere) that is inside the Grande Raccordo Anulare (GRA)?*
2. *How many steps is the main staircase of the department building, from the elevator level to the second floor?*
3. *Make your guess: what percentage of you will correctly answer question* 4 *with an error* $< 30\%$*?*
4. *How many dots are contained in the following figure?*

The corresponding correct answers were: *1.* 36.5, *2.* 69, *3.* 100, *4.* 450.

In this occasion we decided to exploit all the three classes of questions: the first and the second question belong to the *knowledge and reasoning* category, the third to *prediction ability*, while the fourth to *innate and learned abilities*.

What we have observed in this case was a great unified participation. Each student tended to talk with all other students so forming a very cohesive network. As we can easily notice from Figure 2.15a in Chapter 2, no one was isolated and only one user talked with just another individual, while all the others had a degree greater than 3.

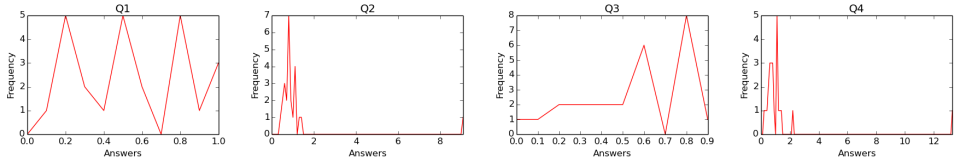


Figure 4.7: [DIAG2] Answers distribution in the first round

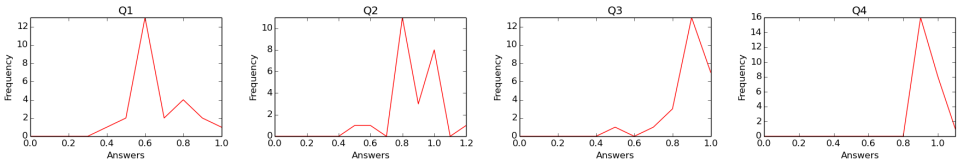


Figure 4.8: [DIAG2] Answers distribution in the second round

In Figures 4.7 and 4.8 we plotted the answers distribution for the two rounds, before and after the social interaction. Even more than the other social experiments, here the answers distribution observed in the second round are rather condensed and centered on the ground truth. While several peaks of the curve result before the social interaction, in the second round and in all the four questions there are just 1 or 2 peaks near to 1.

4.3 Experimental analysis

The main goal of this section is to present all the outcomes obtained from each deployed social experiment. Specifically, we first briefly discuss about the fact

of detecting and considering (or not) outliers, then we show several results concerning measures of group diversity and accuracy in all the experiments. All the code concerning this part of research is available at [31].

Performance measures. Let $A^v_{R_i}$ the answer given by the user v at the round R_i and T the right answer. We introduce $\mathbb{A}^v_{R_i} = \frac{A^v_{R_i}}{T}$, the normalized value of the answer $A^v_{R_i}$. The normalization allows us to observe and compare the results of all the experiments under the same numerical scale. We define the following measures of accuracy. For a user v, her *absolute error* at round i is $a^v_{R_i} = |A^v_{R_i} - T|$ and her *relative error* is $e^v_{R_i} = \frac{a^v_{R_i}}{T} = |\mathbb{A}^v_{R_i} - 1|$. The *average relative error* for the round i is defined as $E_{R_i} = \frac{1}{n}\sum_{v=1}^{n} e^v_{R_i}$. In addition, we define the *no-abs relative error* as the non-absolute version of the relative error: $\epsilon^v_{R_i} = \frac{A^v_{R_i} - T}{T} = \mathbb{A}^v_{R_i} - 1$. This is required to analyze the trend of the answers from the first to the second round. Indeed, while the relative error is the right measure to use in general, its non-absolute version allows not to lose information, such as the direction, that otherwise would be lost using the abs operator. Another fundamental measure to consider in our analysis is the distance between the answers of a same user v in two consecutive rounds: $d^v_{R_{i+1},R_i} = |A^v_{R_i+1} - A^v_{R_i}|$. Also in this case, we define the non-absolute normalized version: $\Delta^v_{R_{i+1},R_i} = \mathbb{A}^v_{R_i+1} - \mathbb{A}^v_{R_i}$.

4.3.1 Detecting and rejecting outliers

In datasets containing real-numbered values, the suspected outliers are the measured values that appear to lie outside the range of most of the other data values. The problem with outliers is that the arithmetic mean is very sensitive to the inclusion of any of them. Therefore, there are two options to try and solve such a problem: the first involves the removal of the suspected outliers, while the second implies the using of another statistic, such as the median. Since in our experiment we employed several measures, as well as the arithmetic mean, we investigated more about the detection and possible removal of suspicious outliers. As already disclosed in Section 4.1, there is a big debate in the research community. Since this problem goes beyond the purpose of our research, here we just introduce and discuss what we accomplished in our experiments and what we observed.

Suspected outliers were likely due to the fact that some users misunderstood how to properly answer the question (e.g., 20 instead of 20 M) or just because they had no idea about the answer. For this reason, a first step involved the observation of values one-by-one and rejection of those lying outside a reasonable range. This "rough" and preliminary operation was then replaced by an automatic detection in order to treat all the values and all the

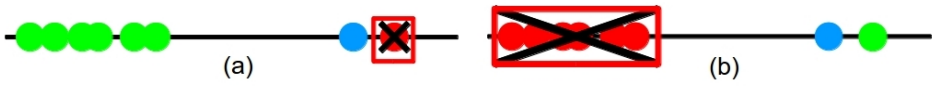


Figure 4.9: (a) Chauvenet's criterion. (b) Criterion based on the ground truth.

experiments following a same criterion. We thus investigated the Chauvenet's criterion [73]. The idea behind this criterion is to find a reasonable probability range containing all "n" samples of a dataset and centered on the mean of a normal distribution. In doing so, any value lying outside this probability range can be considered to be an outlier. It is then removed in order to compute a new mean and standard deviation on the remaining values. However, a strong criticism about this criterion is that it just provides an objective and quantitative method for data rejection, but not a scientific or methodological practice, especially in small datasets or where a normal distribution cannot be assumed. Since this criticism exactly reflects our condition of small datasets and uncertain normal distributions, we then considered an alternative method based on the ground truth (i.e., the correct answer for each question). As example of comparison between the two methods, consider a portion of a population answering similar values within a certain range very far from the ground truth, while just one user gives a good answer. According to the Chauvenet's criterion the closest answer to the ground truth should be rejected, but this may be an incorrect choice (see Figure 4.9a). Vice versa, if we consider a certain range centered on the ground truth where $A^v_{R_i} \in X$ and $\frac{T}{\varepsilon} \leqslant X \leqslant T \cdot \varepsilon, \forall \varepsilon \in \mathbb{R}$, that guarantees a rejection with respect to the ground truth. All answers should be smaller than ε times the ground truth and greater than the ground truth fractioned of a factor ε. Nevertheless, the opposite case to the Chauvenet's criterion may occur, too. Consider a dataset where most of the users give answers close to each other but very far from the ground truth. According to the criterion based on the ground truth they should be all rejected remaining without a valid dataset to analyze (see Figure 4.9b).

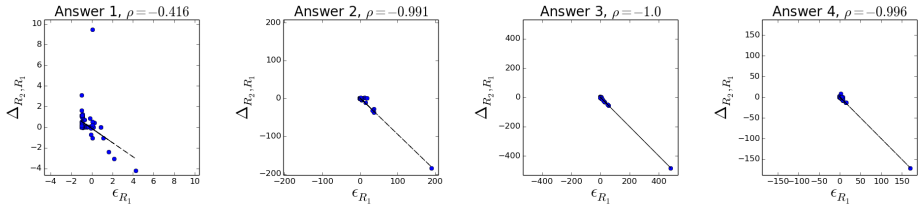
At this point, an important issue arises: what kind of outliers we have to reject? The ones far from the ground truth or the ones far from the average of all the answers? In our datasets we investigated both the mechanisms discovering that a great portion of answers are rejected in both of them. In the Priverno dataset, for instance, we observed that 12 of 60 values, namely the 20%, were detected as outliers and then rejected using the Chauvenet's criterion. Similarly, implementing the criterion based on the ground truth with a $\psi = 1000$, 8 users of 60 were rejected, i.e., around the 13.3%. Since percentages of this entity in small datasets could compromise the core of the data, then we decided to analyze our results with and without suspected outliers.

In the following sections we will show some results obtained in every setting in order to compare them.

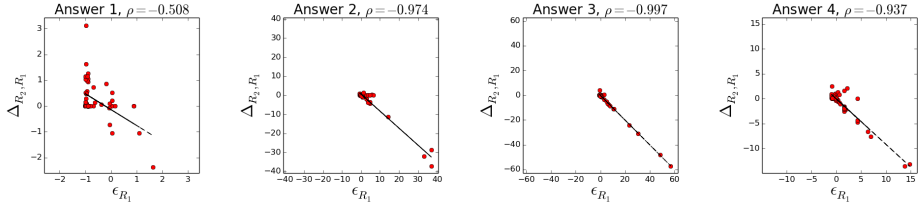
4.3.2 Results

A common result among all the real-world experiments is a trend of adjustment of the answers in the second round inversely proportional to the error made in the first phase. Such a behavior is partly responsible of the better accuracy achieved by the crowd after the social interaction. In other terms, a user underestimating the answer in the first round tends to increase her answer in the second round. Vice versa, a user overestimating the answer in the first round tends to decrease her answer in the second round. All those users whose answers are near to the ground truth tend to change just a little their estimation, not varying too much the previous value. Clearly, this behavior leads towards a convergence of all the answers to the ground truth. This result was achieved by correlating the error made by each user in the first round with respect to the ground truth, namely $\epsilon^v_{R_1}$, and the distance between their answers in the first and the second round, i.e., $\Delta^v_{R_2,R_1}$. This suggests how some of the users being certain of their estimation are less willing to change their answer. Such a behavior may be associated to the stubbornness and expertise of the individual. Of course, its intensity is not always the same and it can depend on several factors, such as the nature of the individual, the kind of question and the confidence of the neighborhood. In Section 4.4 we better analyze this peculiarity when the models tested in our experiments will be introduced.

Figures 4.10, 4.11, 4.12 and 4.13 show the results obtained in WSDM, DIAG1, Priverno and DIAG2, respectively. The figures denoted with the letter *(a)* show the results involving all the answers, while in the figures denoted with the letter *(b)*, we deleted suspected outliers using the Chauvenet's criterion. Similar results are obtained applying the criterion based on the ground truth. As can be easily observed the trend of the answers approximates a 45° falling line in both the cases (excepting a pair of answers where the slope is flatter). This gives a clear evidence of how social interaction influences users on their final estimation converging towards the right answer. In other terms, if the user's error $\epsilon^v_{R_1} = x$, then her distance $\Delta^v_{R_2,R_1} \approx -x$; vice versa, if $\epsilon^v_{R_1} = -x$, the distance will be $\Delta^v_{R_2,R_1} \approx x$. This means that users, after the social interaction, tend to adjust their estimation of a similar quantity equal to the error made before exchanging opinion with others. This is an interesting result considering that users do not have previous knowledge of their error. In addition, each plot reports the Person correlation coefficient ρ_P [170] between all the $\epsilon^v_{R_1}$ and all the $\Delta^v_{R_2,R_1}$, which is, in most of the cases, near to -1. Such a value indicates that the two vectors of data are strongly correlated (i.e., data



(a) All individuals



(b) No outliers

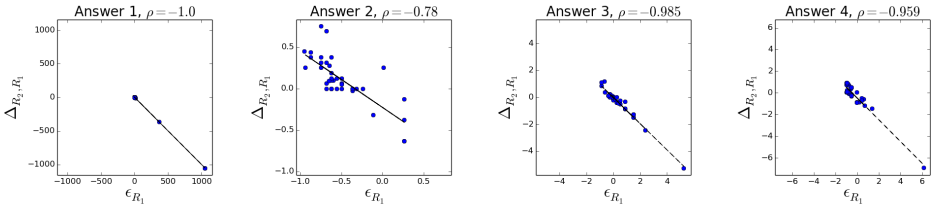
Figure 4.10: [WSDM] The correlation between the error made by participants in Round 1 (x-axis) and the distance of their answers between Round 2 and Round 1 (y-axis).

points lying exactly on a line) with a slope $m < 0$.

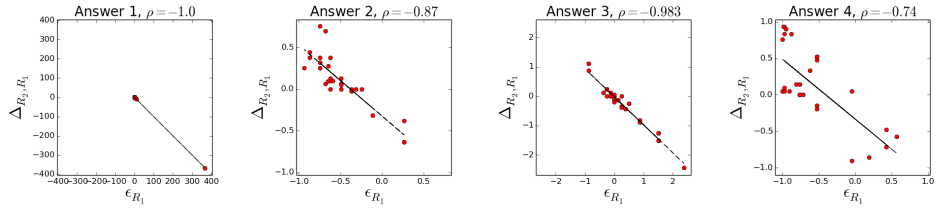
To better analyze and compare the accuracy reached by the crowd in both the rounds, we plotted the average relative error E_{R_i} for each question in both the two rounds. Figure 4.14 shows the results of all the experiments. Since the relative error is normalized by the ground truth, this plot gives an idea of the percentage of error made by the crowd. Of course, the more the error is low, the more the accuracy is high. We can notice how, in all the experiments and for each question, there was an improvement (i.e., a lower average relative error) in the second round. This means that the social interaction caused a positive effect on the wisdom of the network.

Another measure to consider for the analysis of the wisdom is the mean of all the normalized answers $\mathbb{A}^v_{R_i}$. Figure 4.15 shows the results for all the experiments. The black line $mean = 1$ depicts the right answer, consequently the more the bar is near to this line, the more the *average accuracy*[1] will be good. In 3 of 4 questions, the second round gives a better average estimation. Specifically, in the WSDM and DIAG2 experiments, the crowd always improves the estimation in the second round, while in the experiments Priverno and DIAG1, participants did better in 3 of 4 questions. This means that in

[1] Notice that the average network accuracy may be good even though single users' accuracy are worst. For instance, assume that the ground truth is, as usual, equal to 1. If a user answers 0.2 and another user answers 1.8, their single accuracy is not good at all, but their average exactly gives 1.

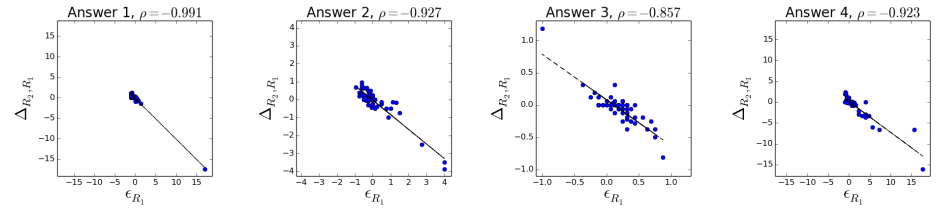


(a) All individuals

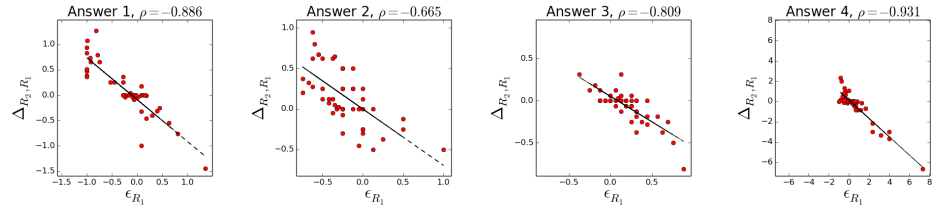


(b) No outliers

Figure 4.11: [DIAG1] The correlation between the error made by participants in Round 1 (x-axis) and the distance of their answers between Round 2 and Round 1 (y-axis).

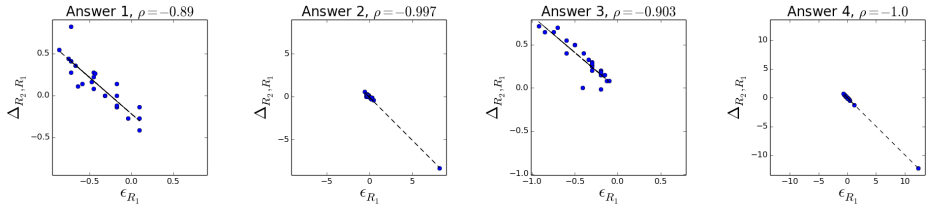


(a) All individuals

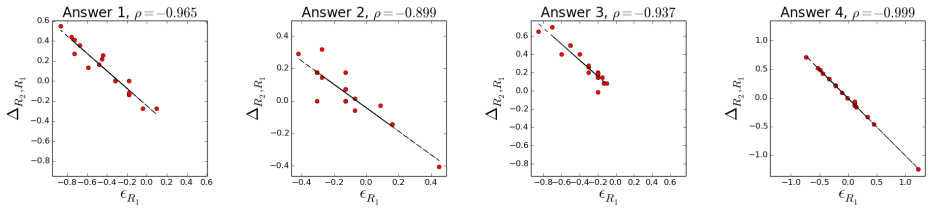


(b) No outliers

Figure 4.12: [Priverno] The correlation between the error made by participants in Round 1 (x-axis) and the distance of their answers between Round 2 and Round 1 (y-axis).



(a) All individuals



(b) No outliers

Figure 4.13: [DIAG2] The correlation between the error made by participants in Round 1 (x-axis) and the distance of their answers between Round 2 and Round 1 (y-axis).

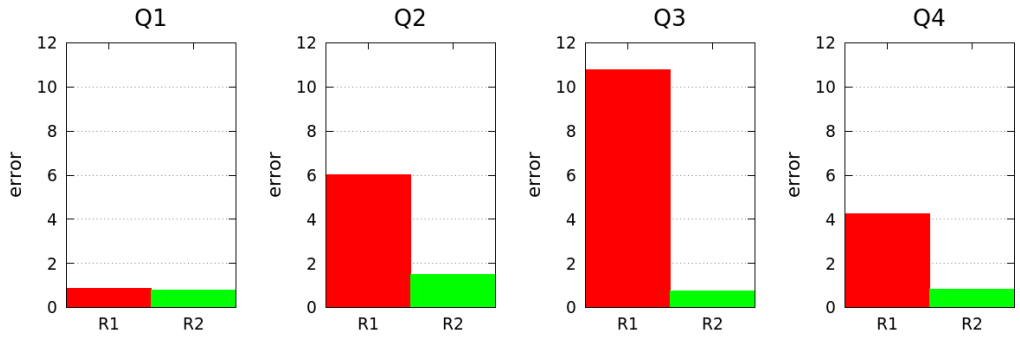
87.5% of the cases the social interaction brought a greater benefit to the network in terms of average estimation. Similar results are also achieved for the datasets filtered deleting suspected outliers. In this case, the crowd always performs better in the second round for each question. Due to the similarity of results, henceforth we will just show the analysis made by considering all the answers in the datasets.

At this point, since such results do not exactly reflect previous work on this phenomenon, we wondered what could be the reasons that brought a better accuracy after the social interaction. First of all, we must recall that in all experiments conducted by Lorenz et al. [147] and in other researches discussed in Section 4.1, the crowd was never allowed to interact in a F2F fashion, but rather participants were indirectly aware about pieces of information determined by authors. These types of setting could affect the level of confidence, consequently participants could answer values totally different from what they really think or are convinced. Assume that the n-th user has to give her estimation for a question. She has in mind to answer 10, but all the other $(n-1)$ users before him answered on average 30. With a high probability, the user will revise her opinion and will tend to answer a value near to crowd's opinion. This result is also discussed by Lorenz where the group diversity is very low in the cases of "aggregated info" and "full info". Of course, this is a good result in terms of consensus, but it could not be the same for the accuracy if the

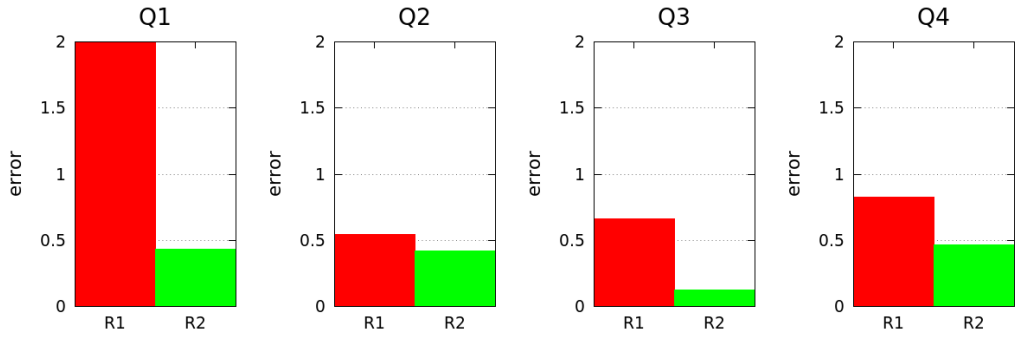
average value is far from the ground truth. Conversely, in our experiments we allowed participants to interact without any restriction, neither we influenced them with aggregated values or other kinds of similar information. We just measured their interactions, in a setting comparable to the real life. Clearly, this approach is pretty different with respect to the ones adopted so far. Since individuals usually rely on who they know to be a well-informed person, a good reasoner or an expert, participants of the experiments tend to form natural clusters in a network which will result more heterogeneous. Although the group diversity (i.e., the standard deviation) is lower in the second round than in the first one (see Figure 4.16), the accuracy is now given by opinions of different groups of participants, not by a homogeneous crowd sharing the same idea of answer. Analyzing all the figures discussed in this section, we can notice that, when the group diversity is very low (standard deviation near to 0), the accuracy may still be high (low error and mean near to 1), consequently previous results asserting that a low group diversity can negatively affect the wisdom of crowds do not seem to be confirmed in our real-world F2F social experiments. Actually, only the first question of the WSDM experiment has a lower standard deviation in the first round, but the accuracy is still better in the second one. For the cases where the average estimation is better in the first round (i.e., $Q4$ of the DIAG experiment and $Q1$ of Priverno), their errors and standard deviations are in any case lower in the second round. Therefore, we can affirm that, generally, even though the group diversity decreases after the social interaction, this effect does not compromise the wisdom of the crowd in such a natural setting.

In a next step we compared the users' accuracy with their authority in the network, coinciding with the degree in undirected graphs (see Chapter 1). Indeed, one natural question that could arise is: "Is the most popular user also the most accurate in guessing the true answer?". Well, apparently this does not seem to stand out as global property. We correlated the users' degree with both of their relative errors, $e^v_{R_1}$ and $e^v_{R_2}$, but this did not show anything of significant. Additionally, we compared their single distance between the two rounds $\Delta^v_{R_2,R_1}$ and their degree, but also in this case nothing emerged. We repeated all the correlations comparing other types of user centrality, such as betweenness, closeness and eigenvectorness. The finding was always the same. Thus, the only thing that can be state is that there is not any correlation between the single users' authorities and the accuracy of their answers. This may be explained by the fact that the influence of each user follows a so intricate and complex pattern that a direct comparison between users' centrality and answers' accuracy is not a feasible measure.

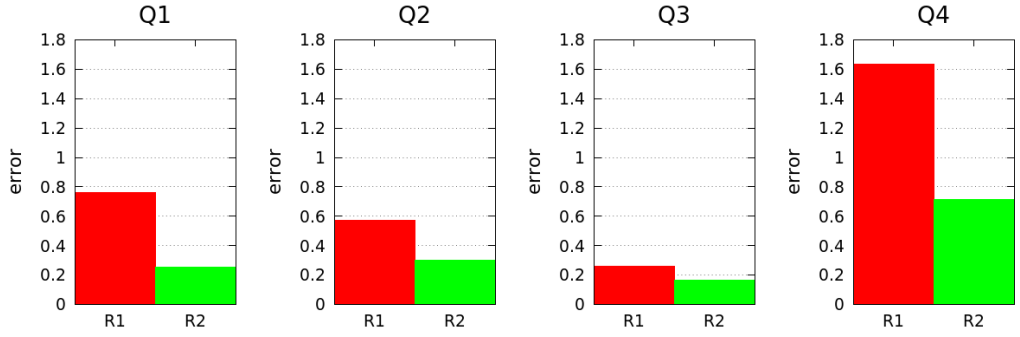
Instead, we observed that one of the factors that may bring improvement in terms of accuracy after the social interaction is the inclusion of the ground truth between the answers of two users having an edge. Figure 4.17 shows



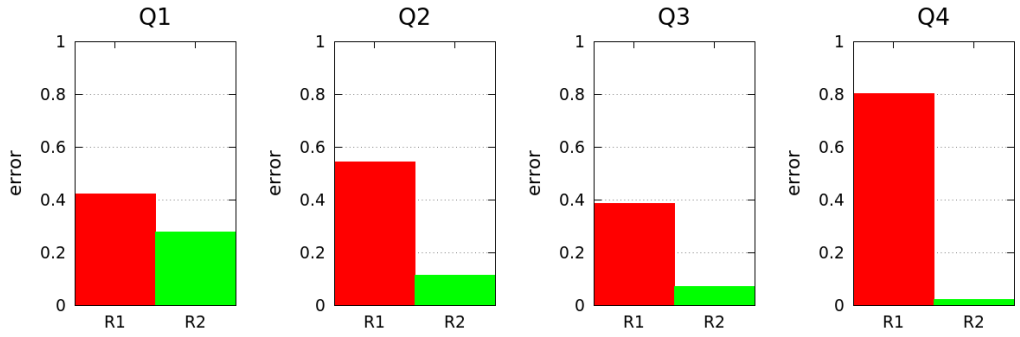
(a) WSDM



(b) DIAG1

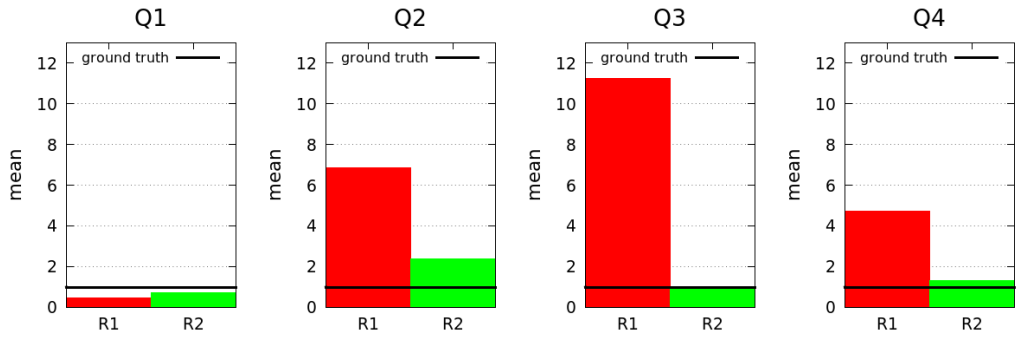


(c) Priverno

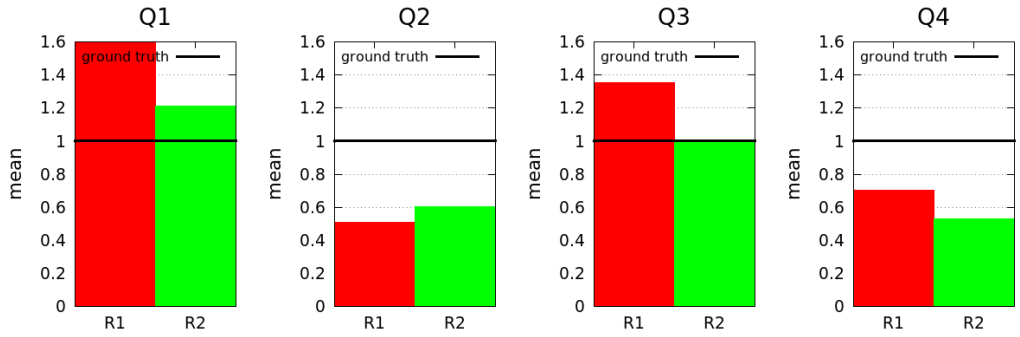


(d) DIAG2

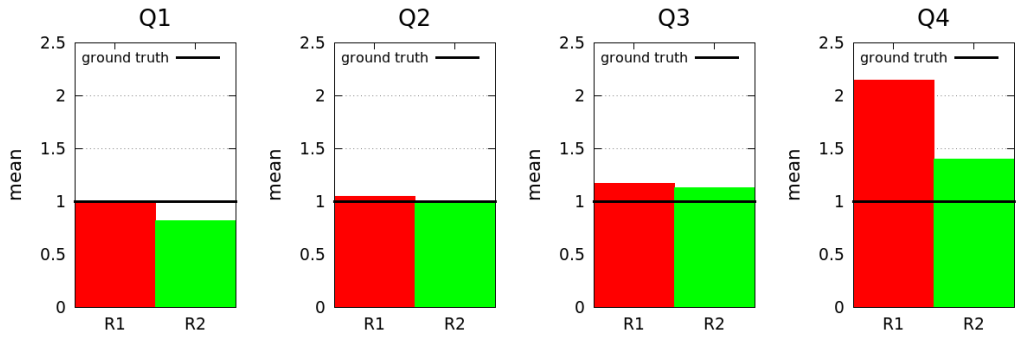
Figure 4.14: Plots showing the average relative error of all the answers in both the two rounds.



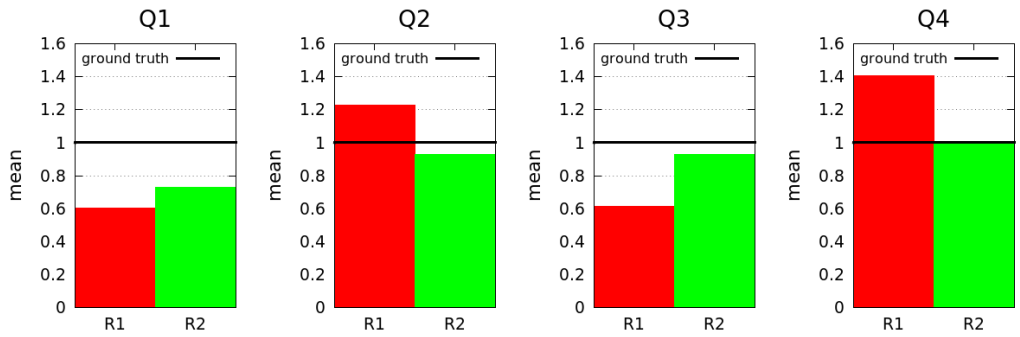
(a) WSDM



(b) DIAG1

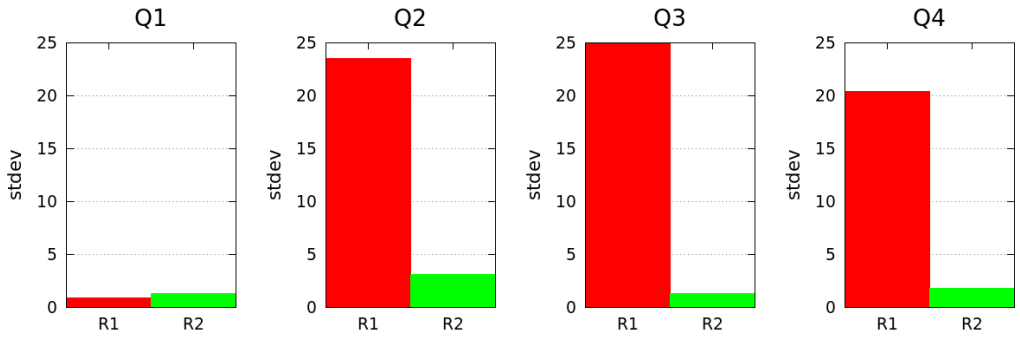


(c) Priverno

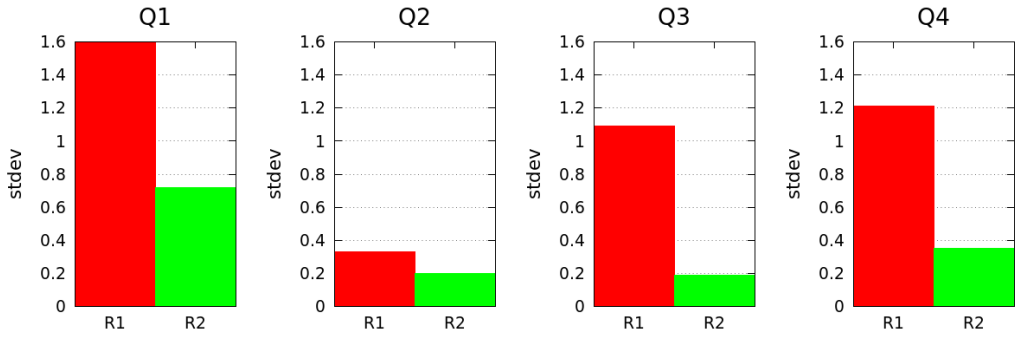


(d) DIAG2

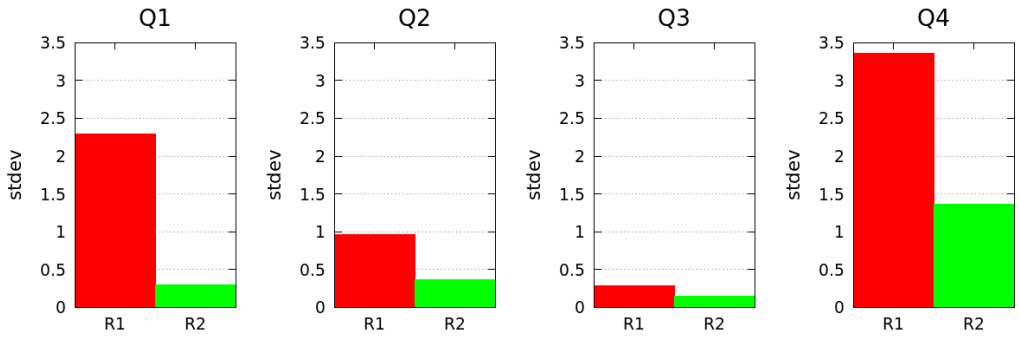
Figure 4.15: Plots showing the average of all the normalized answers in both the two rounds.



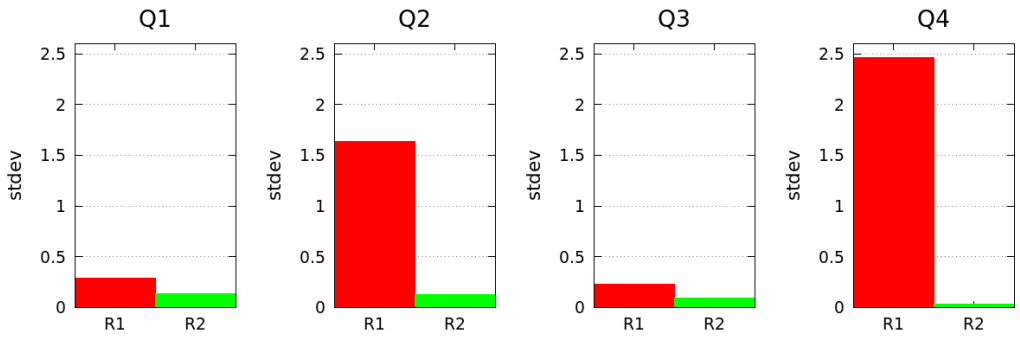
(a) WSDM



(b) DIAG1



(c) Priverno



(d) DIAG2

Figure 4.16: Plots showing the standard deviation of all the normalized answers in both the two rounds.

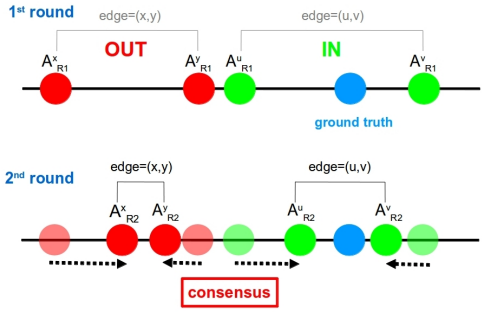


Figure 4.17: A sketch of how users tend to converge towards a common answer.

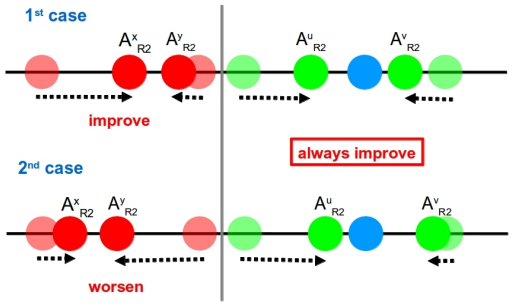


Figure 4.18: The two cases that can happen after the social interaction.

exactly such a principle. As simple instance, assume to have four users forming two couples. The two users for each couple will have a social interaction in the second round. In the figure showing the first round, the couple depicted with the green color includes the ground truth, namely one estimation is at the right of the correct answer, while the other estimation is on the left. Vice versa, the couple in red is out of the right answer. As previously discussed, since users having a social interaction tend to converge towards a common answer, the absolute error of each user in the couple including the ground truth will automatically decrease in the second round. Of course, the average improvement of the accuracy in the second round strictly depends on the number of couples that includes the ground truth in the first phase. However, there may be a case where the pair of individuals, although not including the right answer, could decrease its average error (see Figure 4.18). If one of those users is more confident about her own answer and she was already pretty near to the ground truth in the first phase, she may convince the other participant to answer a value similar to her estimation (see the first case of Figure 4.18). In doing so, although the users' estimations do not include the correct answer, the average of the couple may be closer to the ground truth in the second round than in the first one. However, the opposite case may also occur (see the second case of Figure 4.18).

We investigated this concept clustering each pair of users in two groups: in the first one the couples include the correct answer (dubbed as "in-group" in the following), while the second one is formed by all the other couples (dubbed as "out-group" in the following). The two groups were clustered based on the answers of users in the first round, but taking into account the couple formed during the social interaction. In this way, we are able to analyze if the couples

clustered in the first round and including the right answers, improve their accuracy in the second round due to their social interaction. For both of them we measured the average relative error on the edge $\mathfrak{e} = \langle u, v \rangle$, for all the pair of users:

$$E^{\mathfrak{e}}_{R_i} = \frac{1}{k} \sum_{\mathfrak{e}=1}^{k} \frac{|\mathbb{A}^u_{R_i} - 1| + |\mathbb{A}^v_{R_i} - 1|}{2}$$

Figures 4.19, 4.20, 4.21, 4.22 shows the results of each social experiment. All the plots are in spider representation where each question is on the border of an ax. The first round is depicted by purple and pink lines for the in-group and the out-group, respectively. Regarding the second round, the in-group and the out-group are represented by the blue and red lines, respectively. The main scope of these plots is to emphasize how, in most of the cases (i.e., at least 3 of 4 questions), the average relative error E_{R_2} of the in-group is usually less than the error of the out-group. This can be easily examined by the inclusion of the blue lines within the red lines. Of course, this intuition is not valid for the first round where people did not choose their partners for interacting yet.

The validation of this intuition proves that one of the factors affecting the improvement of the accuracy is the convergence of individuals towards a common answer due to the social interaction. This phenomenon always decreases the average relative error in cases where the ground truth is included between the two answers of the couple, consequently the more there are couples of this type, the more the crowd will have a good chance to improve its estimation.

4.4 Study and analysis of opinion formation models

In this section we investigate the ability of some models proposed in social sciences to describe the dynamical aspects of social influence in opinion-formation dynamics. In particular, we want to evaluate models based on the DeGroot work [83] and considered in [105], [61], [113] to account for the way in which the wisdom of crowds phenomenon is affected by the social network.

The repeated updating process employed by models of this type is simple and captures some of the basic aspects of social learning and structure. Since social science is an ancient discipline, it is not surprising that it has a long research activity behind it. One of the first studies was the sociological measures of centrality and prestige introduced by Leo Katz [130]. Other several papers discussing variations and improvements of this framework and mainly concerning the consensus rather than the wisdom, were presented in the following years by John R. P. French, Jr. [104], Frank Harary [114] and Noah E. Friedkin with Eugene C. Johnsen [106]. Thus, we start our evaluation analyzing the model in the DeGroot version, then we investigate other descending

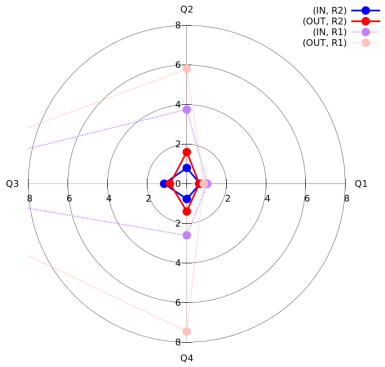


Figure 4.19: [WSDM] Clusters plot

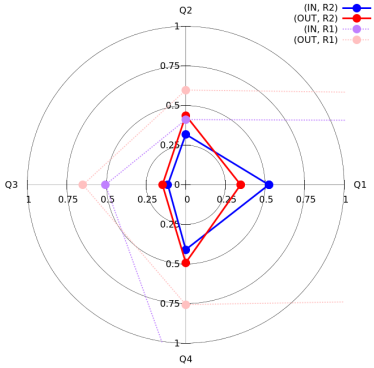


Figure 4.20: [DIAG1] Clusters plot

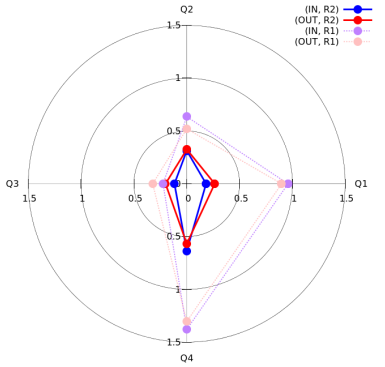


Figure 4.21: [Priverno] Clusters plot

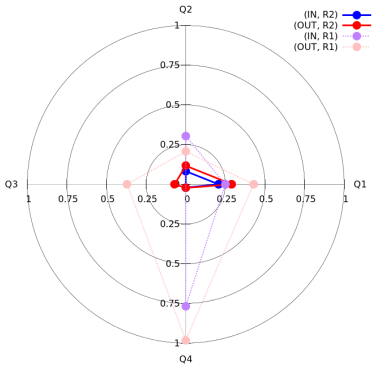


Figure 4.22: [DIAG2] Clusters plot

models using always the same modalities.

In the last part of this research, we propose a new variant of [61] which introduces a factor of weight δ_i on each internal user's opinion $p_i^{(0)}$. Our model seems to be more efficient and adaptable on spurious data in all the four social experiments.

4.4.1 The DeGroot model

One of the first studies about opinion formation and consensus in social networks was made by DeGroot in 1974 [83]. The author presented an iterative model which describes how a finite set $N = \{1, 2, \ldots, n\}$ of individuals, acting as a team or committee, may reach a common decision by exchanging their individual opinions with others. The social structure of this set is described

by a weighted and possibly (but not necessarily) directed network. Users have some own beliefs about some common question of interest and at each time period they communicate with their neighbors in the social network in order to update their opinions. Therefore, a new agent's belief is given by the weighted average of her neighbors' beliefs from the previous time period. The interaction network is represented by an $n \times n$ non-negative matrix $\mathbf{W}$, where w_{ij} indicates that i pays attention to j. Since the network may be directed, we can have that $w_{ij} > 0$, while $w_{ji} = 0$. This matrix is right-stochastic, in which each row sums to 1:

$$\sum_j w_{ij} = 1$$

Because each agent has a belief $p_i^{(t)} \in \mathbb{R}$ at time $t \in \{0, 1, 2, \ldots\}$, the vector of beliefs at time t is written as $\mathbf{p}^{(t)}$ and the updating iterative process is given by the following set of equations:

$$\mathbf{p}^{(t)} = \mathbf{W}\,\mathbf{p}^{(t-1)}$$

and so:

$$\mathbf{p}^{(t)} = \mathbf{W}^t\,\mathbf{p}^{(0)} \tag{4.1}$$

where $\mathbf{p}^{(0)}$ is the vector of beliefs at time $t = 0$, before any social interaction. The process goes on until a consensus is reached. In other terms, the beliefs of all agents in a network converge to well-defined limits if there is a $\mathbf{p}^{(\infty)}$ such that:

$$\lim_{t\to\infty} \mathbf{p}^{(t)} = \mathbf{p}^{(\infty)}$$

Now, we recall w_{ij} denoting the element in row i and column j of the matrix $\mathbf{W}$. Then it follows from (4.1) that a consensus is reached if and only if an *influence vector* $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$, for $i \in \{1, \ldots, n\}$ and $j = 1, \ldots, n$ exists, such that:

$$\lim_{t\to\infty} w_{ij}^{(t)} = \pi_j \tag{4.2}$$

In other words, the matrix $\mathbf{W}$ is convergent if, at the t-th iteration, the rows of the matrix are all equal to the influence vector $\boldsymbol{\pi}$, therefore it follows from (4.1) and (4.2) that:

$$\left(\lim_{t\to\infty} \mathbf{W}^t\,\mathbf{p}^{(0)}\right)_i = \boldsymbol{\pi}\,\mathbf{p}^{(0)} \tag{4.3}$$

for every $i \in \{1, \ldots, n\}$. This means that there is a unique left (or row)

eigenvector $\boldsymbol{\pi}$ of $\mathbf{W}$ corresponding to eigenvalue 1. To show why there is one involved eigenvector, first notice that:

$$\lim_{t\to\infty} \mathbf{W}^t\,\mathbf{p}^{(0)} = \lim_{t\to\infty} \mathbf{W}^t\,(\mathbf{W}\mathbf{p}^{(0)})$$

consequently, from (4.3) it must be:

$$\boldsymbol{\pi}\,\mathbf{p}^{(0)} = \boldsymbol{\pi}\,\mathbf{W}\,\mathbf{p}^{(0)}$$

for every $\mathbf{p}^{(0)} \in \mathbb{R}^n$. This implies that $\boldsymbol{\pi} = \boldsymbol{\pi}\,\mathbf{W}$, and so $\boldsymbol{\pi}$ is the unique left eigenvector of $\mathbf{W}$.

The condition of convergence is equivalent to assert that the graph is *strongly connected* and $\mathbf{W}$ is *aperiodic*. The characteristic of aperiodicity can be defined as [179]:

Definition 7. An $n \times n$ non-negative matrix $\mathbf{W}$ is *aperiodic* if for every pair i, i of its index set there exists a $T < \infty$ such that $(\mathbf{W}^t)_{ii} > 0$ for all $i \in \{1, \ldots, n\}$ and all $t \geqslant T$.

In addition, if a matrix is strongly connected, then it is irreducible [179].

Definition 8. An $n \times n$ non-negative matrix $\mathbf{W}$ is *irreducible* if for every pair i, j of its index set there exists a $T < \infty$ such that $(\mathbf{W}^t)_{ij} > 0$ for all $i, j \in \{1, \ldots, n\}$ and all $t \geqslant T$.

It can be proved that irreducible aperiodic non-negative matrices are the same as primitive matrices. The following definition establishes a relationship between primitivity and aperiodicity.

Definition 9. Let $\mathbf{W}$ be a strongly connected and stochastic matrix. It is aperiodic if and only if it is primitive.

Finally, the following definition lies the concept of convergence to the primitivity.

Definition 10. Let $\mathbf{W}$ be a strongly connected and stochastic matrix. If it is convergent, then it is primitive.

Example 1. As simple instance, let us consider the following transition matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 \end{pmatrix}$$

and the corresponding $\mathbf{W}^t$ for $t = 100, 101$:

$$\mathbf{W}^{100} = \begin{pmatrix} \frac{2}{3} & 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{2}{3} & 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{2}{3} \end{pmatrix}$$

$$\mathbf{W}^{101} = \begin{pmatrix} 0 & \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{2}{3} & 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{2}{3} & 0 & \frac{1}{3} & 0 \end{pmatrix}$$

Observing $\mathbf{W}^{101}$ we can assert that the above conditions of aperiodicity and irreducibility are not satisfied, so $\mathbf{W}$ is not primitive and a consensus cannot be reached. This can be even noticed by the fact that for $t = 2k$ $\mathbf{W}^t = \mathbf{W}'$, while for $t = 2k + 1$ $\mathbf{W}^t = \mathbf{W}''$, so proving the periodicity of the states.

Let us now consider the following matrix and its corresponding powers for $t = 100, 101$:

$$\mathbf{W} = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}$$

$$\mathbf{W}^{100} = \mathbf{W}^{101} = \begin{pmatrix} 0.14285714 & 0.14285714 & 0.28571429 & 0.42857143 \\ 0.14285714 & 0.14285714 & 0.28571429 & 0.42857143 \\ 0.14285714 & 0.14285714 & 0.28571429 & 0.42857143 \\ 0.14285714 & 0.14285714 & 0.28571429 & 0.42857143 \end{pmatrix}$$

In this case the primitivity is confirmed and, as expressed by (4.3), we can first take $\boldsymbol{\pi} = (0.14285714, 0.14285714, 0.28571429, 0.42857143)$ substantiating $\boldsymbol{\pi} = \boldsymbol{\pi}\,\mathbf{W}$, then we multiply the influence vector by a randomly selected vector of initial beliefs $\mathbf{p}^{(0)} = (0.1, 0.3, 0.6, 0)$:

$$\mathbf{p}^{(\infty)} = \boldsymbol{\pi}\,\mathbf{p}^{(0)} = (0.22857143, 0.22857143, 0.22857143, 0.22857143)$$

The resulting $\mathbf{p}^{(\infty)}$ is the vector of the final beliefs where a consensus is reached.

4.4.2 The Friedkin-Johnsen model

The second analyzed model we have taken into account is the Friedkin-Johnsen model [105] presented in 1990. This model is a DeGroot generalization and it has been taken as reference for other variations presented subsequently, such

as in [82]. According to the authors, the model has to force each node i to maintain a persistent internal opinion $p_i^{(0)}$ at each time period t. Thus, at each iteration, the output vector of beliefs is given by the linear combination of the vector of beliefs computed in the previous time period (endogenous variables), and properly multiplied by a scalar weight α, and the initial vector of beliefs $\mathbf{p}^{(0)}$ (exogenous variables), multiplied by a scalar weight β:

$$\mathbf{p}^{(t)} = \alpha\, \mathbf{W}\, \mathbf{p}^{(t-1)} + \beta\, \mathbf{p}^{(0)} \tag{4.4}$$

where $\mathbf{W}$ is a $n \times n$ matrix representing the interaction network. The general definition of the Friedkin-Johnsen model does not impose any constraints on the selection of parameters neither on the matrix, but in one of its realization the authors suggest to use the following parameters: $0 < \alpha < 1$ and $\beta = 1 - \alpha$, so as to have a proportionate impact of exogenous and endogenous variables.

Notice that the presence of the vector of initial beliefs $\mathbf{p}^{(0)}$ in each iteration prevents repeated averaging from bringing all nodes to the same opinion as happens in the DeGroot model. However, as proved by the authors, the set of equations given by (4.4) brings to the following state of equilibrium:

$$\mathbf{p}^{(\infty)} = \alpha\, \mathbf{W}\, \mathbf{p}^{(\infty)} + \beta\, \mathbf{p}^{(0)}$$

Thus, the following definition applies:

Definition 11. Suppose the equilibrium $\mathbf{p}^{(\infty)}$ is reached. Then, there exists a $T < \infty$ such that $\mathbf{p}^{(t)} = \mathbf{p}^{(t+1)} = \dots$ for all $t \geqslant T$.

4.4.3 The BKO model

In 2011 Bindel et al. [61] published a fundamental paper analyzing how much your own opinion is influenced from others, and in which way: good or bad? Their work built on the basic model of DeGroot with the extension given by Friedkin and Johnsen, in which each individual i holds a persistent internal opinion $p_i^{(0)}$. The updating process is given by the following rule:

$$p_i^{(t)} = \frac{p_i^{(0)} + \sum_{j \in \mathcal{N}(i)} m_{i,j} p_j^{(t-1)}}{1 + \sum_{j \in \mathcal{N}(i)} m_{i,j}} \tag{4.5}$$

where $\mathcal{N}(i) = \{\, j : m_{i,j} > 0 \,\}$ is the set of neighbors of user i, while the quantity $\sum_{j \in \mathcal{N}(i)} m_{i,j}$ denotes her weighted degree. The corresponding matrix notation is:

$$\mathbf{p}^{(t)} = \mathbf{W}\, \mathbf{p}^{(t-1)} + \mathbf{A}\, \mathbf{p}^{(0)} \tag{4.6}$$

where:

$$\mathbf{A} = diag(\alpha_1, \ldots, \alpha_n) \quad \text{with} \quad \alpha_i = \frac{1}{1+\sum_{j \in \mathcal{N}(i)} m_{i,j}}$$
$$\mathbf{W}_{i,j} = \frac{m_{i,j}}{1+\sum_{j \in \mathcal{N}(i)} m_{i,j}}$$

Observing (4.6), we can easily notice how this model is, de facto, a little variation of (4.4). The main difference is the selection of the parameter α. While in the Friedkin-Johnsen model the parameter α is a scalar weight that may be static (i.e., $\alpha_1 = \alpha_2, = \ldots$) or dynamic (i.e., changing at every iteration), the BKO model assigns every user a predetermined own α, so that they compose a diagonal matrix A. Eventually, (4.6) converges to:

$$\mathbf{p}^{(\infty)} = \mathbf{W}\,\mathbf{p}^{(\infty)} + \mathbf{A}\,\mathbf{p}^{(0)}$$

Consequently, Definition 11 given by the Friedkin-Johnsen model, applies also in this case.

4.4.4 Our model

The model we propose and that we have successfully tested during our research activity is a variation of the BKO model. In our opinion, the approach adopted by Bindel et al. can be, in general, the most suitable in estimating the vector of final beliefs $\mathbf{p}^{(\infty)}$. Thanks to their choice to assign a specific factor α_i to each user. In doing so, the model adapts its estimation to the characteristics of the users, which are, in reality, the main actors of the model.

However, the estimation of each α_i in the BKO model is fixed a-priori, just depending on the sum of the edge weights between each pair of nodes i and j, with $j \in \mathcal{N}(i)$. This may be a little constraining because it implicitly imposes that the weight given to the internal belief $p_i^{(0)}$ just depends on i's relationships. Generally, this could not be true since each user could change her own idea about her initial estimation basing on both her relationships and her own *stubbornness*. Therefore, we investigated the possibility to introduce a further factor of stubbornness δ_i, so that (4.5) becomes:

$$p_i^{(t)} = \frac{\delta_i\, p_i^{(0)} + \sum_{j \in \mathcal{N}(i)} m_{i,j} p_j^{(t-1)}}{\delta_i + \sum_{j \in \mathcal{N}(i)} m_{i,j}} \tag{4.7}$$

where $\mathcal{N}(i)$ and $\sum_{j \in \mathcal{N}(i)} m_{i,j}$ are defined as in the BKO model, while δ_i denotes the factor of weight of the user i on her internal belief $p_i^{(0)}$. From (4.7) it follows the equation in matrix notation:

$$\mathbf{p}^{(t)} = \mathbf{W}\,\mathbf{p}^{(t-1)} + \mathbf{A}\,\mathbf{p}^{(0)} \tag{4.8}$$

where:

$$\mathbf{A} = diag(\alpha_1, \ldots, \alpha_n) \quad \text{with} \quad \alpha_i = \frac{\delta_i}{\delta_i + \sum_{j \in \mathcal{N}(i)} m_{i,j}}$$
$$\mathbf{W}_{i,j} = \frac{m_{i,j}}{\delta_i + \sum_{j \in \mathcal{N}(i)} m_{i,j}} \quad \text{such that} \quad i \neq j$$

The estimation of the stubbornness values δ_i may be obtained by iteratively executing the least-squares fitting (see Appendix A.4) for each user i until the corresponding residual, between the estimation of the beliefs and the vector of the real opinions given after the social interaction, does not reach a local minimum. We compute δ_i as the personal internal weight of each individual that depends on the full set of questions given to users. We compute the residuals as below:

$$r_i^{(t)}\left(\delta_i^{(t)}, \mathfrak{p}_i^{(0)}, \mathfrak{p}_j^{(t-1)}, \mathfrak{p}_i^{(1)}\right) = \left\| \mathfrak{p}_i^{(1)} - \frac{\delta_i^{(t)} \mathfrak{p}_i^{(0)} + \sum_{j \in \mathcal{N}(i)} m_{i,j} \mathfrak{p}_j^{(t-1)}}{\delta_i^{(t)} + \sum_{j \in \mathcal{N}(i)} m_{i,j}} \right\| \tag{4.9}$$

where $\delta_i^{(t)}$ is the stubbornness of the user i computed at the iteration t, $\mathfrak{p}_i^{(0)}$ is the vector of the user i containing the internal opinions related to all the questions, $\mathfrak{p}_j^{(t-1)}$ is the vector of all the users $j \in \mathcal{N}(i)$ containing their opinions at the time $t-1$, while $\mathfrak{p}_i^{(1)}$ is the vector of opinions of the user i given after the social interaction. Updating $\delta_i^{(t)}$ in (4.9) is the same as choosing δ_i to minimize:

$$\left(\delta_i \, \mathfrak{p}_i^{(1)} - \delta_i \, \mathfrak{p}_i^{(0)}\right)^2 + \sum_{j \in \mathcal{N}(i)} m_{i,j} \left(\mathfrak{p}_i^{(1)} - \mathfrak{p}_j^{(t-1)}\right)^2 \tag{4.10}$$

(4.10) denotes the *cost* that the user i incurs by choosing a given value of δ_i. Of course, since we are using as term of comparison the values of the opinions collected after the social interaction, as happens in such optimization problems, the model always tries to minimize this cost so as to reduce the distance from the estimation as much as possible.

Once the vector $\boldsymbol{\delta}$ is selected, we can compute the model as shown in (4.7) and (4.8). We prove that (4.8) converges to a unique state of equilibrium.

Lemma 6. The above model converges to a unique equilibrium if $\delta_i > 0$ for all i.

Proof. Consider any equilibrium $\mathbf{p}^{(\infty)}$ for Equation 4.8. Then, for each of them $\mathbf{p}^{(\infty)}$ satisfies

$$\mathbf{p}^{(\infty)} = \mathbf{W}\,\mathbf{p}^{(\infty)} + \mathbf{A}\,\mathbf{p}^{(0)}$$

Thus, we have

$$(\mathbf{I} - \mathbf{W})\,\mathbf{p}^{(\infty)} = \mathbf{A}\,\mathbf{p}^{(0)}$$

and so

$$\mathbf{p}^{(\infty)} = (\mathbf{I} - \mathbf{W})^{-1}\mathbf{A}\,\mathbf{p}^{(0)}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix. In order that $(\mathbf{I}-\mathbf{W})^{-1}$ may exist, $\mathbf{I}-\mathbf{W} \in \mathbb{R}^{n\times n}$ must be non-singular. For any row i of $\mathbf{W}$ we have that any matrix norm $\|\mathbf{W}\| < 1$ because $\mathbf{W}$ is a sub-stochastic matrix for which $\|\mathbf{W} + \mathbf{A}\| = 1$ is always verified (being $\mathbf{A}$ is a sub-stochastic non-negative matrix). Since the *spectral radius*[2] $\rho(\mathbf{W}) \leqslant \|\mathbf{W}\|$, then the Neumann series $\mathbf{I} + \mathbf{W} + \mathbf{W}^2 + \ldots$ converges and $\lim_{n\to\infty} \mathbf{W}^n = 0$ [152]. In which case, $\sum_{n=0}^{\infty} \mathbf{W}^n = (\mathbf{I} - \mathbf{W})^{-1}$, so $\mathbf{I} - \mathbf{W}$ is invertible and the equilibrium is unique. □

4.4.5 Experimental analysis

In this section we present all the results obtained by running the discussed models on our real-world social experiments. As already described in previous sections, the employed technology allowed us to collect social interactions between individuals. However, we do not have any direction in interactions, consequently all the experiments' networks are represented as undirected graphs. This brings $\mathbf{W}$ to be a stochastic matrix of an undirected graph, where the following condition applies: if $w_{ij} > 0$, then $w_{ji} > 0$. In terms of modeling, this feature does not have any effect.

We run all the models in two different settings. First, we run the simulation applying the iterative rule of each model on the aggregate stochastic matrix $\mathbf{W}$ of every experiment, then we repeat the simulation on each of the corresponding evolving network $\mathcal{G}$ in which, at each time-step t, the stochastic matrix $\mathbf{G}^{(t)}$ is computed.[3] To the best of our knowledge, this is the first attempt in running opinion formation models on evolving networks.

4.4.5.1 The DeGroot model

In the following, we show the results obtained running the DeGroot Model in all our four experiments, computing the final vector of beliefs $\mathbf{p}^{(\infty)}$, its error

[2] The spectral radius of any matrix $\mathbf{A}$ is defined as $\rho(\mathbf{A}) = \max_{\lambda\in\sigma(\mathbf{A})} |\lambda|$, where $\sigma(\mathbf{A})$ is the eigenvalues set of $\mathbf{A}$.

[3] Notice that the notation $\mathbf{G}^{(t)}$ adopted in the evolving case is different from $\mathbf{W}^t$ used in the iterative case. $\mathbf{G}^{(t)}$ indicates the t-th stochastic matrix at time t in the evolving network $\mathcal{G}$, while $\mathbf{W}^t$ is the t-th power matrix of the stochastic matrix $\mathbf{W}$ representing the aggregate graph in the iterative case.

E with respect to the ground truth and the average error $E_{\mathbf{p}^{(1)}}$ with respect to $\mathbf{p}^{(1)}$.

WSDM experiment. In the WSDM experiment the aggregate stochastic matrix $\mathbf{W}$ does not satisfy the primitivity, so a consensus is not reached. For this reason, we cannot show the result related to the iterative case for this experiment. On the contrary, a consensus is reached running the model on the evolving network $\mathcal{G}$, such that $\mathbf{p}^{(t)} = \mathbf{G}^{(t)}\,\mathbf{p}^{(t-1)}$. Table 4.1 shows the results. Due to the high number of suspected outliers, the errors are very high. Actually, among all the experiments, the WSDM one is the most challenging in terms of modeling.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mathbf{p}^{(\infty)}$	E	$E_{\mathbf{p}^{(1)}}$
1	0.727	0.461	0.539	0.659
2	2.406	6.780	5.780	5.092
3	0.901	11.252	10.252	10.341
4	1.323	4.710	3.710	3.700

Table 4.1: [WSDM] The DeGroot evolving case.

DIAG1 experiment. Tables 4.2 and 4.3 shows the results obtained by running the DeGroot model in the DIAG1 experiment for the iterative case and the evolving case, respectively.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mathbf{p}^{(\infty)}$	E	$E_{\mathbf{p}^{(1)}}$
1	1.213	71.217	70.217	70.004
2	0.602	0.507	0.493	0.146
3	1.004	1.140	0.140	0.208
4	0.532	0.672	0.328	0.351

Table 4.2: [DIAG1] The DeGroot iterative case.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mathbf{p}^{(\infty)}$	E	$E_{\mathbf{p}^{(1)}}$
1	1.213	43.665	42.665	42.452
2	0.602	0.575	0.425	0.132
3	1.004	1.131	0.131	0.202
4	0.532	0.679	0.321	0.352

Table 4.3: [DIAG1] The DeGroot evolving case.

Note that the results of the first question are quite compromised in both the cases because of suspected outliers. Applying the model on the network purified by suspicious values, the results improve a lot. For instance, regarding the concerned question, $\mathbf{p}^{(\infty)} = 1.241$ in the iterative case, so both the errors are pretty reduced.

As for the other questions, the value of the consensus reached by the model is quite near to the average of $\mathbf{p}^{(1)}$. Indeed, the error $E_{\mathbf{p}^{(1)}}$ ranges between around 0.13 and 0.35 both in the iterative and evolving case. This is an interesting fact since the value given by the consensus mainly follows the average

of the beliefs obtained after the social interaction. In other words, the crowd starting with a certain vector of beliefs $\mathbf{p}^{(0)}$, gives a final vector of beliefs $\mathbf{p}^{(1)}$ having an average that approximates the estimated $\mathbf{p}^{(\infty)}$.

Priverno experiment. Tables 4.4 and 4.5 show the results obtained in the Priverno experiment.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mathbf{p}^{(\infty)}$	E	$E_{\mathbf{p}^{(1)}}$
1	0.820	0.884	0.116	0.225
2	0.982	1.110	0.110	0.314
3	1.130	1.177	0.177	0.124
4	1.398	1.502	0.502	0.770

Table 4.4: [Priverno] The DeGroot iterative case.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mathbf{p}^{(\infty)}$	E	$E_{\mathbf{p}^{(1)}}$
1	0.820	0.832	0.168	0.227
2	0.982	1.102	0.102	0.312
3	1.130	1.183	0.183	0.127
4	1.398	2.084	1.084	1.107

Table 4.5: [Priverno] The DeGroot evolving case.

Also in this case, the iterative and the evolving processes provide very similar results. The first three questions have both the errors, E and $E_{\mathbf{p}^{(1)}}$, oscillating in a range similar to the previous experiment. On the contrary, the last question in the iterative case has an error greater than 75% compared to $\mathbf{p}^{(1)}$. Higher values are given by the evolving case. We recall that this question asked to guess the number of dots in a figure.

DIAG2 experiment. Finally, tables 4.6 and 4.7 show the results given by running the DeGroot model on the DIAG2 interaction traces.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mathbf{p}^{(\infty)}$	E	$E_{\mathbf{p}^{(1)}}$
1	0.727	0.648	0.352	0.111
2	0.927	0.944	0.056	0.098
3	0.928	0.611	0.389	0.319
4	0.991	1.580	0.580	0.589

Table 4.6: [DIAG2] The DeGroot iterative case.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mathbf{p}^{(\infty)}$	E	$E_{\mathbf{p}^{(1)}}$
1	0.727	0.670	0.330	0.096
2	0.927	1.104	0.104	0.187
3	0.928	0.571	0.429	0.357
4	0.991	1.815	0.815	0.824

Table 4.7: [DIAG2] The DeGroot evolving case.

Similarly to the Priverno's experiment, both the errors of the first three questions range between 0.10 and 0.35, and in the last question, asking the number of dots, the errors are greater than 55% in the iterative case and greater than 80% in the evolving case.

Discussion about the iterative and evolving process. All the results obtained in the iterative and evolving process seem to suggest that the effects on the vector of the estimated beliefs produced by the t-th power of

the aggregate transition matrix $\mathbf{W}$ and the forward-product [179, Chapter 3] of the single transition matrices on the evolving network $\mathcal{G}$ are quite similar. In other terms, the vector of beliefs $\mathbf{p}^{(t)}$ computed at a time-step t, with $t \to \infty$, by the t-th matrix power $\mathbf{W}^t$, approximates the vector computed by using the forward-product $\prod_{\tau=1}^{t} \mathbf{G}^{(\tau)}$. Notice that the sum of each single transition matrix on $\mathcal{G}$ can be approximated to the expected aggregate transition matrix $\mathbf{E}[\mathbf{W}]$[4], namely $\frac{1}{t}\sum_{\tau=1}^{t} \mathbf{G}^{(\tau)} \approx \mathbf{E}[\mathbf{W}]$. Consequently, $(\mathbf{E}[\mathbf{W}])^t = \left(\frac{1}{t}\sum_{\tau=1}^{t} \mathbf{G}^{(\tau)}\right)^t$. Similarly, the expected forward-product can be written as $\mathbf{E}[\prod_{\tau=1}^{t} \mathbf{G}^{(\tau)}] = \prod_{\tau=1}^{t} \mathbf{E}[\mathbf{G}^{(\tau)}] = (\mathbf{E}[\mathbf{W}])^t$. It stands to reason that the two quantities can be approximated each other. Obviously, in general, this is not confirmed, but an approximation exists if we deal with stationary distributions [179]. Although we are not working with similar distributions, the single transition matrices have some properties such that, for a sufficiently large t, this approximation follows.

4.4.5.2 The Friedkin-Johnsen model

The results achieved using the Friedkin-Johnsen model are very similar to the previous ones obtained by DeGroot. In each experiment, we estimated the parameter α by executing a least-squares fitting that minimized the quantity $\mathbf{p}^{(\infty)} - \mathbf{p}^{(1)}$, then we measured the average on the vector of final beliefs $\mu\left(\mathbf{p}^{(\infty)}\right)$ and both the two errors E and $E_{\mathbf{p}^{(1)}}$. We noticed that, in most of the cases, since the minimization provided by the least-squares fitting gave a $\alpha \approx 1$, and so a $\beta \approx 0$ due to the applied realization using $0 < \alpha < 1$ and $\beta = 1 - \alpha$, such a fit approximates, de facto, the Friedkin-Johnsen model to the DeGroot's one.

In the following we just report the WSDM experiment in the iterative case since its results were missed in the DeGroot model due to the lack of primitivity of $\mathbf{W}$. Table 4.8 shows the obtained results. As already observed in the DeGroot model, the average of the final vector of beliefs $\mu\left(\mathbf{p}^{(\infty)}\right)$ is rather far from the ground truth and $\mathbf{p}^{(1)}$. This is due to the being of some initial beliefs very spurious. Deleting suspected outliers the situation improves enough, even though the errors remain pretty high with respect to the other experiments.

4.4.5.3 The BKO model

The latest well-known model tested on our real-world social networks was the one presented by Bindel, Kleinberg and Oren. As already done for the previous models, at first we applied the iterative process (4.6) on the matrices $\mathbf{W}$ and $\mathbf{A}$ built following the BKO rule, then we repeated the simulation by executing

[4] See Chapter 3 for a formal definition of the expected matrix.

Q	α	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	0.539	0.727	0.459	0.685	0.649
2	0.986	2.406	4.572	3.572	3.118
3	0.976	0.901	15.670	14.670	14.768
4	0.999	1.323	2.460	1.479	1.823

Table 4.8: Results of the Friedkin-Johnsen model on the WSDM experiment.

the model on the evolving network $\mathcal{G}$ of each experiment. In this second case, the matrices $\mathbf{W}$ and $\mathbf{A}$ must be replaced with $\mathbf{G}^{(t)}$ and $\mathbf{B}^{(t)}$, recomputed at every time-step. Therefore, the set of equations given by (4.6) becomes:

$$\mathbf{p}^{(t)} = \mathbf{G}^{(t)}\,\mathbf{p}^{(t-1)} + \mathbf{B}^{(t)}\,\mathbf{p}^{(0)}$$

where $\mathbf{G}^{(t)}$ and $\mathbf{B}^{(t)}$ denote the BKO matrices $\mathbf{W}$ and $\mathbf{A}$ computed at time t. Of course, since, in general, $\mathbf{G}^{(t)} \neq \mathbf{G}^{(t-1)}$ and $\mathbf{B}^{(t)} \neq \mathbf{B}^{(t-1)}$ the condition of convergence of $\mathbf{p}^{(t)}$ is not satisfied, so we just stop the iterative process as soon as the evolving network reaches its latest time-step.

In the following we report a table for each method and experiment, so that we can easily compare the two approaches. Tables 4.9, 4.11, 4.13 and 4.15 show the results obtained in the iterative process, while in the tables 4.10, 4.12, 4.14 and 4.16 we reported the results for the evolving case.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	0.727	0.456	0.860	0.624
2	2.406	6.847	5.981	4.682
3	0.901	11.246	10.683	10.538
4	1.323	4.720	4.193	4.08

Table 4.9: [WSDM] The BKO iterative case.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	0.727	0.460	0.790	0.627
2	2.406	6.755	5.867	4.678
3	0.901	10.312	9.739	9.695
4	1.323	4.697	4.088	4.067

Table 4.10: [WSDM] The BKO evolving case.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	1.213	67.828	67.242	67.036
2	0.602	0.510	0.538	0.197
3	1.004	1.351	0.635	0.589
4	0.532	0.701	0.791	0.602

Table 4.11: [DIAG1] The BKO iterative case.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	1.213	69.135	68.767	68.555
2	0.602	0.517	0.511	0.165
3	1.004	1.428	0.551	0.553
4	0.532	0.714	0.657	0.581

Table 4.12: [DIAG1] The BKO evolving case.

As can be noticed, also in the BKO model the two processes produce similar results. Specifically, the evolving case has, in most of the cases, the two errors a little bit lower, but the difference is very minimal.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	0.820	0.983	0.746	0.600
2	0.982	1.053	0.549	0.436
3	1.130	1.173	0.256	0.129
4	1.398	2.144	1.595	1.206

Table 4.13: [Priverno] The BKO iterative case.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	0.820	0.889	0.609	0.508
2	0.982	1.058	0.483	0.444
3	1.130	1.172	0.224	0.115
4	1.398	2.113	1.483	1.107

Table 4.14: [Priverno] The BKO evolving case.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	0.727	0.601	0.415	0.228
2	0.927	1.228	0.530	0.452
3	0.928	0.614	0.386	0.315
4	0.991	1.408	0.758	0.749

Table 4.15: [DIAG2] The BKO iterative case.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	0.727	0.593	0.414	0.187
2	0.927	1.221	0.491	0.443
3	0.928	0.615	0.385	0.313
4	0.991	1.371	0.680	0.669

Table 4.16: [DIAG2] The BKO evolving case.

If we compare these results to the ones obtained in the DeGroot model, we discover that the consensus of users $\mathbf{p}^{(\infty)}$ reached in DeGroot is, in general, quite similar to the average $\mu(\mathbf{p}^{(\infty)})$ given by the BKO model. However, here the error $E_{\mathbf{p}^{(1)}}$ is, in most of the cases, higher than the two previous models. This is mainly due to the fact that the vector of final beliefs $\mathbf{p}^{(\infty)}$ in DeGroot has all equal values due to the consensus, so their standard deviation is always null. As already observed in Figure 4.16, the group diversity is rather small in the second round, so the $E_{\mathbf{p}^{(1)}}$ in the DeGroot model is more likely to be slower than in the BKO model, where the vector $\mathbf{p}^{(\infty)}$ has all different values with a significant standard deviation, unless the estimation given by the BKO model is pretty accurate for each user.

4.4.5.4 Our model

In this section we report the results achieved by running the proposed model on our real-world social experiments. We fixed a vector of stubbornnesses $\boldsymbol{\delta}$ previously computed using the least-squares method, then we run the model assigning every individual i her corresponding stubbornness value δ_i. As already measured in the other investigated models, each table shows the average of the estimated beliefs vector, the average relative error between $\mathbf{p}^{(\infty)}$ and the ground truth, and the average relative error between $\mathbf{p}^{(\infty)}$ and $\mathbf{p}^{(1)}$.

The obtained results are rather better than the ones achieved using the other tested models. If, for instance, we look at the first question of the DIAG1 experiment, which was a very bad affair for all the previous models, we can notice that in our proposed model the error $E_{\mathbf{p}^{(1)}}$ is 1.401. Although this value

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	0.727	0.388	0.808	0.591
2	2.406	2.602	1.683	1.104
3	0.901	1.979	1.426	1.178
4	1.323	1.869	1.212	1.167

Table 4.17: Results for WSDM.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	1.213	0.277	1.514	1.401
2	0.602	0.500	0.514	0.151
3	1.004	1.044	0.336	0.311
4	0.532	0.581	0.782	0.530

Table 4.18: Results for DIAG1

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	0.820	0.705	0.371	0.273
2	0.982	0.953	0.311	0.198
3	1.130	1.156	0.195	0.104
4	1.398	1.445	0.759	0.432

Table 4.19: Results for Priverno.

Q	$\mu\left(\mathbf{p}^{(1)}\right)$	$\mu\left(\mathbf{p}^{(\infty)}\right)$	E	$E_{\mathbf{p}^{(1)}}$
1	0.727	0.699	0.300	0.152
2	0.927	0.900	0.125	0.065
3	0.928	0.760	0.240	0.182
4	0.991	1.024	0.084	0.085

Table 4.20: Results for DIAG2.

is still pretty high because of spurious data, it is much lower than the other models where their estimation was always greater than 65. The measured worst values are always related to the WSDM experiment, but also in this case they are quite improved if compared to the other models. In all the four questions we got an average relative error $E_{\mathbf{p}^{(1)}} < 1.2$ and the average of the estimated beliefs $\mu\left(\mathbf{p}^{(\infty)}\right)$ is closer to the average of the real beliefs after the social interaction $\mu\left(\mathbf{p}^{(1)}\right)$. Vice versa, in all the previous models, the average relative error even reaches values greater than 10 and $\mu\left(\mathbf{p}^{(\infty)}\right)$ is often pretty far from $\mu\left(\mathbf{p}^{(1)}\right)$. Also in the cases of Priverno and DIAG2, our model gets better results in terms of accuracy with respect to $\mu\left(\mathbf{p}^{(1)}\right)$. Similar findings are achieved running the model on the evolving network $\mathcal{G}$.

4.5 Concluding remarks

In this chapter we have first introduced the wisdom of crowds phenomenon, an application analyzed on our F2F social networks, then we have presented the results obtained from the social experiments. We have observed that the social interaction produced by individuals participating to each experiment can improve the accuracy of the corresponding answers. This is a rather curious effect since, so far, previous work said the contrary. We have explained that one of the possible cause producing the opposite findings could be the nature of the network. Finally, we have analyzed some of the most famous opinion-formation models and proved that, with a simple generalization of the BKO model, the accuracy of the final estimation can increase.

Conclusions and Future Work

This thesis addresses several topics in the fields of evolving social networks and collective computation and provides the following main contributions:

- **Collection of a number of state-of-the-art F2F real-world evolving networks**. Most research work on evolving networks is carried out on on-line networks datasets (e.g., tweets and Facebook posts). Moreover, the literature on F2F evolving networks either takes into consideration synthetic graphs (i.e. graphs generated according to a specific model) or real graphs an order of magnitude smaller than the ones collected in our experiments. The collection of our graphs and their effective and efficient representation has required both the analysis/comparison of a number of existing technologies and a careful system design and implementation. Eventually, we selected the SocioPatterns sensing platform as the best fit solution to achieve our goals and we implemented some novel approaches for the effective acquisition of F2F interactions in heterogeneous scenarios.

- **Computing authority on evolving networks**. Estimating authority (e.g., centrality scores using the PageRank model) on huge social networks can be computationally expensive, so the problem of distributing the computational load on single entities has been recently considered in literature. Furthermore, maintaining PageRank under the condition of network updates (i.e., evolving networks) is an important and non trivial task. Most researchers have so far adopted an approach based on the computation of centrality scores at each instant of time. However, it is not clear if such method reflects the fairest measure to be used. We proposed two notions consistent with PageRank in static weighted networks and successfully operating on evolving networks by keeping trace of the past history. A first method considered the score computation on the expected network, while a second one employed the average topology over a suitable window of the most recent snapshots. We further showed that, when an evolving network satisfies stationary and homogeneous

properties, our proposed decentralized algorithms continuously maintain an accurate estimate of the PageRank computed over the current "expected" network. Clearly, real-life evolving networks may not exhibit stationary properties. For this reason, we presented a modified heuristic able to better manage and compute centrality scores on this kind of networks. The experimental analysis, carried out on different datasets, confirmed the validity and the feasibility of the approach presented in this thesis.

- **Wisdom of crowds effect on real-world social networks**. This phenomenon has been largely studied and analyzed over the past decade, but no one has examined its dynamics in social networks resulting from physical interactions of people. Most researchers deployed social experiments allowing users to "implicitly" interact via computer. Interactions among users occurred for the simple fact that information about the answers of other participants in the crowd is gathered. On the other hand, we exploited some real-world social experiments to study the wisdom of crowds effect in a more natural and intuitive way. Obtained results suggest that social interactions can enhance the wisdom of a crowd. This gives a clear evidence of how physical contact can improve the average wisdom of people. On second thoughts, this actually occurs in real life too. When in need of advice or help, we usually turn to a friend or a relative we consider wise and reliable, not to a random stranger we may happen to meet. As a final step, we analyzed and evaluated various opinion formation models in order to understand if they work alright on real data. Interestingly, we discovered that not only are models pretty accurate in some cases (i.e., not particularly affected by spurious data), but they also produce similar results if directly executed on evolving networks instead of using the iterating process on the aggregate graph. We finally proposed a generalized model based on a previous work that proved to be more accurate also in those cases where spurious data is largely present.

In the future, we plan to extend the real-world experiments to a higher number of individuals in order to study bigger and denser networks. In this perspective, we are working to further improve the data collection process in order to deploy experiments in a faster and easier way. Indeed, one of the most challenging logistics problems in our experiments is the instrumentation of the experimental area with a suitable number of readers. To address this problem we are working on a solution that employs mobile phones as gateways for data collection.

The availability of bigger and denser evolving networks will allow us to study relevant phenomena, such as the spreading of information, epidemics

and opinion trends formation at an unprecedented scale, to possibly reveal new and interesting behaviors. We also plan to further enhance the computation of centrality scores on evolving networks beyond the PageRank. We stress again that, at present, a standard definition of centrality scores in evolving networks does not exist, so the main effort will be devoted to find such definition and propose it to the research community at large. We are also planning to work on applications that may use the distributed heuristics developed in this thesis as core concept to compute centrality scores; an example could be a smartphone application to assess fame and popularity of users playing a social game.

Finally, regarding the wisdom of crowds phenomenon, we plan to investigate the main differences in various types of network in real-world scenarios (e.g., predefined or supervised networks). In particular, we want to replicate social experiments deployed in previous works as well as arrange cyber-world social experiments. This will hopefully let us better compare what happens to the wisdom of a crowd under different network conditions. To this end, we are programming a new chat software, named *WoChat* [32], enabling users to interact with other participants via private-channels and to properly answer questionnaires.

Annexes

List of publications

Part of this thesis has been published in the following journal articles, conferences and workshop proceedings:

- Capturing Interactions in Face-To-Face Social Networks. Francesco Ficarola and Andrea Vitaletti. In Proceedings of the of the 11th International Conference on Web Information Systems and Technologies (WEBIST), 2015 (forthcoming).

- Extending TETRA with Wireless Sensor Networks. Mario Paoli, Francesco Ficarola, Ugo Maria Colesanti, Andrea Vitaletti, Simona Citrigno, Domenico Saccà. In International Journal of Intelligent Engineering Informatics, 2015 (forthcoming).

- Distributed Sensor Network for Multi-robot Surveillance. A. Pennisi, F. Previtali, F. Ficarola, D.D. Bloisi, L. Iocchi, A. Vitaletti. In Procedia Computer Science, 2014. vol. 32, no. 0, pp. 1095 - 1100.

- T. Arzilli, F. Ficarola, K. Massri, A. Vitaletti, F. Loriga, I. De Marinis, A. Ferraresi, R. Bloise, and M. Goretti. 2013. ProvinciaSense: extending the capillary WiFi infrastructure of Lazio region with static and mobile sensor networks. In Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys '13). ACM, New York, NY, USA, , Article 68 , 2 pages.

- Luca Becchetti, Lorenzo Bergamini, Francesco Ficarola, Francesco Salvatore, Andrea Vitaletti. First Experiences with the Implementation and Evaluation of Population Protocols on Physical Devices. 2012 IEEE International Conference on Green Computing and Communications, pp. 335-342, 2012 IEEE International Conference on Green Computing and Communications, 2012.

- Luca Becchetti, Lorenzo Bergamini, Francesco Ficarola, and Andrea Vitaletti. 2012. Population protocols on real social networks. In Proceedings of the 9th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN '12). ACM, New York, NY, USA, 17-24.

- Luca Becchetti, Lorenzo Bergamini, Francesco Ficarola, and Andrea Vitaletti. 2012. Population protocols on real social networks. In Proceedings of the Fifth Workshop on Social Network Systems (SNS '12). ACM, New York, NY, USA, , Article 15 , 2 pages.

Appendix A

Appendices

A.1 Eigenvectors and Eigenvalues

Given an $n \times n$ square matrix $\mathbf{A}$, an *eigenvector* $\mathbf{x}^* \in \mathbb{R}^n$ is a vector such that

$$\mathbf{A}\,\mathbf{x}^* = \lambda\,\mathbf{x}^* \tag{A.1}$$

for some $\lambda \in \mathbb{C}$, which is called the *eigenvalue* of $\mathbf{x}^*$. Of course, a vector with zero components always solves (A.1), but we usually interested in nonzero solutions.

Notice that eigenvectors come in two shapes: *right eigenvectors* or *left eigenvectors*, which are also called *column* and *row eigenvectors*, respectively. The terms "right" and "left" simply refer to their multiplication side with respect to the matrix $\mathbf{A}$. Thus, the vector $\mathbf{x}^*$ in (A.1) is a $n \times 1$ right eigenvector, while in the following formula it is a $1 \times n$ left eigenvector:

$$\mathbf{x}^*\,\mathbf{A} = \lambda\,\mathbf{x}^*$$

The usefulness of eigenvectors can be seen in many applications (e.g., computation of eigenvector or Katz centrality or PageRank). Furthermore, the Perron-Frobenius theorem (see [179]) implies that if $\mathbf{A}$ is a nonnegative left stochastic matrix, with each column summing to 1, then there exists a nonnegative right eigenvector $\mathbf{x}^*$ solving (A.1) and having a corresponding eigenvalue $\lambda = 1$. This theorem also works for right stochastic matrices, with each row summing to 1, and left eigenvectors. Let's see an example to further clarify the concept.

Example 2. Let $\mathbf{A}$ be a 3×3 right stochastic matrix:

$$\mathbf{A} = \begin{pmatrix} 0.1 & 0.2 & 0.7 \\ 0.5 & 0. & 0.5 \\ 1. & 0. & 0. \end{pmatrix}$$

We want to find a left (row) eigenvector having eigenvalue $\lambda = 1$. To do that, we use the *Power iteration* algorithm (see Appendix A.2). Thus, assuming to have an initial $1 \times n$ vector $\mathbf{x}^{(0)} = (0.1, 0.3, 4.)$, from Power iteration it follows that

$$\lambda\, \mathbf{x}^{(t)} = \mathbf{x}^{(t-1)}\, \mathbf{A}$$

but, since $\lambda = 1$, then

$$\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)}\, \mathbf{A} = \mathbf{x}^{(0)}\, \mathbf{A}^t$$

Eventually, the algorithm converges finding the eigenvector $\mathbf{x}$:

$$\mathbf{x}^* = \mathbf{x}^{(0)}\, \mathbf{A}^*$$

In our example we set $t = * = 100$. We checked that this value is sufficiently big to ensure convergence of our stochastic matrix $\mathbf{A}$, hence:

$$\mathbf{x}^* = \mathbf{x}^{(0)}\, \mathbf{A}^{100}$$

where:

$$\mathbf{A}^{100} = \mathbf{A}^* = \begin{pmatrix} 0.5 & 0.1 & 0.4 \\ 0.5 & 0.1 & 0.4 \\ 0.5 & 0.1 & 0.4 \end{pmatrix}$$

and the resulting left eigenvector is $\mathbf{x}^* = (2.2, 0.44, 1.76)$.

Let's compute other two eigenvectors starting from other two initial vector states $\mathbf{x}^{(0)}$. So, for $\mathbf{x}^{(0)} = (1., 2., 3.)$ we have $\mathbf{x}^* = (3., 0.6, 2.4)$, and for $\mathbf{x}^{(0)} = (0.3, 0.4, 0.3)$ we have $\mathbf{x}^* = (0.5, 0.1, 0.4)$. If we now build a matrix formed by all the three left eigenvectors we have found, we have:

$$\mathbf{V} = \begin{pmatrix} 2.2 & 0.44 & 1.76 \\ 3. & 0.6 & 2.4 \\ 0.5 & 0.1 & 0.4 \end{pmatrix}$$

then, we can verify that the following equality is verified:

$$\mathbf{V}\, \mathbf{A} = \mathbf{\Lambda}\, \mathbf{V} \tag{A.2}$$

where $\mathbf{\Lambda} = diag(1, 1, 1)$ is a diagonal matrix having on its diagonal all unit components, corresponding to the eigenvalues 1. From (A.2) it follows that if

$\mathbf{V}$ is invertible, then $\mathbf{A}$ is said *diagonalizable*:

$$\mathbf{A} = \mathbf{V}^{-1}\,\mathbf{\Lambda}\,\mathbf{V}$$

This decomposition is useful to compute higher powers of $\mathbf{A}$ and more generally to calculate speeds of convergence:

$$\mathbf{A}^t = \mathbf{V}^{-1}\,\mathbf{\Lambda}^t\,\mathbf{V}$$

If we now transform $\mathbf{A}$ into its corresponding transpose $\mathbf{A}^T$ in order to get a left (column) stochastic matrix, and repeat the calculation with column vectors $\mathbf{x}^{(0)}$ having the same components of the row vectors used in this example, we exactly obtain the right eigenvectors with the same values of the left ones.

A.2 Power Iteration

Given a matrix $\mathbf{A}$, the power iteration algorithm will return an eigenvalue λ and the corresponding eigenvector $\mathbf{x}$ such that $\lambda\,\mathbf{x} = \mathbf{A}\,\mathbf{x}$. The algorithm starts with a vector $\mathbf{x}^{(\mathbf{0})}$ being an approximation to the dominant eigenvector or a random vector, then the method goes on by iterating the following expression:

$$\mathbf{x}^{(t)} = \frac{\mathbf{A}\,\mathbf{x}^{(t-1)}}{\|\mathbf{A}\,\mathbf{x}^{(t-1)}\|}$$

so, at every iteration, $\mathbf{x}^{(t)}$ is multiplied by the adjacency matrix $\mathbf{A}$ and normalized. The algorithm will eventually find only one eigenvalue (the one with the greatest absolute value) $\lambda = \|\mathbf{A}\,\mathbf{x}^{(*)}\|$ and the corresponding eigenvector $\mathbf{x}^{(*)}$.

A.3 Dynamic Network Format

Specifications

- DNF is case sensitive.
- Comments in DNF starts with the hash character (`#`) and extend to the end of the physical line.
- Whitespaces or blank lines are not considered.

A.3.1 Syntax

A DNF file is composed by three main sections: *header*, *nodes* and *edges*. Each of them is described in the remainder of the section.

Header The header is initialized by the following code:

```
[header]
```

In the header section we can put all the information about the graph, its possible dynamics and the attributes of the nodes and edges:

```
1 graphtype:{static | dynamic}, defaultedgetype:{undirected |
    directed | mixed}
2 dynamics:{timetype=value,start=value,end=value,timeunit=value}
3 nodeattrs:{label,attr1,attr2,...}, edgeattrs:{label,weight,attr1,
    attr2,...}
```

In the first line of the code above the graph type (static or dynamic) and the default edge type (undirected or directed) are expressed. The second line can be skipped if the network we are going to represent is static, otherwise we can put all the information related to the dynamics of the network. `timetype` and `start` attributes are mandatory, whereas the `end` attribute can be omitted. If a specific start time does not exist, then `start=0` must be set. The `timetype` attribute explicits what "type of time" we want to use. At present it supports: *UNIX timestamp* [29], *dateTime* [17] (expressed in the full zulu form) or *custom*. The custom type can be defined in the corresponding scenario or software. In addition, it is possible to use the `timeunit` attribute to indicate how long each gap is. For instance, if we have a `timeunit=5`, then a gap is far from its previous gap five units of time. Regarding continuous gaps, their total duration is given by the time unit value multiplied by the continuous gap, e.g., +3 coincides with 15. If the `timeunit` attribute is not expressed, then the default time unit is equal to 1. In the third line, nodes and edges attributes are listed. If the label is equal to the object ID, then we can omit it.

Nodes This section starts with the following line:

```
[nodes]
```

Here we can add all the information about the nodes of the graph, according to the following syntax:

```
[nodeID] {label_value, attr1_value,attr2_value,...} (gap1,gap2,
    gap3,...)
```

where the label and the attribute values are present if and only if in `nodeattrs` of the header section we have entered the `label` and the attributes keywords, respectively. In the case they are not expressed, the line must be written in the following way:

```
[nodeID] (gap1,gap2,gap3,...)
```

where the sequence `(gap1,gap2,gap3,...)` is present if and only if the graph type is dynamic.

Edges The edges' section is initialized as below:

```
[edges]
```

In this section we insert all the information regarding undirected edges according to the following syntax:

```
[sourceID,targetID] {label_value,weight_value,attr1_value,
    attr2_value,...} (gap1,gap2,gap3,...)
```

or directed edges according to the following line:

```
[sourceID>targetID] {label_value,weight_value,attr1_value,
    attr2_value,...} (gap1,gap2,gap3,...)
```

As for undirected edges, the two nodes are separate by `,` while regarding directed edges the direction is expressed by `>`. The first node is the source, whereas the second node is the target.

Excepting the weight value, which is present if and only the `weight` keyword has been expressed in the `edgeattrs` of the header, the rest of the attributes follow the same rules already seen in the nodes section.

A.3.2 Examples

In this section we list two examples to further clarify the just described syntax.

Example 3. Static graph, with no attributes and labels.

```
1  # Graph configuration
2  [header]
3  graphtype:{static}, defaultedgetype:{undirected}
4  nodeattrs:{}, edgeattrs:{}
5
6  # Information about nodes
7  [nodes]
8  [1001]
9  [1002]
10 [1003]
11 [1004]
12
13 # Information about edges
14 [edges]
15 [1001,1002]
16 [1001,1003]
17 [1002,1004]
```

Example 4. Mixed dynamic graph with attributes and weights.

```
# Graph configuration
[header]
graphtype:{dynamic}, defaultedgetype:{mixed}
dynamics:{timetype=timestamp,start=1318836335}
nodeattrs:{gender,age}, edgeattrs:{weight}

# Information about nodes
[nodes]
[1001] {M,22} (10,+3,2,+4)
[1002] {F,23} (11,+2,31)
[1003] {M,20} (9,+10,2)
[1004] {F,25} (12,1,31,+2)

# Information about edges
[edges]
[1001,1002] {2} (11,1)
[1001>1003] {2} (10,+3)
[1002>1004] {3} (12,1,31)
```

A.4 Least-squares fitting

The method adopted to estimate the parameters of models was the least-squares fitting. The least-squares is a technique of optimization (or regression) that allows to find a function such that its curve of regression is near to the given dataset as much as possible. Specifically, this function has to minimize the sum of the squares of the residuals, which are the distances between the dataset and the curve representing the function. For instance, suppose we want to fit a set of data $\{\mathbf{x}_i, \mathbf{y}_i\}$ to a known model $\mathbf{f}(\mathbf{x}, \boldsymbol{\beta})$, where $\boldsymbol{\beta}$ is the vector of parameters for the model that need to be found. Then, the residual is usually defined for each observed data-point as:

$$r_i(\boldsymbol{\beta}, \mathbf{x}_i, \mathbf{y}_i) = \|\mathbf{y}_i - \mathbf{f}(\mathbf{x}, \boldsymbol{\beta})\|$$

An objective function to pass to any of the previous minimization algorithms to obtain a least-squares fit can be:

$$S(\mathbf{p}) = \sum_{i=1}^{n} r_i^2(\mathbf{p})$$

Thus, the least-squares method finds its optimum when the sum S is minimum.

Bibliography

[1] Auditorium Antonianum. http://www.auditoriumantonianum.it/. 63, 112

[2] Centrality contribution code. https://github.com/francesco-ficarola/centrality. 101

[3] Cytoscape. http://cytoscape.github.io/. 22

[4] D3.js. http://d3js.org/. 22

[5] DNF Plug-ins. https://github.com/umbertogriffo/EvolvingDynamicNetwork. 34

[6] Dynamic Network Format (DNF). https://github.com/francesco-ficarola/dnf. 5, 32, 33

[7] GDF. http://guess.wikispot.org/The_GUESS_.gdf_format. 18

[8] Gephi. https://gephi.github.io/. 20, 22, 60

[9] GEXF. http://gexf.net/format/. 20

[10] GEXF Schema. http://gexf.net/format/schema.html. 20

[11] Gexf4j. https://github.com/francesco-ficarola/gexf4j. 22, 23

[12] GML. http://www.fim.uni-passau.de/index.php?id=17297&L=1. 18

[13] GraphML. http://graphml.graphdrawing.org/. 19

[14] GraphML Schema. http://graphml.graphdrawing.org/specification/schema_element.xsd.htm. 19

[15] GUESS. http://graphexploration.cond.org/. 18, 22

[16] iGraph. http://igraph.org/. 21

[17] ISO 8601. http://en.wikipedia.org/wiki/ISO_8601. 154

[18] Maven Central Repository. http://search.maven.org/. 23

[19] MIT Center for Collective Intelligence. http://scripts.mit.edu/~cci/HCI/. 3

[20] NetworkX. https://networkx.github.io/. 21

[21] nRF24L01 radio. http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01. 51

[22] OpenBeacon. http://www.openbeacon.org/. 5, 51

[23] Pajek. http://pajek.imfm.si/doku.php. 22

[24] RFC 4180. http://tools.ietf.org/html/rfc4180. 17

[25] sigmajs. http://sigmajs.org/. 22

[26] SocioPatterns. http://www.sociopatterns.org/. 5, 45

[27] Stanford Network Analysis Project. http://snap.stanford.edu/. 2

[28] The R Project for Statistical Computing. http://www.r-project.org/. 22

[29] UNIX Time. en.wikipedia.org/wiki/Unix_time. 154

[30] W3C XML Schema Definition Language (XSD). http://www.w3.org/TR/xmlschema11-2/. 33

[31] Wisdom of crowds contribution code. https://github.com/francesco-ficarola/wisdom. 118

[32] WoChat. https://github.com/francesco-ficarola/wochat. 147

[33] WSDM Conference 2013. http://www.wsdm2013.org/. 63, 112

[34] XML. http://www.w3.org/XML/. 19

[35] MCU PIC16F688. http://www.microchip.com/PIC16F688, 2009. 51

[36] 300 Visitors. http://miltosmanetas.com/MACROeo-300-Visitors, 2011. 61

[37] DIAG0 dataset. http://www.dis.uniroma1.it/~ficarola/research/social-experiments/diag-experiment/, 2011. 60

[38] DIAG0 live demo. http://www.dis.uniroma1.it/~ficarola/live/diag/, 2011. 61

[39] MACRO dataset. http://www.dis.uniroma1.it/~ficarola/research/social-experiments/macro-experiment/, 2011. 62

[40] MACRO live demo. http://www.dis.uniroma1.it/~ficarola/live/macro/, 2011. 61

[41] The SOCIOMAC patent. https://www.google.com/patents/US8660490, 2014. 53

[42] Norman Abramson. The aloha system: Another alternative for computer communications. In *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference*, AFIPS '70 (Fall), pages 281–285, New York, NY, USA, 1970. ACM. 46

[43] Eytan Adar. Guess: a language and interface for graph exploration. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 791–800. ACM, 2006. 22

[44] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005. 45

[45] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed computing*, 18(4):235–253, 2006. 58

[46] Nils Aschenbruck, Aarti Munjal, and Tracy Camp. Trace-based mobility modeling for multi-hop wireless networks. *Computer Communications*, 34(6):704 – 714, 2011. 49

[47] James Aspnes and Eric Ruppert. An introduction to population protocols. In *Middleware for Network Eccentric and Mobile Applications*, pages 97–120. Springer, 2009. 58

[48] Chen Avin, Michal Kouckỳ, and Zvi Lotker. How to explore a fast-changing world (cover time of a simple random walk on evolving graphs). In *Automata, Languages and Programming*, pages 121–132. Springer, 2008. 30, 79

[49] Konstantin Avrachenkov, Nelly Litvak, Danil Nemirovsky, and Natalia Osipova. Monte carlo methods in pagerank computation: When one iteration is sufficient. *SIAM Journal on Numerical Analysis*, 45(2):890–904, 2007. 76, 77, 81, 82, 88, 89, 90, 96

[50] Bahman Bahmani, Abdur Chowdhury, and Ashish Goel. Fast incremental and personalized PageRank. *Proceedings of the VLDB Endowment*, 4(3):173–184, 2010. 75, 76, 78, 83, 89

[51] Bahman Bahmani, Ravi Kumar, Mohammad Mahdian, and Eli Upfal. PageRank on an evolving graph. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 24–32. ACM, 2012. 75, 78, 83

[52] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999. 42

[53] A. Barrat, C. Cattuto, V. Colizza, F. Gesualdo, L. Isella, E. Pandolfi, J.-F. Pinton, L. Ravà, C. Rizzo, M. Romano, J. Stehlé, A.E. Tozzi, and W. Broeck. Empirical temporal networks of face-to-face human interactions. *The European Physical Journal Special Topics*, 222(6):1295–1309, 2013. 52

[54] Alain Barrat, Ciro Cattuto, Vittoria Colizza, Jean-François Pinton, Wouter Van den Broeck, and Alessandro Vespignani. High resolution dynamical mapping of social interactions with active rfid. *CoRR*, abs/0811.4170, 2008. 45

[55] Alain Barrat, Ciro Cattuto, Martin Szomszor, Wouter Van den Broeck, and Harith Alani. Social dynamics in conferences: Analysis of data from the live social semantics application. In *Proceedings of the 9th International Semantic Web Conference (ISWC 2010)*, 2010. 52

[56] Mathieu Bastian, Sebastien Heymann, Mathieu Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. *ICWSM*, 8:361–362, 2009. 22

[57] Vladimir Batagelj and Andrej Mrvar. *Pajek—analysis and visualization of large networks*. Springer, 2004. 22

[58] Luca Becchetti, Lorenzo Bergamini, Francesco Ficarola, and Andrea Vitaletti. Population protocols on real social networks. In *Proceedings of the 9th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, PE-WASUN '12, pages 17–24, New York, NY, USA, 2012. ACM. 58, 59

[59] Joyce E Berg, Forrest D Nelson, and Thomas A Rietz. Prediction market accuracy in the long run. *International Journal of Forecasting*, 24(2):285–300, 2008. 3, 107

[60] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside PageRank. *ACM Transactions on Internet Technology (TOIT)*, 5(1):92–128, 2005. 75, 77

[61] David Bindel, Jon Kleinberg, and Sigal Oren. How bad is forming your own opinion? In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 57–66, Washington, DC, USA, 2011. IEEE Computer Society. 7, 129, 130, 134

[62] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008. 61

[63] Vladimir Boginski, Sergiy Butenko, and Panos M Pardalos. Mining market data: a network approach. *Computers & Operations Research*, 33(11):3171–3184, 2006. 30

[64] Béla Bollobás. *Random graphs*. Springer, 1998. 38

[65] Phillip Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2(1):113–120, 1972. 28

[66] John Adrian Bondy and Uppaluri Siva Ramachandra Murty. *Graph theory with applications*, volume 290. Macmillan London, 1976. 2

[67] Stephen P Borgatti. Centrality and network flow. *Social networks*, 27(1):55–71, 2005. 26

[68] Stephen P Borgatti and Martin G Everett. A graph-theoretic perspective on centrality. *Social networks*, 28(4):466–484, 2006. 26

[69] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998. 29, 77, 81

[70] David V Budescu and Eva Chen. Identifying expertise to extract the wisdom of crowds. *Management Science*, 2014. 109

[71] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-mac: A short preamble mac protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 307–320, New York, NY, USA, 2006. ACM. 47

[72] Ciro Cattuto, Wouter Van den Broeck, Alain Barrat, Vittoria Colizza, Jean-François Pinton, and Alessandro Vespignani. Dynamics of person-to-person interactions from distributed rfid sensor networks. *PloS one*, 5(7):e11596, 2010. 30, 54, 60

[73] W. Chauvenet. *A Manual of Spherical and Practical Astronomy: Embracing the General Problems of Spherical Astronomy, the Special Applications to Nautical Astronomy, and the Theory and Use of Fixed and Portable Astronomical Instruments, with an Appendix on the Method of Least Squares*. A Manual of Spherical and Practical Astronomy. J. B. Lippincott & Company, 1863. 119

[74] Gal Chechik, Eugene Oh, Oliver Rando, Jonathan Weissman, Aviv Regev, and Daphne Koller. Activity motifs reveal principles of timing in transcriptional control of the yeast metabolic network. *Nature biotechnology*, 26(11):1251–1259, 2008. 30

[75] Judith A Chevalier and Dina Mayzlin. The effect of word of mouth on sales: Online book reviews. *Journal of marketing research*, 43(3):345–354, 2006. 3

[76] Alvin Chin, Bin Xu, Hao Wang, and Xia Wang. Linking people through physical proximity in a conference. In *Proceedings of the 3rd international workshop on Modeling social media*, MSM '12, pages 13–20. ACM, 2012. 52

[77] Tanzeem Choudhury, Sunny Consolvo, Beverly Harrison, Jeffrey Hightower, Anthony LaMarca, Louis LeGrand, Ali Rahimi, Adam Rea, G Bordello, Bruce Hemingway, et al. The mobile sensing platform: An embedded activity recognition system. *Pervasive Computing, IEEE*, 7(2):32–41, 2008. 50

[78] Tanzeem Choudhury and Alex Pentland. Sensing and modeling human networks using the sociometer. In *2012 16th International Symposium on Wearable Computers*, pages 216–216. IEEE Computer Society, 2003. 50

[79] Andrea Clementi and Francesco Pasquale. Information spreading in dynamic networks: An analytical approach. In *Theoretical Aspects of Distributed Computing in Sensor Networks*, pages 591–619. Springer, 2011. 3, 79

[80] Marco Conti, Silvia Giordano, Martin May, and Andrea Passarella. From opportunistic networks to opportunistic computing. *Communications Magazine, IEEE*, 48(9):126–139, 2010. 48, 53

[81] Corinna Cortes, Daryl Pregibon, and Chris Volinsky. Computational methods for dynamic graphs. *Journal of Computational and Graphical Statistics*, 12(4), 2003. 12, 32

[82] Abhimanyu Das, Sreenivas Gollapudi, Rina Panigrahy, and Mahyar Salek. Debiasing social wisdom. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 500–508, New York, NY, USA, 2013. ACM. 134

[83] Morris H DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974. 3, 7, 129, 130

[84] Ilker Demirkol, Cem Ersoy, and Fatih Alagoz. Mac protocols for wireless sensor networks: a survey. *Communications Magazine, IEEE*, 44(4):115–121, 2006. 53

[85] Wouter Van der Broeck, Ciro Cattuto, Alain Barrat, Martin Szomszor, Gianluca Correndo, and Harith Alani. The live social semantics application: a platform for integrating face-to-face presence with on-line social networking. In *First International Workshop on Communication, Collaboration and Social Networking in Pervasive Computing Environments (PerCol 2010)*, Apr 2010. 52

[86] Reinhard Diestel. *Graph Theory* {*Graduate Texts in Mathematics; 173*}. Springer-Verlag Berlin and Heidelberg GmbH & Company KG, 2000. 23

[87] Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268, 2006. 2, 30, 98

[88] J-P Eckmann, S Oliffson Kamphorst, and David Ruelle. Recurrence plots of dynamical systems. *EPL (Europhysics Letters)*, 4(9):973, 1987. 36

[89] Victor M Eguiluz, Dante R Chialvo, Guillermo A Cecchi, Marwan Baliki, and A Vania Apkarian. Scale-free brain functional networks. *Physical review letters*, 94(1):018102, 2005. 30

[90] Amre El-Hoiydi and J-D Decotignie. WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In *Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium on*, volume 1, pages 244–251. IEEE, 2004. 48

[91] Nicole B Ellison et al. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, 2007. 1

[92] Nicole B Ellison, Charles Steinfield, and Cliff Lampe. The benefits of facebook "friends:" social capital and college students' use of online social network sites. *Journal of Computer-Mediated Communication*, 12(4):1143–1168, 2007. 1

[93] P. Erdős and A. Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960. 38, 69

[94] P. Erdős and A. Rényi. On random graphs. I. *Publ. Math. Debrecen*, 6:290–297, 1959. 38

[95] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pages 27–34. ACM, 2003. 49

[96] Eugene F Fama. Efficient capital markets: A review of theory and empirical work*. *The journal of Finance*, 25(2):383–417, 1970. 109

[97] Thomas S Ferguson. On the rejection of outliers. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 253–287. University of California Press Berkeley, 1961. 110

[98] Francesco Ficarola. OpenBeacon Logger. https://github.com/francesco-ficarola/OpenBeaconLogger, 2013. 51

[99] Francesco Ficarola. OpenBeacon Parser. https://github.com/francesco-ficarola/OpenBeaconParser, 2013. 59

[100] Herbert Fleischner. The square of every two-connected graph is hamiltonian. *Journal of Combinatorial Theory, Series B*, 16(1):29–34, 1974. 29

[101] Dániel Fogaras, Balázs Rácz, Károly Csalogány, and Tamás Sarlós. Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments. *Internet Mathematics*, 2(3):333–358, 2005. 77, 81

[102] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977. 27

[103] Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1979. 26

[104] John RP French Jr. A formal theory of social power. *Psychological review*, 63(3):181, 1956. 129

[105] Noah E Friedkin and Eugene C Johnsen. Social-influence and opinions. *Journal of Mathematical Sociology*, 15(3-4):193–205, 1990. 129, 133

[106] Noah E Friedkin and Eugene C Johnsen. Social influence networks and opinion change. *Advances in Group Processes*, 16(1):1–29, 1999. 129

[107] Georg Ferdinand Frobenius, Ferdinand Georg Frobenius, Ferdinand Georg Frobenius, and Ferdinand Georg Frobenius. *Über Matrizen aus nicht negativen Elementen*. Königliche Akademie der Wissenschaften, 1912. 28

[108] Thomas MJ Fruchterman and Edward M Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991. 60

[109] Francis Galton. The ballot-box. *Nature*, 75:509–510, 1907. 3, 107

[110] Francis Galton. Vox populi. *Nature*, 75:450–451, 1907. 3, 107

[111] Laetitia Gauvin, André Panisson, and Ciro Cattuto. Detecting the community structure and activity patterns of temporal networks: A non-negative tensor factorization approach. *PLoS ONE*, 9(1):e86028, 01 2014. 52

[112] Edgar N Gilbert. Random graphs. *The Annals of Mathematical Statistics*, pages 1141–1144, 1959. 38

[113] Benjamin Golub and Matthew O Jackson. Naive learning in social networks and the wisdom of crowds. *American Economic Journal: Microeconomics*, 2(1):112–149, 2010. 129

[114] Frank Harary. Status and contrastatus. *Sociometry*, pages 23–43, 1959. 129

[115] Christopher G Harris. The beauty contest revisited: measuring consensus rankings of relevance using a game. In *Proceedings of the First International Workshop on Gamification for Information Retrieval*, pages 17–21. ACM, 2014. 110

[116] Shawndra Hill, Deepak K Agarwal, Robert Bell, and Chris Volinsky. Building an effective representation for dynamic networks. *Journal of Computational and Graphical Statistics*, 15(3), 2006. 12, 32

[117] Robin M Hogarth. A note on aggregating opinions. *Organizational Behavior and Human Performance*, 21(1):40–46, 1978. 107

[118] Petter Holme and Jari Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012. 3, 12, 30

[119] C Hsieh, Christopher Moghbel, Jianhong Fang, and Junghoo Cho. Experts vs the crowd: Examining popular news prediction perfomance on twitter. In *Proceedings of the WWW13 conference, Rio de Janeiro*, 2013. 109

[120] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 244–251. ACM, 2005. 2, 45, 50

[121] Kurtis Kredo II and Prasant Mohapatra. Medium access control in wireless sensor networks. *Computer Networks*, 51(4):961–994, 2007. 47

[122] Lorenzo Isella, Mariateresa Romano, Alain Barrat, Ciro Cattuto, Vittoria Colizza, Wouter Van den Broeck, Francesco Gesualdo, Elisabetta Pandolfi, Lucilla Ravá, Caterina Rizzo, and Alberto Eugenio Tozzi. Close encounters in a pediatric ward: Measuring face-to-face proximity and mixing patterns with wearable sensors. *PLoS ONE*, 6(2):e17144, 02 2011. 52

[123] Matthew O Jackson. *Social and economic networks*. Princeton University Press, 2010. 11, 23

[124] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PLoS ONE*, 9(6):e98679, 06 2014. 21, 61

[125] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM, 2007. 2

[126] Dinesh Babu Jayagopi, Taemie Kim, Alex (Sandy) Pentland, and Daniel Gatica-Perez. Recognizing conversational context in group interaction using privacy-sensitive mobile sensors. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, MUM '10, pages 8:1–8:4, New York, NY, USA, 2010. ACM. 50

[127] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, pages 271–279. ACM, 2003. 81

[128] Ruoming Jin, Scott Mccallen, C Liu, Y Xiang, E Almaas, and XH Zhou. Identify dynamic network modules with temporal and spatial constraints. In *Pacific Symposium on Biocomputing*, 2009. 30

[129] K.D. Joshi. *Foundations of Discrete Mathematics*. Wiley, 1989. 14

[130] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953. 26, 129

[131] David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 504–513. ACM, 2000. 34

[132] Andrew J King, Lawrence Cheng, Sandra D Starke, and Julia P Myatt. Is the true 'wisdom of the crowd' to copy successful individuals? *Biology letters*, page rsbl20110795, 2011. 6, 108, 110

[133] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170. ACM, 2000. 41

[134] Leonard Kleinrock and Fouad A Tobagi. Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics. *Communications, IEEE Transactions on*, 23(12):1400–1416, 1975. 46

[135] D.E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. ACM Press, 2009. 12

[136] KONECT. Facebook wall posts. http://konect.uni-koblenz.de/networks/facebook-wosn-wall. 101

[137] David Kotz and Tristan Henderson. Crawdad: A community resource for archiving wireless data at dartmouth. *Pervasive Computing, IEEE*, 4(4):12–14, 2005. 49

[138] Oleksii Kuchaiev and Natasa Przulj. Learning the structure of protein-protein interaction networks. In *Pacific Symposium on Biocomputing*, volume 14, pages 39–50. Citeseer, 2009. 30

[139] Koen Langendoen and Andreas Meier. Analyzing mac protocols for low data-rate applications. *ACM Trans. Sen. Netw.*, 7(1):10:1–10:34, August 2010. 48

[140] Amy N Langville and Carl D Meyer. Deeper inside pagerank. *Internet Mathematics*, 1(3):335–380, 2004. 29, 75, 77, 81, 88

[141] Amy N Langville and Carl D Meyer. A survey of eigenvector methods for web information retrieval. *SIAM review*, 47(1):135–161, 2005. 75, 77

[142] Jaron Lanier. Digital maoism. *The Edge. org*, 2006. 108

[143] Benoît Latré, Bart Braem, Ingrid Moerman, Chris Blondia, and Piet Demeester. A survey on wireless body area networks. *Wirel. Netw.*, 17(1):1–18, January 2011. 49

[144] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187. ACM, 2005. 83, 101

[145] Kevin Lewis, Jason Kaufman, Marco Gonzalez, Andreas Wimmer, and Nicholas Christakis. Tastes, ties, and time: A new social network dataset using facebook. com. *Social networks*, 30(4):330–342, 2008. 2

[146] Wei Liu, Andrey Kan, Jeffrey Chan, James Bailey, Christopher Leckie, Jian Pei, and Ramamohanarao Kotagiri. On compressing weighted time-evolving graphs. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2319–2322. ACM, 2012. 12, 29, 32

[147] Jan Lorenz, Heiko Rauhut, Frank Schweitzer, and Dirk Helbing. How social influence can undermine the wisdom of crowd effect. *Proceedings of the National Academy of Sciences*, 108(22):9020–9025, 2011. 3, 6, 108, 109, 123

[148] R Duncan Luce and Albert D Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116, 1949. 15

[149] Spyros Makridakis and Robert L Winkler. Averages of forecasts: Some empirical results. *Management Science*, 29(9):987–996, 1983. 107

[150] Norbert Marwan, M Carmen Romano, Marco Thiel, and Jürgen Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, 438(5):237–329, 2007. 36

[151] Pavlin Mavrodiev, Claudio J Tessone, and Frank Schweitzer. Effects of social influence on the wisdom of crowds. *arXiv preprint arXiv:1204.3463*, 2012. 6, 109, 110

[152] Carl D Meyer. *Matrix analysis and applied linear algebra*. Siam, 2000. 137

[153] Othon Michail. An introduction to temporal graphs: An algorithmic perspective. *arXiv preprint arXiv:1503.00278*, 2015. 12

[154] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967. 41

[155] RV Mises and Hilda Pollaczek-Geiringer. Praktische verfahren der gleichungsauflösung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 9(1):58–77, 1929. 28

[156] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42. ACM, 2007. 2

[157] David Moss and Philip Levis. Box-macs: Exploiting physical and link layer boundaries in lowpower networking. Technical report, 2008. 48

[158] Mirco Musolesi and Cecilia Mascolo. Mobility models for systems evaluation. In Benoît Garbinato, Hugo Miranda, and Luís Rodrigues, editors, *Middleware for Network Eccentric and Mobile Applications*, pages 43–62. Springer Berlin Heidelberg, 2009. 49

[159] Jerome L Myers, Arnold Well, and Robert Frederick Lorch. *Research design and statistical analysis*. Routledge, 2010. 36

[160] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006. 26

[161] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004. 26

[162] Vincenzo Nicosia, John Tang, Mirco Musolesi, Giovanni Russo, Cecilia Mascolo, and Vito Latora. Components in time-varying graphs. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(2):023101, 2012. 35

[163] Jason W Osborne and Amy Overbay. The power of outliers (and why researchers should always check for them). *Practical assessment, research & evaluation*, 9(6):1–12, 2004. 110

[164] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1999. 3, 27, 29, 75, 77, 81

[165] E.M Palmer and A.J Schwenk. On the number of trees in a random forest. *Journal of Combinatorial Theory, Series B*, 27(2):109–121, 1979. 14

[166] Raj Kumar Pan and Jari Saramäki. Path lengths, correlations, and centrality in temporal networks. *Physical Review E*, 84(1):016105, 2011. 35

[167] Vilfredo Pareto. *Cours d'économie politique*. Librairie Droz, 1964. 42

[168] Josiane Xavier Parreira, Debora Donato, Sebastian Michel, and Gerhard Weikum. Efficient and decentralized PageRank approximation in a peer-to-peer web search network. In *Proceedings of the 32nd international conference on Very large data bases*, pages 415–426. VLDB Endowment, 2006. 3, 76, 78

[169] Josiane Xavier Parreira, Sebastian Michel, Matthias Bender, Tom Crecelius, and Gerhard Weikum. P2P authority analysis for social communities. In *Proceedings of the 33rd international conference on Very large data bases*, pages 1398–1401. VLDB Endowment, 2007. 76

[170] Karl Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58(347-352):240–242, 1895. 36, 120

[171] Adam Perer and Ben Shneiderman. Balancing systematic and flexible exploration of social networks. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):693–700, 2006. 21

[172] Nicola Perra and Santo Fortunato. Spectral centrality measures in complex networks. *Physical Review E*, 78(3):036107, 2008. 29

[173] Oskar Perron. Zur theorie der matrices. *Mathematische Annalen*, 64(2):248–263, 1907. 28

[174] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 95–107, New York, NY, USA, 2004. ACM. 47

[175] John Prpic, Piper Jackson, and Thai Nguyen. A computational model of crowds for collective intelligence. *Prpić, J., Jackson, P., & Nguyen*, 2014. 3

[176] Liangfei Qiu, Huaxia Rui, and Andrew B Whinston. Effects of social networks on prediction markets: Examination in a controlled experiment. *Journal of Management Information Systems*, 30(4):235–268, 2014. 110

[177] Injong Rhee, Ajit Warrier, Mahesh Aia, Jeongki Min, and Mihail L Sichitiu. Z-mac: a hybrid mac for wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 16(3):511–524, 2008. 48

[178] Atish Das Sarma, Anisur Rahaman Molla, Gopal Pandurangan, and Eli Upfal. Fast distributed PageRank computation. In *Distributed Computing and Networking*, pages 11–26. Springer, 2013. 76, 78, 89, 96, 97

[179] Eugene Seneta. *Non-negative matrices and Markov chains*. Springer, 2006. 29, 132, 140, 151

[180] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003. 22

[181] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996. 21

[182] SNAP. CAIDA AS Relationships Datasets. http://snap.stanford.edu/data/as-caida.html. 101

[183] SNAP. High-Energy Physics citation network. http://snap.stanford.edu/data/cit-HepPh.html. 101

[184] Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101, 1904. 36

[185] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Regis, Bruno Lina, and Philippe Vanhems. High-resolution measurements of face-to-face contact patterns in a primary school. *PLoS ONE*, 6(8):e23176, 08 2011. 52

[186] G. Strang. *Linear Algebra and Its Applications*. Thomson, Brooks/Cole, 2006. 14

[187] James Surowiecki. *The Wisdom of Crowds*. Anchor Books, 2004. 3, 6, 54, 58, 63, 107, 108, 109

[188] Sunil Taneja and Ashwani Kush. A survey of routing protocols in mobile ad hoc networks. *International Journal of Innovation, Management and Technology*, 1(3):2010–0248, 2010. 53

[189] John Tang, Salvatore Scellato, Mirco Musolesi, Cecilia Mascolo, and Vito Latora. Small-world behavior in time-varying graphs. *Physical Review E*, 81(5):055101, 2010. 35

[190] Sana Ullah, Henry Higgins, Bart Braem, Benoit Latre, Chris Blondia, Ingrid Moerman, Shahnaz Saleem, Ziaur Rahman, and Kyung Sup Kwak. A comprehensive survey of wireless body area networks. *J. Med. Syst.*, 36(3):1065–1094, June 2012. 49

[191] Tijs van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, pages 171–180, New York, NY, USA, 2003. ACM. 47

[192] Stijn Van Dongen and Anton J Enright. Metric distances derived from cosine similarity and pearson and spearman correlations. *arXiv preprint arXiv:1208.3145*, 2012. 85

[193] Philippe Vanhems, Alain Barrat, Ciro Cattuto, Jean-François Pinton, Nagham Khanafer, Corinne Régis, Byeul-a Kim, Brigitte Comte, and Nicolas Voirin. Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. *PLoS ONE*, 8(9):e73970, 09 2013. 3, 52

[194] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 37–42. ACM, 2009. 83, 101

[195] Nicolas Voirin, Cécile Payet, Alain Barrat, Ciro Cattuto, Nagham Khanafer, Corinne Régis, Byeul Kim, Brigitte Comte, Jean-Sébastien Casalegno, Bruno Lina, et al. Combining high-resolution contact data with virological data to investigate influenza transmission in a tertiary care hospital. *Infection Control & Hospital Epidemiology*, pages 1–7, 2015. 3, 52

[196] FrankEdward Walter, Stefano Battiston, and Frank Schweitzer. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16(1):57–74, 2008. 45

[197] Yuan Wang and David J DeWitt. Computing PageRank in a distributed internet search system. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 420–431. VLDB Endowment, 2004. 3, 78

[198] Stanley Wasserman. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994. 2, 11, 23

[199] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-world networks. *nature*, 393(6684):440–442, 1998. 25, 40, 41

[200] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270. ACM, 2010. 76

[201] David J. Wheeler and Roger M. Needham. TEA, a Tiny Encryption Algorithm. In *FSE*, pages 363–366, 1994. 53

[202] Hongyi Wu and Yi Pan. *Medium Access Control in Wireless Networks*. Nova Science Publishers, 2008. 45, 46

[203] B Bui Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(02):267–285, 2003. 34

[204] Elias Yarrkov. Cryptanalysis of XXTEA. Cryptology ePrint Archive, Report 2010/254, 2010. http://eprint.iacr.org/. 53

[205] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(3):493–506, June 2004. 47

[206] Sheng Kung Michael Yi, Mark Steyvers, Michael D Lee, and Matthew J Dry. The wisdom of the crowd in combinatorial problems. *Cognitive science*, 36(3):452–470, 2012. 109

[207] Han Yu, Zhiqi Shen, Chunyan Miao, C. Leung, and D. Niyato. A survey of trust and reputation management systems in wireless communications. *Proceedings of the IEEE*, 98(10):1755–1772, Oct 2010. 45

[208] George Kingsley Zipf. Human behavior and the principle of least effort. 1949. 42