UNIVERSITÀ DI TRENTO

Dipartimento di Matematica

DEPARTMENT OF MATHEMATICS

MASTER DEGREE IN MATHEMATICS

Analyzing and modelling the temporal network of Telegram's chats

Supervisor

Claudio Agostinelli

Signature:

Candidate

Aurora Vindimian

Co-supervisor

Thomas Louf

Riccardo Gallotti

Academic Year: 2023-2024

Date of discussion: 18th October 2024

Contents

Introduction

Human behaviour has long been an object of study for many researchers. Social network science focuses on the study of interactions between individuals, including real-life friendships, sexual contacts, scientific paper citations and, more recently, online interactions. In the last decades, the success of social media has significantly increased the volume of online human interactions. This phenomenon has made available a huge quantity of data concerning human activities, expanding our knowledge of social dynamics and its underlying mechanisms. The availability of data from these platforms has enabled researchers to study humans in a wider and broader way than did with real-life interactions. Indeed, these apps provide wider datasets recording additional information too, such as the timing or the content of an interaction, for every user. This allows for larger and more precise studies by having bigger and more heterogeneous samples. Every platform differs from others by the services it provides which implies having different types of users and ways of interacting. A study of a specific app enables one not only to characterize how people behave on it but also to compare it with studies on other social media to identify recurrent patterns. At the same time, when working, for instance, with retweets on Twitter or messages on WhatsApp, the interactions are characterized by the flow of information. This allows for the study of phenomena such as information spread and misinformation.

The interest and main focus of this thesis regards the study of forwarded messages on Telegram. Telegram is a messaging app which offers users the possibility to create channels. A channel is a group chat where only admins can post, allowing to share information quickly to a broad audience. It can be created by any Telegram user and others can subscribe or join. If it is public, then the chat's content can be seen on the Web by non Telegram users too. Each channel can have an associated discussion chat where participants can interact and comment channel's posts. Telegram, additionally, offers a high privacy level, which has increased its use across users. The high level of security that this app ensures and the impunity it provides to users, has increased the usage across extremists and misinformation groups. Indeed, in the name of freedom of speech, Telegram does not moderate the content sent on the app, allowing users to share, in some cases, illegal material too. Recently, the way of doing of the app has been questioned, culminating with the arrest of its founder. In the recent years, Telegram has been the object of various studies for misinformation spreading on the app [5, 29]. Others have focused, instead, on the structure of specific communities on it, such as the UK far-right, as done in [15]. However, despite its growing popularity, this app has not been studied as much as older platforms, such as Twitter or Facebook.

In this thesis we are going to consider forwarded messages as interactions between chats. Forwards are thought to be correlated with the diffusion of misinformation. As a result, some social platforms have started to introduce measures to limit it. On WhatsApp, for instance, you can forward a message to at most 5 users at the same time. The study of interactions and timings becomes then fundamental in understanding and preventing this phenomenon. In [12], the authors analyzed data from WhatsApp to see whether the measure introduced was effective in blocking the diffusion of misinformation. They discovered that limiting the forwards delays the propagation of fake news but are unsuccessful in blocking it. In [8], the authors tested different strategies to mitigate disinformation on networks generated by known models or real ones. Thus, creating a model that reproduces forwards on Telegram allows, in the future, to study the efficacy of measures to combat disinformation on the social media. At the same time, the study of forwards allows not only the development of a model, but to discover also how people behave on these apps. Many social interactions have been studied through the years, showing consistent properties, such as heterogeneous number of connections [2], small-world dynamics [41] and reinforcement of old ties [40]. Some models have been developed to reproduce specific patterns of the network, an example is the Barabasi-Albert model which generates scale-free networks. Oftentimes though, classical models such as this one are not able to fully capture all the properties of a real network, thus a more specific study of the data and the mechanisms behind the process has to be performed. Employing network science in social studies provides a structure to effectively exploit relationships among components, but simplifying the network as static is myopic, as it does not portray the real on-going process behind the data. Indeed, the network structure may change over time, showing different properties depending on the time window. At the same time, temporal aspects can tell us a lot about humans too. As in the network science framework, humans show consistent patterns in real datasets, where the timing of their interactions is often found correlated and heterogeneous [21]. To reproduce the heavy tails in timings many models have been proposed. A famous one is the activity model [34], which is able to recreate heavy tails in the distribution of inter-event times, but fails to reproduce correlation. By assuming independence between a user's actions, it fails to capture the real mechanism that generated the data. To discover the driving forces of the studied process, it is necessary to analyze at first the data as a network and then focus on the temporal aspects.

The thesis is organized into five chapters. Chapter 1 discusses the problem to face and presents the dataset with some exploratory analyses. Then, in Chapter 2, methods of temporal network science are introduced and reviewed. In Chapter 3, the temporal network is analyzed using methods presented in the previous chapter. In Chapter 4, the models to reproduce Telegram's data are defined and the results obtained from simulations are shown. Finally, in Chapter 5, we briefly summarize the results obtained in the thesis suggesting possible directions for future works.

CHAPTER 1

Problem and dataset overview

The aim of this thesis is to develop a model that can recreate the forwards of messages between chats of Telegram. The problem is viewed under two perspectives: structure and timings of interactions between chats. Both aspects are key to recreate such phenomenon and to design accurate simulations. In the future, this model could be used to study the correlation with misinformation spreading and message forwards, but not only. Another objective of this study is to gain knowledge regarding human behaviour on these apps. For instance, how they interact on this online world, if they are in contact with chats speaking the same language and if channels aggregate in communities. Regarding the timing, we will try to discover if there is correlation between the time at which we forward messages and whether it is influenced by daily rhythms.

In this first chapter, a brief presentation and analysis of the dataset of interest is made. The dataset used in this thesis is the so-called ”Pushshift Telegram Dataset” by Baumgartner et al. [4]. This data has been collected starting from around 250 channels of Telegram. The majority are English-speaking, the main topic of 124 of them is right-wing politics, while the remaining ones are cryptocurrency-related. All messages sent in these channels, from their creation to the date of the retrieval, have been collected. Then, using a snowballing procedure, if one of these messages has been forwarded from a chat that was not already in the dataset, that chat is added to the list. They continued this procedure until they reached 27 801 channels. The messages are retrieved not only for the channel's chat itself but also for the associated other chat, such as the discussion chat. In particular, for the 27 801 channels studied, 1 860 have a related discussion chat.

The data provided was unstructured, nested JSON data and divided into 2 parts: channels and messages. Both of them contained additional information, for channels we had various properties such as whether it was a possible scam, the number of participants, the creation time, whether it was verified or not, the description, the presence of a discussion chat and its properties, and much more. For messages, the provided information was, for instance, the sending date, the type of media attached, the text, whether it has been forwarded, from who it has been forwarded and many others.

1.1 Channel

This subsection focuses on the channel dataset and some exploratory studies of its features. Given the high number of features provided, a first analysis has been performed to choose and study the most interesting and relevant ones. So, after translating the raw data into a data frame, a feature selection was made. To understand the meaning of every property given, the Telegram API documentation [38] has been used as a reference. Some properties that were kept were, for instance, the title of the chat, its identification number, the description, the creation date and some more. This procedure has generated a reduced dataset with some properties which are summed up in the following table.

Type	**Total**	**Scam**	**Verified**
Channel	27 801	4	47
Discussion chat	1 860	0	0

Table 1.1: Statistics of discussion chats and channels: the table shows how many chats are possible scams and verified.

Focusing on the date parameter, the first chat in the dataset was created on September 18th, 2015, and the last one on October 2nd, 2019. All the times are specified with a resolution of one second. Figure 1.1 provides an idea of the distribution of creation times of the channels and discussion chats. Telegram introduced channels as a feature for the first time in September 2015 [39], which can explain the peak in the plot around the first months. Discussion chats, instead, were introduced in 2019, which again explains the peak in that year. Note that some discussion chats were born before 2019, this may be because some chats were created and later on converted into discussions.

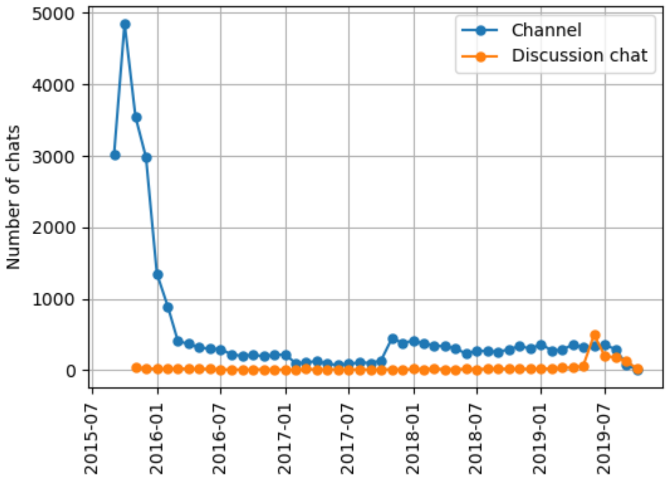


Figure 1.1: The number of channels (blue) and discussion chats (orange) created per month. The underlying grid corresponds to a 6 months range.

Another interesting feature to analyze is the number of participants in a channel.

The minimum is 0 while the maximum is of the order of 10^6. The number of participants varies across a wide range as can be seen in Figure 1.2.

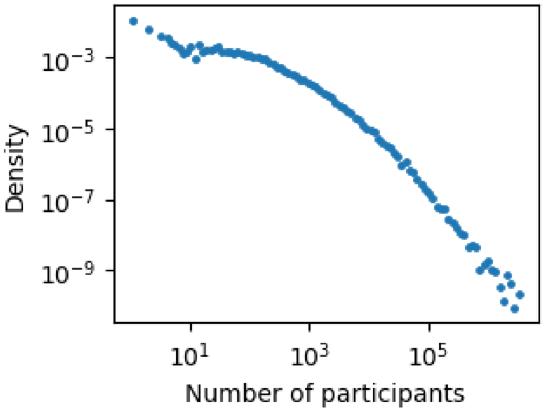


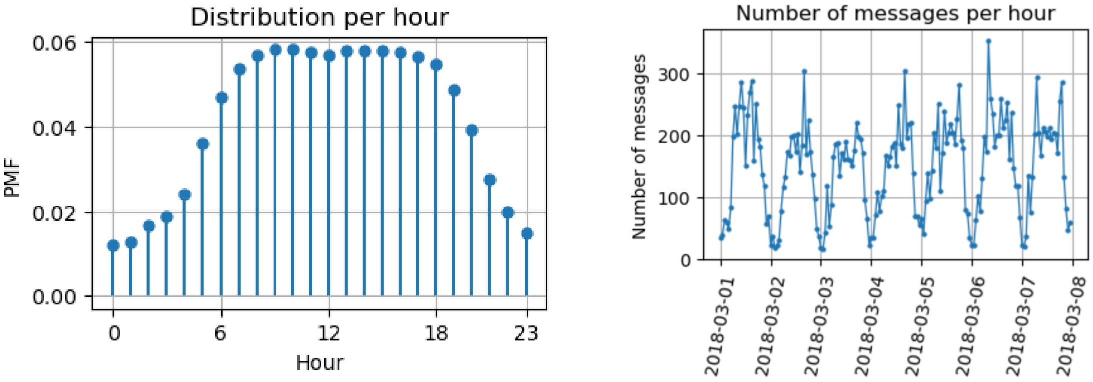
Figure 1.2: Probability density function of the number of participants in each channel.

1.2 Message

This subsection focuses on the message dataset and some exploratory studies of its features. For the aim of this thesis, the main interest is on forwarded messages, which are going to form the links between the nodes of the network. Out of the totality of chats with collected messages, 26 537 of them contain forwarded messages from other chats. Furthermore, the number of chats that have forwarded a message from the studied chats is actually quite lower, in particular 22 650. In the following parts of the thesis, the network corresponding to these studied chats is going to be constructed, so a network of 29 661 nodes at most. In this setting, just the connected component is going to be kept. This choice will be better explained in the next chapter. This filtering reduces the number of events, i.e. number of forwarded messages, from the total initial of 17 932 793 to 7 500 509. In the following part of this section, the data considered is going to be filtered in order to be consistent with respect to the next chapters.

Looking more in-depth into some features of messages, the date at which messages are sent is particularly informative. First of all, the first forwarded message of the filtered dataset was sent on the 19th of September 2015, while the last one on the 6th of November 2019. Looking at the count of messages per hour of the day in Figure 1.3, it is immediate to see that the sending date of the message is dominated by circadian patterns. Weekly patterns does not seem to appear, as can be seen in Figure 1.4(a), which is confirmed by investigating different time windows. An example is shown in panel (b) of Figure 1.4.

Finally, the monthly occurrences of messages have a clear initial increase and final drop, as can be seen in Figure 1.5. Regarding the initial low values, the first chats, as mentioned in Section 1.1, were created in mid-September 2015. Consequently, the messages dataset misses many days of September 2015. In the first months, more channels are created, thus the number of forwarded messages slightly increases. Regarding the final times, the drop is due to the retrieval date of the data, which started in November. Unlike daily rhythms, there seems to be no yearly pattern. Overall, an increasing function would have been expected, because, even if in small part, every month the number of new channels created is positive as seen in Figure 1.1. Still, a sort of stabilization seems to appear, which can be explained intuitively by the fact that some channels created in 2015 are not used much in 2018 for example, so the number of new channels created is roughly equal to the number of deprecated ones. However, in the end, there is a clear peak around August and September 2019. This peak can be explained by the introduction in Telegram of discussion chats [39]. Indeed, many of them automatically forward content from the main channel, which raises the number of forwards. This is confirmed by the fact that from June 2019, 66.1% of forwarded messages are sent by discussion chats.



(a)

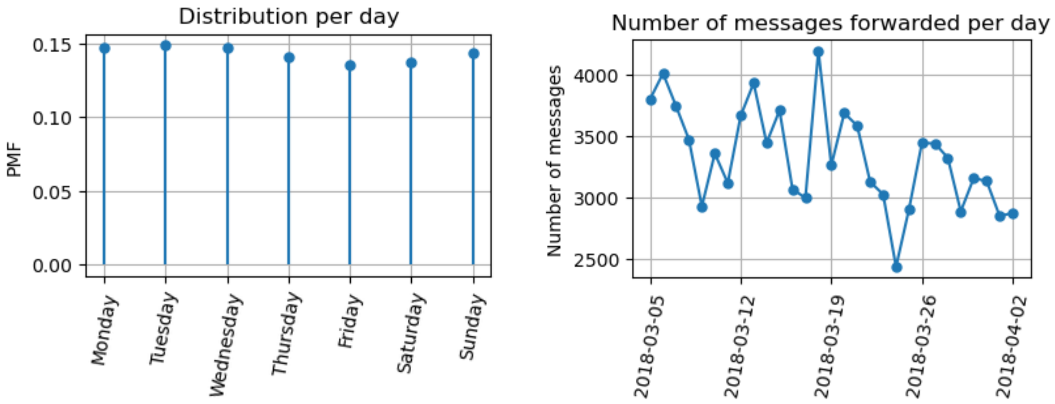
(b)

Figure 1.3: Study of the daily patterns: (a) PMF of messages per hour in the filtered dataset. (b) Count of messages sent in an arbitrary window. The grid represents 1 day intervals.

1.3 Language

Human relationships are strongly driven by the language of speakers. Clearly, given a group of people, it is easier for speakers of the same language to bond and interact than with others. For these reasons, assigning each chat to a language will not only allow us to discover which idioms are more present on Telegram, but also to investigate the existence of language homophily too.

To recognize the language, every text message sent in a chat is analyzed. Additionally, if present, the description of the chat, the so-called about, is considered too. All these strings might contain entities so they were cleaned by removing hashtags, mentions and URLs. To recognize languages, the Python library lingua [36] has been employed. Then the language of each message is detected where the recognition is kept if its confidence is at least 0.5. After completing these steps for every string, the most recognized language across the messages is set to be the language



(a)

(b)

Figure 1.4: Study of the weekly patterns: (a) PMF of forwarded messages per day. The grid represents a one-day range. (b) Count of messages sent in an arbitrary window. The grid represents 1 week intervals.

of the chat. If no text message is present in a chat, then the language is set to unknown. In Figure 1.6 it is possible to see the language distribution across chats and messages, respectively. The latter is meant as follows, a message is considered of a certain language, for instance Italian, if it has been sent in an Italian channel. The language distribution across chats is very different from messages, meaning that chats of some languages, such as Hebrew for instance, are not many, but they are very active.

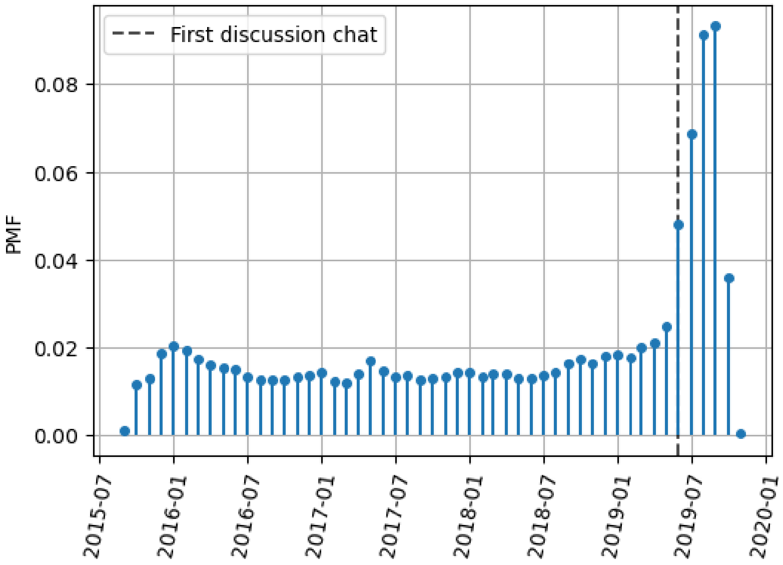
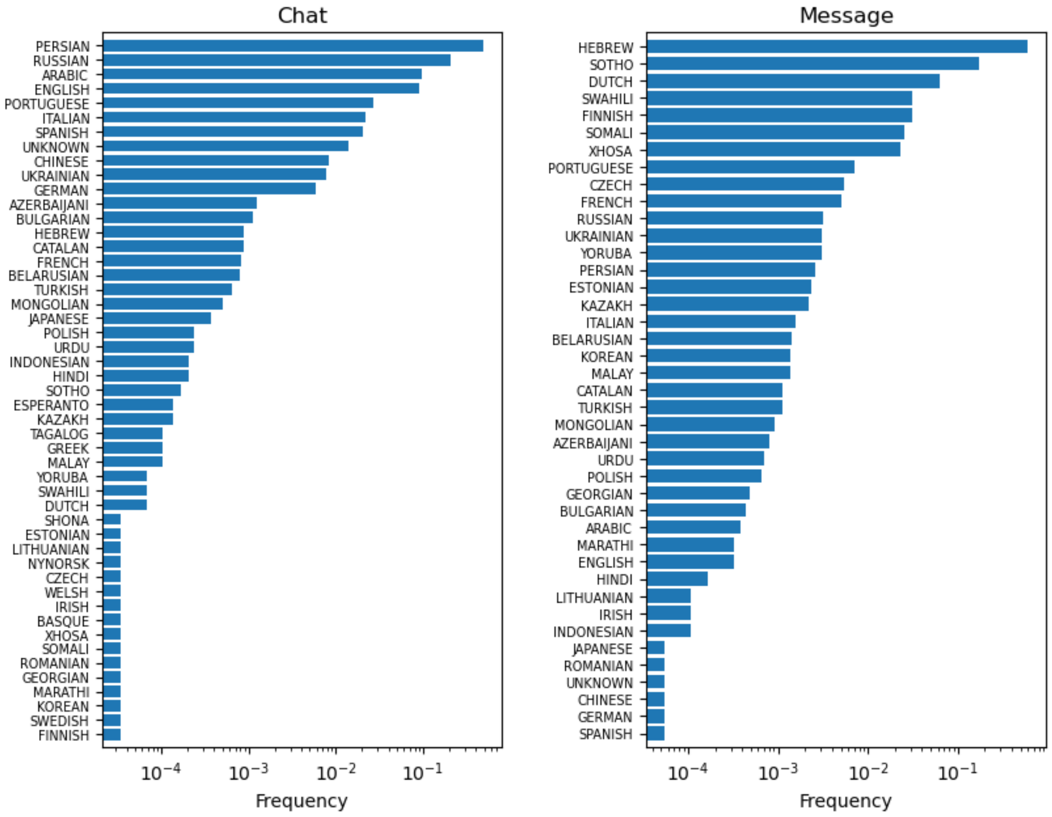


Figure 1.5: PMF of forwarded messages per month. The grid represents a six-month range. A peak appears corresponding to the introduction of discussion chats on Telegram.



(a)

(b)

Figure 1.6: Distribution of languages across (a) chats and (b) messages.

CHAPTER 2

Methods

To address the objectives stated in the previous chapter, some background notions are needed. This chapter contains definitions, properties and observations which can be used to analyze the Telegram dataset. While the first part concerns general useful quantities, the following two contain the most important information, regarding, respectively, network science and time series.

2.1 Statistical methods

In this section, fat-tailed distributions and Bayesian optimization are presented. These objects are going to be used in the following chapters.

2.1.1 Fat-tailed distribution

In general, we refer to fat-tailed, long-tailed or heavy-tailed distributions, as distributions that at large values decay slower than an exponential one. These shapes may appear in a variety of real datasets, such as the population of a city or the intensity of earthquakes [27]. Other quantities, such as the height of people, are peaked around their means, meaning that the fluctuations of such variables are possible but limited around that value. In contrast, in fat-tailed distributions, the sample mean is not representative of the variable's value, because it varies across a wider range. The population of cities, for instance, can vary a lot. In particular, it has a broad tail, meaning that there exists a small but consistent number of cities with populations much larger than the mean by several orders of magnitude. These shapes are recurrent in network science too, especially in the form of a power law, for this reason, a brief introduction is made [1, 3, 9].

Definition 2.1.1. We say that a random variable X follows a power law if its probability density function has the following shape:

$$p(x) \propto x^{-\gamma}$$

where γ is called the exponent. Note that in many real cases, x follows such a form for $x \geq x_{min}$.

The definition can be extended to a discrete X and, in this case, the normalization constant is the zeta function:

$$C = \frac{1}{\zeta(\gamma, x_{min})} = \frac{1}{\sum_{n=0}^{\infty}(n + x_{min})^{-\gamma}}$$

while in the continuous case:

$$C = \frac{\gamma - 1}{x_{min}^{-\gamma+1}}$$

The parameter γ can be estimated via maximum likelihood estimator (MLE) or minimizing the Kolmogorov-Smirnov distance between the true distribution and the empirical one, which can be done with the powelaw package [1]. x_{min} can be estimated in the same way, however, in the majority of cases we have knowledge regarding its value. Power law distribution is often called scale-free because all moments of order $m \geq \gamma - 1$ diverge. So, if $\gamma = 2$, already the mean diverges.

2.1.2 Bayesian optimization

Bayesian optimization is a procedure of reinforcement learning to estimate parameters when optimizing a function f [7]. It is particularly useful when that function has no closed form of dependence on those parameters and when its computation is very costly. It can be used also to minimize functions, simply by considering $-f$.

Let f be the function to maximize with respect to some parameters $\mathbf{x}$. Initially, f is computed on a set of random values of $\mathbf{x}$. These points form the starting knowledge at our disposal $D_0 = \{(\mathbf{x}_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_M, f(\mathbf{x}_M))\}$. At each step, another value of $\mathbf{x}$ is explored and the corresponding f is observed. Then, the new observed pair $(\mathbf{x}_t, f(\mathbf{x}_t))$ is added to D_{t-1}, obtaining at step t $D_t = \{D_{t-1}, (\mathbf{x}_t, f(\mathbf{x}_t))\}$. It is assumed that $f(\mathbf{x})$ follows a Gaussian process with a certain mean and variance. Then, the posterior predictive distribution of $f(\mathbf{x}_{t+1})$ follows again a normal law:

$$P(f_{t+1}|D_t, \mathbf{x}_{t+1}) = N(\mu_t(\mathbf{x}_{t+1}), \sigma_t(\mathbf{x}_{t+1}))$$

When selecting the value of $\mathbf{x}$ to test at each step, we want both to explore areas in which we have little data and to test regions where we expect optimal values. We can try to achieve both through the information provided by the posterior predictive distribution. To balance the trade-off between exploitation and exploration, an acquisition function u which depends on $\mathbf{x}$ is introduced. Then, at each step we select $\mathbf{x}$ that maximizes u. Clearly, u should have a simpler form than f. The acquisition function used in this thesis is called upper confidence bound:

$$u(\mathbf{x}) = \mu_t(\mathbf{x}) + k\sigma_t(\mathbf{x})$$

where $\mu_t(\mathbf{x})$ and $\sigma_t(\mathbf{x})$ are, respectively, the posterior mean and standard deviation of the Gaussian process given the knowledge up to that step, while k is a constant to be fixed (by default $k = 2.576$). A high value of u can be achieved both by high $\sigma_t(\mathbf{x})$, which means high uncertainty in that area, and by high $\mu_t(\mathbf{x})$, obtained in regions where we expect high values of f. Thus, by maximizing u we favour both exploitation and exploration. This process can be stopped when the maximum observed value of f is stable for some iterations.

2.2 Elements of network science

In this section, some elements of network science are introduced. A network is constituted by nodes and edges which represent connections. The rise of network science started in the '90 with the development of theoretical and computational backgrounds. It became more and more useful through the years to represent complex systems, i.e. systems made by many entities that interact with each other. Using a network representation enables to study relationships and connections in a way that other frameworks are not able to. Network science has been successfully applied to different areas. Some instances are social interactions, such as friendships or sexual contacts, infrastructural networks, biological ones, such as neural connections or protein interactions, and information networks, such as citation of papers and the World Wide Web (WWW). All these fields deal with different topics and challenges, however, for each of them, the analysis of the interactions between the entities is fundamental. Applying network science to the network of sexual contact can, for instance, make us understand which types of structure in the graph enhance the diffusion of disease and devise prevention measures to limit it. In the following, an introduction to general concepts of network science is made. Two main references are used: a book written by Barabási et al. [3] and another book by Coscia [10]. Unless otherwise specified, in this and the following section the main reference is [3].

A network is a graph $G = (V, E)$ where V is the set of vertices or nodes, and E is the set of edges or links. It can be undirected or directed respectively if links are undirected or directed. In the same way, it is unweighted if the links have no weight associated to them, or equivalently, if they are all associated to a weight 1. Otherwise, it is called weighted and the weight associated to a link from node i to j is going to be denoted w_{ij}. If there exists a link between i and j, then they are said to be connected and it is denoted as $i \sim j$.

The structure of a network can be completely defined by the adjacency matrix A where $A_{ij} = w_{ij}$ if there is a link from node i to node j with weight w_{ij} and 0 otherwise. If the network is undirected, then A is symmetric. If it is unweighted then we denote $A_{ij} = a_{ij}$.

A path between two nodes is a sequence of connected vertices such that, starting from the first one, it is possible to reach the last one by traversing links. The length of a path is the number of traversed links. The shortest path between two nodes is the one with the lowest length, its length is then called the distance between the two nodes.

Network properties are studied using various measures which allow to identify certain behaviours. Those measures may differ depending on whether the network is directed and weighted, but they are defined in similar ways and with the same goal of identifying a specific behaviour. For this reason, the following section will be split into 2 subsections.

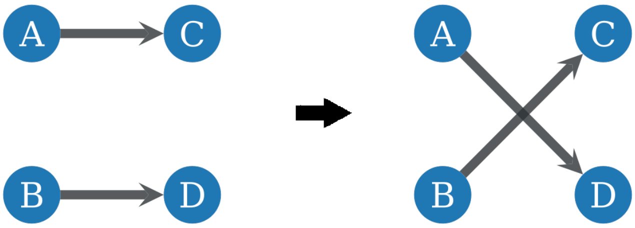


Figure 2.1: Example of a step of degree-preserving randomization procedure. If the new connections form a self-loop or a multi-link then the swap is discarded.

2.3 Undirected and unweighted

In this subsection, the undirected and unweighted case is studied. Some networks are intrinsically undirected, such as Facebook friendships, while other relationships are instead directed, such as Instagram follows. Even in the last case, it may be useful to study the network as undirected to, first, gain insights on a simpler case and, secondly, to see whether new patterns can appear by considering the direction and weights of the links too.

First of all, let $N = |V|$ be the number of nodes and $L = |E|$ the number of links, then we say that the network is sparse if $L << L_{max}$ where $L_{max} = \frac{N(N-1)}{2}$ which is the maximum number of links that could be present in a network with N vertices.

Definition 2.3.1. A connected component G_C of a graph G is a subgraph $G_C \subseteq G$ such that from i there is a path to j for every i, j nodes of G_C.

Definition 2.3.2. The degree of a node i is $k_i = \sum_{j=1}^{N} a_{ij}$, so the number of nodes that i is connected to. We will denote the average degree as $\langle K \rangle$.

In many real networks, such as film actors, telephone calls or protein interactions networks [27], the degree distribution p_k has shown fat-tailed behaviour, which suggests the existence of a significant number of nodes, the so-called hubs, with a very large number of connections, while the majority have few.

In some cases, it may be interesting to study the effects of the degree distribution on some properties of the network. Indeed, a broad-tailed distribution of the degrees may influence some other network properties. To assess that, those properties are computed on a null model which keeps fixed only p_k and N while changing other properties by performing a randomization procedure. The null model that will be used is built starting from the structure of the original network. Then, at every step two links are picked uniformly at random and the endpoints of those are swapped, if it does not create any self-loops or multi-links, as in Figure 2.1. Note that the randomization can be applied both to undirected and directed networks. This procedure creates then a randomized network with the same degree distribution as the original one, which allows us to compare whether the degree influences or not other properties.

To detect and study relevant nodes, the degree is just the first quantity which can be computed. To address this task, many others were developed.

2.3.1 Degree centrality

There are many different measures to compute the relevance of a node in a network depending on different factors. If the degree is considered relevant, then the classic degree centrality is the best choice, on the other hand, if distances are an important feature to consider, then betweenness or closeness centralities are a better option. In the following, different definitions of centrality are introduced.

Definition 2.3.3. The degree centrality of a node i is $DC(i) = \frac{k_i}{N-1}$ which indicates the fraction of nodes i is connected to. Then, the degree centrality of a network is defined as

$$DC = \frac{1}{N}\sum_{i=1}^{N}(DC_{max} - DC(i))$$

where $DC_{max} = \max_i DC(i)$.

Note that $DC \in [0,1]$, in the extremes $DC = 0$ for a graph in which all nodes have the same number of connections, while $DC = 1$ for a star graph, i.e. a network in which the only links are the ones that connect a central node to any other one.

Other interesting measures define as central a node based on different characteristics such as distances from other nodes or how central they are in a path that connects other vertices. While DC is a local measure, the following ones are more global since they take into account the whole structure of the graph instead of just its neighbours.

Definition 2.3.4. The betweenness centrality of a node i is

$$BC(i) = \sum_{k,l \neq i} \frac{g_{kl}(i)}{g_{kl}}$$

where g_{kl} is the number of shortest paths from k to l, while $g_{kl}(i)$ is the number of shortest paths from k to l passing through i.

This measure is particularly useful in human mobility problems. In that setting, a node, which is for instance a bus station, is relevant not only based on the number of transports that can bring people to that place but also on the number of times we need to pass through that place to reach another one.

Another centrality measure based on shortest paths is the closeness centrality, which rewards nodes that can reach others in the smallest amount of steps.

Definition 2.3.5. The closeness centrality of a node i is the inverse of the average shortest path distance to i:

$$CC(i) = \frac{N-1}{\sum_{j=1}^{N-1} d_{ij}}$$

A major drawback of the measures above is the expensive computational cost [10].

Finally, a different point of view can be to measure the centrality based on the centrality of the neighbours.

Definition 2.3.6. Let the centrality of node i be x_i, then $\lambda\mathbf{x} = A\mathbf{x}$ for a certain λ. Using this definition

$$x_i = \frac{1}{\lambda}\sum_{j=1}^{N} a_{ij}x_j = \frac{1}{\lambda}\sum_{j\in N(i)} x_j$$

where $N(i)$ is the set of neighbours of i. Then, x_i is called the eigenvector centrality of node i.

2.3.2 Clustering coefficient

In real world networks, a pattern which appears more frequently than in random graphs is a high clustering of nodes. By this we mean a high number of nodes who are friend of their friends. Thinking about human relationships it is a tendency that can be expected and understood. This behaviour can be encapsulated in the computation of the clustering coefficient, which can be obtained in two different ways using a local or global coefficient.

The latter exploits the count of triangles in the network, which represents the connection between friends of a node, over the total number of times that friends of that node could have been friends, computed as the number of triplets.

Definition 2.3.7. The global clustering coefficient of a network G is defined as

$$C_{gl}(G) = \frac{3 \times Number\ of\ triangles}{Number\ of\ connected\ triplets}$$

where a connected triplet is a set of three vertices i, j, h such that $i \sim j$ and $j \sim h$.

In the above definition, the factor 3 normalizes the coefficient: there are indeed 3 triplets in a triangle. Even if this measure is able to capture the clustering of the network, the major drawback stands in its global nature. Having a single value of clustering per node would enable a better understanding of the behaviour of each vertex allowing comparisons between different characteristics of single nodes.

The local clustering coefficient is instead defined per node and it computes the number of friends of a node which are connected themselves over the total amount of possible connections between them.

Definition 2.3.8. Let i node of a network G, the inter-connectivity of node i is the number of neighbours of i who are neighbours themselves, so:

$$IC(i) = |\{(j,k) \in E | j,k \in N(i)\}|$$

The clustering coefficient of i is then defined as:

$$C(i) = \frac{IC(i)}{\binom{k_i}{2}}$$

where $C(i) = 0$ if $k_i = 0, 1$. The clustering coefficient of G is then:

$$C(G) = \frac{1}{N}\sum_{i=1}^{N} C(i)$$

As anticipated many real networks, such as email messages, protein interactions and Internet [27], have been shown to feature a higher $C(G)$ than expected in a randomly wired graph. In particular, $C(G)$ is respectively 0.16, 0.071 and 0.39. Finally, it is interesting to use this measure to compare the centrality of nodes, via the degree, to the connectedness among their friends, via $C(i)$.

2.3.3 Distances

Distances play a central role in network analysis. As said before, they can be used to define central nodes, but also to recognize certain behaviours, such as small-world property, that will be investigated in this section.

Average shortest path

The average shortest path length $\langle d \rangle$ is able to capture in how many steps, on average, starting from a random node, we can reach another one.

Definition 2.3.9. The average shortest path length of a network G is computed as the average length of the shortest paths between connected nodes, so:

$$\langle d \rangle = \frac{\sum_{i \neq j,\ i \sim j} d_{ij}}{N(N-1)}$$

This measure has become famous thanks to the 6 degrees of separation theory for which every human being is connected to another one via on average only 6 links [25]. Many real networks not only show a small $\langle d \rangle$ value but also, as anticipated in the previous section, a high clustering coefficient, creating a phenomenon called small-world behaviour. For a network to have such that behaviour means that even if the total number of nodes is high, every node knows many friends of their friends and it is able to reach every other node in a small amount of steps. A priori it is a counterintuitive pattern for very large networks, but it is constantly found in real situations and it is characterized by large $C(G)$ and small $\langle d \rangle$.

Diameter

The diameter of a connected network measures the highest number of steps required to move from one node to another one.

Definition 2.3.10. The diameter of a network G is $d_m = \max_{i \neq j} d_{ij}$.

2.3.4 Degree correlation

In a general randomly wired graph, we expect nodes to be linked to others independently of their degree. Nevertheless, in many social interactions, a different pattern is detected. It is not unusual, indeed, that more popular nodes interact with other popular ones, while less popular ones tend to bond with each other. This behaviour can be seen in many human interactions from scientific collaborations to mobile phone calls [3]. At the same time, the contrary may happen: very central nodes are linked to less central ones. At first, it may seem very unlikely to observe a pattern like this in human interactions, but let's consider for instance a network whose nodes are shops and people, where there is an edge between two vertices if one is a client of the other. In this graph, nodes with high degrees, which will probably be shops, are going to be linked with small degree nodes, so clients, rather than to other highly connected vertices. Another example is the World Wide Web (WWW) network which shows the same behaviour. Studying how nodes link to each other based on their degrees is thus going to tell us more about the characteristics and mechanisms behind the network.

First of all, we call a network neutral if the number of connections between nodes of degree k and k' coincides with what we expect by a randomly wired network, for every k and k'. If we assume that each node chooses randomly its connections, the probability of two nodes with degrees k and k' to be linked to each other is $p_{k,k'} = \frac{kk'}{4L^2}$. If $p_{k,k'}$ of a network deviates from the one of a random graph, then we say that there is a degree correlation of some type. A network is called assortative when hubs tend to connect to each other and avoid unpopular nodes, while unpopular nodes tend to connect to each other. Finally, a network is called disassortative if popular nodes avoid each other to connect to unpopular ones.

Recalling the computations of [3], we can introduce the degree correlation measures.

Degree correlation matrix

Let G an undirected network, p_k be the distribution of the degrees and consider every undirected link (i, j) as a double directed link $i \to j$ and $j \to i$. Let $E = \{e_{ij}\}_{ij}$ be the degree correlation matrix, where e_{ij} is the probability that selecting randomly a link it has as source a node with degree i and as target a node with degree j. Note that in an undirected network $e_{ij} = e_{ji}$. In real cases, it is computed as the number of times a node with degree i has a neighbour with degree j normalized with $2L$. Let $q_k = \sum_j e_{kj}$, then

$$q_k = \sum_j \frac{\#\text{ connections between nodes with degree } k \text{ and } j}{2L} = p_k N \cdot k \cdot \frac{1}{2L}$$

where the first term is the number of nodes with degree k and the second one is the total number of connections each node of degree k has. Then q_k can be written as:

$$q_k = \frac{kp_k}{\langle K \rangle}$$

where $\langle K \rangle$ is the average degree. This quantity can be interpreted as the probability of randomly selecting a link with source a node of degree k. Since E is symmetric in the undirected case, the sum of row k is equal to the sum of column k. This implies

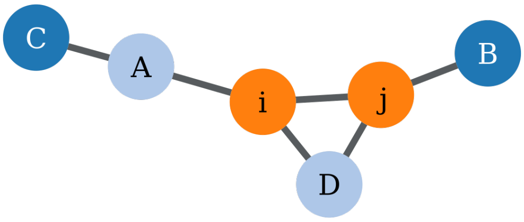


Figure 2.2: Example of a simple network. The colours correspond to different degrees of the nodes.

that q_k is also the probability that selecting a random link the target is a node with degree k. Then in a neutral network, we have $e_{ij} = q_i q_j$, which is expected from random graphs.

Note that the matrix E does not contain actual correlation coefficients, nonetheless, since in the literature it is known under that name, the same terminology is adopted in this thesis.

This matrix can be used as a visualization tool for correlation, however, it presents some drawbacks. First of all, it is not straightforward to extract meaningful information by visually inspecting the matrix, especially for real networks where we may have some variability and a wide range of degrees which generates a matrix with large size. Moreover, it is difficult to compare networks with different correlations because it is not possible to understand the magnitude of such correlations.

Degree correlation function

To overcome the limitations of the matrix inspection, we can compute degree correlation based on the average degree of the neighbours of a node with a certain degree. In particular, the degree correlation function is defined as

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

where $P(k'|k)$ is the conditional probability that a node with degree k is connected to a node with degree k'. Thus, $k_{nn}(k)$ is the average degree of the neighbours of nodes with degree k. To make this definition clearer, $k_{nn}(k)$ of the network in Figure 2.2 is computed. Nodes i and j both have degree 3, so

$$k_{nn}(3) = 1 \cdot \frac{1}{6} + 2 \cdot \frac{3}{6} + 3 \cdot \frac{2}{6}$$

where the first term accounts for the unique 1-degree connection, the second term accounts for links with 2-degree nodes which have 3 links with i, j. The last part considers 3-degree nodes where we have to consider twice the link (i, j): the idea is that i is connected to a 3-degree node once and the same hold for j.

In a neutral network, we have:

$$P(k'|k) = \frac{P(k,k')}{q_k} = \frac{e_{kk'}}{q_k} = \frac{q_k q_{k'}}{q_k} = q_{k'}$$

So then:

$$k_{nn}(k) = \sum_{k'} k' q_{k'} = \sum_{k'} k' \frac{k' p_{k'}}{\langle K \rangle} = \frac{\langle K^2 \rangle}{\langle K \rangle}$$

For a neutral network, we then expect a constant degree correlation function. Instead, in an assortative one, nodes with similar degrees connect with each other, leading to an increasing degree correlation function. With the same reasoning, for a disassortative network, we have a decreasing $k_{nn}(k)$.

However, it is not possible to draw immediate conclusions just by looking at the plot of $k_{nn}(k)$. Indeed some networks may show disassortative behaviour while having an underlying assortative or neutral pattern. For instance, the presence of a scale-free degree distribution implies that there is a small number of nodes with a high degree and many with a much lower one. Consequently, hubs cannot link with many other hubs simply due to the small presence of nodes with such a high degree. Since this limit, which is intrinsic to the structure of the graph, makes the network appear as disassortative, this phenomenon is called structural disassortativity. In particular this problem emerges when k is bigger than a threshold called k_s. As shown in [3], the threshold is $k_s \propto (\langle K \rangle N)^{1/2}$. Still, if data shows a disassortative behaviour, it is not certain, even if we are in the structural regime, that the pattern is induced simply by the shape of the degree distribution. To check this, we can simply run a degree-preserving randomization procedure that maintains the number of nodes and the degree distribution of the real network unchanged. This procedure will instead destroy every type of correlation between degrees. Visualizing the degree correlation function of the randomized network, we will then be able to see whether the decrease in the function is caused by the fat tail of the degree distribution or not. Indeed, if $k_{nn}(k)$ of the randomized does not show any decrease, then we can conclude that the disassortativity is not structural, otherwise, it is.

Correlation coefficients

It is possible to characterize the correlation of degrees with a single number. There are two ways to define a coefficient for correlation, the first is called Pearson degree correlation coefficient, as introduced in [26] by Newman, while the second one is based on Spearman correlation.

Definition 2.3.11. The Pearson degree correlation coefficient is defined as:

$$r = \sum_{j,k} \frac{jk(e_{jk} - q_j q_k)}{\sigma^2}$$

where $\sigma^2 = \sum_k k^2 q_k - \left[\sum_k k q_k\right]^2$.

r is thus simply the Pearson coefficient between the degrees found at the end of the same link. Recall that $-1 \leq r \leq 1$ and if $|r| = 1$ then we have linear correlation.

Thus, in this setting, a high absolute value of r indicates that the degree at the end of a link linearly depends on the degree at the other extreme. The sign of r indicates the sign of the slope of the linear fit, suggesting in this case assortativity or disassortativity.

If instead of studying linear dependence we want to focus on monotonic dependence, then the Spearman rank coefficient is able to measure monotonic correlations between variables.

Definition 2.3.12. The Spearman rank correlation coefficient is defined as

$$r_s = \frac{Cov(R(K), R(K'))}{\sqrt{Var(R(K))Var(R(K'))}} = r_{R(K),R(K')}$$

where $R(K)$ and $R(K')$ are the ranked samples of K and K' degrees at the end of a link. Thus, it is the Pearson coefficient of the ranked random variables.

In this case, if $|r_s|$ is close to 1 then K' depends monotonically on K, again a positive (negative) sign suggests increasing (decreasing) monotonicity, respectively.

Coefficients are a quick and immediate way to get an idea of the correlation in a network. However, this immediateness is also a drawback in terms of reliability. The Spearman rank is detecting just monotonicity, which is deeply connected to the behaviour we want to detect, but at the same time, a straight line with a small negative slope value would be considered as neutral behaviour from a network perspective, while $r_s = -1$. At the same time, r suffers from some criticality. It is indeed a measure regarding linear dependence, thus if data is clearly assortative but can be poorly fitted with a straight line, r is going to be small. Finally, r_s and r are both coefficients that sum up the behaviour of many nodes, thus they can be overly impacted by outliers. For these reasons, the degree correlation function provides a better option for the study of correlations since it allows for a broader analysis on top of which r and r_s can be considered too.

2.3.5 Characteristics correlation

Interactions between nodes can be influenced not only by their degree but by some other characteristics too. Every vertex may have a different attribute which can be categorical or not. For instance, every node can represent a human which can be female or male. It may be interesting to study the presence of correlation between the characteristic of neighbours to detect a particular behaviour of the dataset, which in this example would be how different genders interact with each other. We are going to focus just on categorical attributes because others can be easily analyzed using the degree correlation measures.

First of all, consider an attribute for every node and let m_{ij} be the total number of contacts between nodes of category i and j, so

$$m_{ij} = \sum_{k<l} a_{kl} \cdot \delta(Attribute(k) = i, Attribute(l) = j)$$

Note that $M = \{m_{ij}\}_{i,j}$ is symmetric since $m_{ij} = m_{ji}$.

This matrix encapsulates the correlation between the chosen characteristic by counting how many links are present between different categories. Clearly, this count is not so informative since it is not normalized. Indeed a common attribute will have high values in all entries simply because there are more of those nodes with that characteristic available, thus, probably, they will have a higher number of connections with respect to underrepresented ones.

To solve this issue, the matrix M^* is introduced. M^* is obtained from M as follows:

$$m^*_{ij} = \frac{2m_{ij}}{\sum_k m_{ik} + \sum_k m_{jk}}$$

In this way, M^* is not only considering correlations between attribute values but also accounting for the fact that it may be caused by the high or low presence of one of them. By plotting M^*, it will then be possible to inspect its values and if the diagonal terms are bigger than the other entries, then an assortative behaviour for that attribute is detected.

Finally, as done for the degree, it is possible to sum up the results by computing the assortativity coefficient. Historically, the coefficient is defined similarly to the degree case, both presented in the paper by Newman [26]. Consider $E = \{e_{ij}\}_{i,j}$ and q_j as defined in Subsection 2.3.4.

Definition 2.3.13. The feature assortativity coefficient is defined as:

$$r_L = \frac{\sum_i e_{ii} - \sum_i q_i q_i}{1 - \sum_i q_i q_i} = \frac{Tr(E) - sum(E^2)}{1 - sum(E^2)} \tag{1}$$

where $sum(E^2)$ is the sum of all elements of the matrix E^2.

Some remarks on the above definition can be done:

- The second equality indeed holds: recall that $[E^2]_{ij} = \sum_k e_{ik}e_{kj}$, then:

$$\begin{aligned}\sum_{i,j}[E^2]_{ij} &= \sum_{i,j}\sum_k e_{ik}e_{kj} = \sum_k \sum_{i,j} e_{ik}e_{kj} \\ &= \sum_k \left(\sum_i e_{ik}\right)\left(\sum_j e_{kj}\right) = \sum_k q_k q_k\end{aligned}$$

- Apart from extreme cases, this formula is always well defined. Let $\sum_j e_{kj} < 1$ for every k, so assume that the sum of every row is less than 1. This hypothesis is not restrictive from an application point of view: if it does not hold for some row k' then every other row has all entries equal to 0, which corresponds to categories different than k' being not active. Clearly, it would not hold also for matrices with just 1 row, which again does not make sense for the scope of this formula. With this assumption is now possible to prove that the denominator is always different than 0:

$$\sum_{i,j}[E^2]_{ij} = \sum_{i,j}\sum_k e_{ik}e_{kj} = \sum_k \sum_i e_{ik}\left(\sum_j e_{kj}\right) < \sum_{k,i} e_{ik} = 1$$

Depending on the behaviour of the nodes, r_L has different values:

- For perfectly assortative mixing E is diagonal and $\sum_i e_{ii} = 1$, then $r_L = 1$.
- For random mixing $e_{ij} = q_i q_j$, implying $r_L = 0$.
- Under a perfectly disassortative mixing, meaning there exists no link between nodes with the same attribute, then $\sum_i e_{ii} = 0$, thus

$$r_L^{min} = -\frac{\sum_i q_i q_i}{1 - \sum_i q_i q_i} \in [-1, 0)$$

The idea behind this coefficient is to consider an overall view of assortativity in the network. In this way, r_L will be high if over-represented attributes in the dataset are assortative because in the computation every vertex has the same weight. From the network perspective, it is reasonable to assume so. Otherwise, assigning the same weight to every feature's value rather than to every node, under-represented categories would have too much influence. For instance, consider a network with 3 groups where the first 2 are made of many nodes while the 3rd only contains 2 vertices. If the first 2 groups are disassortative, while the third one has the opposite behaviour, then r_L is going to define the network as disassortative, while other measures such as the one defined by Gupta et al. in [16] will detect the assortativity of the last group.

As said, to analyze the overall behaviour of the network it is correct to assign to each node the same weight. At the same time, to study the behaviour of each attribute is better to proceed as done above for the mixing matrix.

2.3.6 Communities

This section will face the problem of community detection, following the work of Peixoto [31], discussing the possible algorithms to use. Community detection is an important task in network science. In some cases, we may need to divide nodes into groups to maximize or minimize a function, which can happen for instance with technological networks. There the objective may be to reduce space or energy by placing and grouping certain vertices. In social or biological networks, instead, it may be interesting to identify communities who collaborate and work together, to understand better the habits of the components. For instance, consider the network of contacts where there is a disease spreading across people, in this case, recognizing the communities is key to understand the diffusion of the virus and prevent it. In these cases, discovering these structures can tell a lot about the behaviour of the nodes and on the process behind the creation of the network itself.

Given the importance of this task, many methods to address it have been developed. They follow either a descriptive or an inferential approach. The first detects groups based on descriptive quantities of the network rather than finding an explanation behind the formation of these communities. On the other hand, the latter does exactly the opposite, starting from a generative model to explain the emergence of these groups. While the majority of algorithms for community detection are descriptive they should be used just in specific situations. In particular, they are thought to be used when the aim is to find a partition of the network to optimize some tasks, rather than find communities themselves. Applying such optimization algorithms outside these cases presents various drawbacks. First of all, the idea behind descriptive methods is to maximize a function and a usual common choice is modularity. The modularity of a network with adjacency matrix A is:

$$Q(A,b) = \frac{1}{2L}\sum_{i,j}\left(A_{ij} - \frac{k_i k_j}{2L}\right)\delta_{b_i,b_j}$$

where $b = (b_1, \ldots, b_N)$ indicates the node's community, k_i is the degree of node i and L is the number of edges of the network. This measure compares the existence of an edge in the real network G to the probability that such that edge exists in a null model. The underlying assumption of these algorithms is that a partition of a network is significant if the occurrence of links between vertices of the same group is higher than what we would expect with a random null model without communities. The partition chosen is then the one that maximizes Q, so:

$$b_{max} = \text{argmax}_b Q(A,b)$$

This method is prone to overfitting, being able to recognize the presence of communities also in fully random null models. This happens because it does not consider the deviation from the random case in a proper statistical way. Indeed the formula of Q, which should compute that deviation, ignores the maximization step. Consider a randomly wired network, varying b, the distribution of $Q(A,b)$ is expected to be centered around 0. Consequently, there might exist a b with a high value of Q, but the majority of the partitions will have Q close to 0. Using modularity maximization, b_{max} is chosen as the partition with highest Q, ignoring the fact that for the vast majority of b, no clear community is detected. This will cause community recognition even in random null models, clearly overfitting the data. At the same time, due to a resolution limit, it may underfit data. Indeed it cannot find more than $\sqrt{2L}$ communities in a connected network. Furthermore, many networks present a high modularity plateau, leading to a difficult choice among partitions which share the same high value of Q. It also tends to find groups of similar sizes, specifically with the same sum of degrees.

Since descriptive methods present all these limitations, an inferential approach may be preferred. First of all, Stochastic Block Model (SBM) inference will be introduced, to then deepen the actual method used for the recognition.

Stochastic Block Model

A stochastic block model (SBM) [30] is a generative model able to create a network from nodes partitioned into B groups. Let $b = (b_1, \ldots, b_N)$ where $b_i \in \{1, \ldots, B\}$ represents the block membership of node i, which are considered as iid random variables with a certain distribution. Consider the matrix π where π_{rs} is the probability that there exists a link between nodes of community r and s. Then, in a simple graph where $A_{ij} \in \{0,1\}$ we assume:

$$A_{ij}|\pi, b \sim Be(\pi_{b_i b_j})$$

which implies:

$$p(A|\pi, b) \propto \prod_{i<j} \pi_{b_i b_j}^{A_{ij}} (1 - \pi_{b_i b_j})^{1-A_{ij}}$$

Note that it can be generalized to non-simple graphs where $A_{ij} \in \mathbb{N}$ by considering e instead of π, where e_{rs} is the actual number of links between groups r and s. It can be extended to directed networks too by allowing e to be asymmetric. Due to the different nature of the network, in this case, the placement of the links is modelled as a geometric distribution.

To detect communities, the goal is now to find the partition b which has generated the empirical network A under the assumption that it was generated by an SBM. Thus, the ultimate goal is to find $p(b|A)$ and in particular b that maximizes this posterior probability. This can be done via different approaches in a Bayesian framework after choosing the priors.

It is possible to develop a hierarchical procedure [32] by acknowledging that communities themselves can be seen as nodes of a multigraph where e_{rs} is the number of edges between nodes r and s. This multigraph can be considered as built from an SBM too, and proceeding recursively we end up with a final model with only one block. This approach provides different level of resolution with respect to the one level approach.

2.3.7 Graph models

Many network models were developed to recreate what has been observed in real datasets. In this section two famous ones are presented: Erdős-Renyi and Barabasi-Albert. To show the different characteristics of graphs generated by these models, the values of many of the measures just presented are given [3]. These models can then be used as a benchmark to compare real networks to them.

Erdős-Renyi model

Many models for static networks have been developed through the years. The oldest and easiest one is the Erdős-Renyi model.

Definition 2.3.14. Given N, the number of vertices, and p a probability, then Erdős-Renyi model generates a network G with N vertices where at each step two random nodes i, j are connected with probability p. This procedure is repeated for each pair. This graph is also called random network or random graph.

The degree follows a binomial distribution :

$$p_k \sim Bi(p, N-1)$$

which implies an average degree of $\langle K \rangle = p(N-1)$.

The local clustering coefficient is independent on the node degree since $C(i) = p$ for every i, then clearly $C(G) = p = \frac{\langle K \rangle}{N-1}$.

In a random graph, the average number of nodes at distance d should be approximately $\langle K \rangle^d$, then the expected number of vertices at distance at most d from a random node is:

$$N(d) \approx 1 + \langle K \rangle + \cdots + \langle K \rangle^d = \frac{\langle K \rangle^{d+1} - 1}{\langle K \rangle - 1}$$

where $N(d)$ is limited by the total number of nodes in the network. Thus, letting d_m be the diameter it should hold $N(d_m) \approx N$. Then, assuming $\langle K \rangle >> 1$, we have:

$$\langle K \rangle^{d_m} \approx N \;\Rightarrow\; d_m \approx \frac{\ln N}{\ln \langle K \rangle}$$

In many real networks, however, this formula fits better $\langle d \rangle$ rather than d_m because d_m is influenced by few long paths, while $\langle d \rangle$ is an average over all couples of vertices. In both cases, this equation indicates a logarithmic dependence between $\langle d \rangle$ or d_m and the size of the network N, which is seen as small world behaviour regarding distances.

Being a random model, Erdős-Renyi networks are neutral and present no instance of communities.

Barabasi-Albert model

From the necessity of explaining the presence of fat-tailed distributions of the degree in real world networks, the Barabasi-Albert model was developed.

Definition 2.3.15. The Barabasi-Albert model is defined as follows: start with a network of m_0 nodes where each of them has at least one link. Then at every time step a node with m edges is added. The probability that one of these links connects to an already existent node i is:

$$p(i) = \frac{k_i}{\sum_j k_j}$$

This mechanism is known as preferential attachment (PA) for which popular nodes are going to be more popular having a higher probability of being chosen for a connection by new vertices.

After t steps, the network is made by $t+m_0$ nodes and m_0+tm links. It is possible to prove that this model generates networks with power-law degree distribution of exponent 3.

The diameter d_m scales as

$$d_m \propto \frac{\ln N}{\ln \ln N}$$

and $\langle d \rangle$ has the same scaling for large N. The average clustering coefficient $C(G)$ scales as

$$C(G) \propto \frac{(\ln N)^2}{N}$$

and assuming k_i to be large, $C(i)$ is independent on k_i itself for every i.

Again, since nodes are connected just based on PA, then this network is neutral and no communities are expected to be found using SBM.

2.4 Directed and weighted

This section will provide all the instruments to carry out the same analyses as before but for directed and weighted networks. Many measures are adapted from the undirected case, while for others new quantities are introduced.

Definition 2.4.1. Let G be a directed graph, then:

- $G_{SC} \subseteq G$ is said to be a strongly connected component of G if for every i, j nodes of G_{SC} there exists a directed path from i to j;
- $G_{WC} \subseteq G$ is said to be a weakly connected component of G if for every i, j nodes of G_{WC} there exists an undirected path from i to j.

Note that weakly connected components coincide with the connected components of G as undirected.

2.4.1 Strength

The natural adaptations of degree for directed and weighted networks are the in and out-strengths.

Definition 2.4.2. Let i node of a network G, the in and out-strength of i are

$$s_i^{in} = \sum_j w_{ji}, \quad s_i^{out} = \sum_j w_{ij}$$

where w_{ij} is the weight of the link (i, j). If $w_{ij} = 1$ for every i, j, then the in and out-strength are called in and out-degree respectively.

2.4.2 PageRank

Not many centrality measures can be generalized to directed weighted networks, and when it is possible they do not necessarily exploit these additional characteristics in the best way. Just thinking about distance-based centrality measures, they can be adapted easily by considering the length of the shortest directed path between two nodes. However, many real graphs have multiple strongly connected components. This implies that for many i, j there is no path from i to j, so d_{ij} cannot be computed. Thus, by convention, many nodes have, for instance, a closeness or betweenness centrality value of 0.

A famous and common centrality measure that was developed especially for directed networks (but can be extended to weights) is the PageRank, which was introduced for webpages to detect the more relevant ones. The idea behind it is to consider as more central, pages on which a user has more probability of ending up by following the links across other pages. This translates into a discrete-time random walk on the directed network. For the theoretical background, the reference is [6], while for the PageRank results, we refer to [24].

Definition 2.4.3. Consider a stochastic process $\{X_t\}_{t\in\mathbb{N}}$ with set of states A. $\{X_t\}$ is a Markov chain if it satisfies the Markov property:

$$P(X_{t_n+1}|X_{t_n},\dots,X_{t_0}) = P(X_{t_n+1}|X_{t_n})$$

for every $t_0 < t_1 < \dots < t_{n+1}$.

Let $P(X_{t+1} = j|X_t = i) = T_{ij}(t)$, then the Markov chain is time homogeneous if $T_{ij}(t) = T_{ij}$ for every t. The matrix $T = \{T_{ij}\}_{ij}$ is called transition matrix. Given an initial distribution on the states $X_0 \sim p(0)$ where $p_i(0) = P(X_0 = i)$, define $p_i(t)$ as the probability that the chain, at time t, is in state i:

$$p(t) = p(0)T$$

Definition 2.4.4. A state of the Markov chain $i \in A$ is called:

- transient: if the probability of first return to i starting from i is $\rho_{ii} < 1$.
- recurrent: if the probability of first return to i starting from i is $\rho_{ii} = 1$. If the average time of the first return is finite then the state is called positive recurrent, otherwise it is called null recurrent.
- absorbing: if $T_{ii} = 1$.

Let $B \subseteq A$ be a set of states, if, for every $i, j \in B$, i leads to j then B is said to be irreducible. The Markov chain is irreducible when the set of all states is irreducible.

Definition 2.4.5. A probability distribution π on the states is said to be invariant or stationary with respect to the transition probability matrix T if $\pi = \pi T$.

It is called ergodic if for any initial distribution on the states $(p_i(0))_i$, it holds $\lim_{t\to\infty} p_i(t) = \pi(i)$.

Theorem 2.4.6. *The following results regarding a stationary distribution can be proved.*

- *If there is a finite number of states, then there exists at least one invariant distribution*
- *An irreducible Markov chain admits a unique stationary distribution if and only if all the states are positive recurrent.*

Definition 2.4.7. A Markov chain is said to be regular if there exists $t \geq 1$ such that $p(t, i, j) > 0$ for every state i, j. Equivalently, there exists $t \geq 1$ such that all entries of T^t are positive.

Note that a regular Markov chain is irreducible and positive recurrent.

Theorem 2.4.8. *Consider a regular Markov chain and π the unique invariant distribution. Then π is ergodic.*

Now, it is possible to introduce the Markov chain that will be used to compute the PageRank as done in [24]. Consider a random walker in discrete time, at each step, the walker at node $i \in \{1, \dots, N\}$ can jump to one of the out-neighbours of i according to the transition probability

$$T_{ij} = \frac{A_{ij}}{k_i^{out}}$$

which defines the transition matrix $T = \{T_{ij}\}_{ij}$. If i has no out-neighbours then $T_{ii} = 1$. The chain is time-homogeneous since T_{ij} does not depend on time. The probability that the walker at time $t+1$ is in j is given by

$$p_j(t+1) = \sum_{i=1}^{N} p_i(t) T_{ij}$$

Then the idea is to define the PageRank as the unique solution of this equation independent on the choice of $(p_i(0))_i$, so as the ergodic distribution of the chain. In general, there not exist a unique stationary distribution. In real graphs, the number of states is finite, thus at least one exists, however, since it is likely to have more absorbing states, then it is not unique. For instance, consider the simple network

$$A \leftarrow B \rightarrow C$$

then $T_{BA} = T_{BC} = \frac{1}{2}$ and $T_{AA} = T_{CC} = 1$, while the other entries are 0. In this case, both $(1, 0, 0)$ and $(0, 0, 1)$ are stationary distributions, and there not exists an ergodic one. Clearly, every strongly connected component of the network forms an irreducible class of states, but the chain itself is not irreducible.

To avoid this problem, some tricks are introduced to transform the Markov chain as regular. One of these methods is to introduce teleportation regulated by a rate α:

$$p_j(t+1) = \alpha \sum_{i=1}^{N} p_i(t) T_{ij} + (1-\alpha) u_j$$

where u_j represents the probability of the walker going to j when it teleports and $1-\alpha$ is the probability of teleportation. If $u_i \neq 0$ for every i and $0 \leq \alpha < 1$, then from every node it is possible to reach another one at every step, so the modified Markov chain is regular. This ensures the existence of a unique ergodic stationary distribution p^*, which is the PageRank of the network G. This measure defines a node i as relevant if it can be reached via many links and if the sources of the incoming edges are important vertices with a small out-degree. The choice of α will affect the computations: a value closer to 0 will fasten the convergence but reduce the effect of the true underlying network. Thus, a value closer to 1 is usually picked, which reduces the effect of teleportation but still transforms the chain as regular. The common choices are $\alpha = 0.85$ and $u_i = \frac{1}{N}$ for every i.

The generalization to weights is trivial: if an edge from i to j has weight w_{ij}

then the probability of moving from i to j is

$$T_{ij} = \frac{w_{ij}}{\sum_k w_{ik}}$$

2.4.3 Clustering coefficient

In the undirected case, the clustering coefficient is able to draw a picture of the closest relationship between a node and its friends. The basic structures that this measure considers are triplets and triangles, however, in the directed case, more triangle configurations, in total 8, need to be considered to study the interactions between a node and its neighbours. Furthermore, taking into account the weights of the links enables us to weigh the strength of a relation, thus the definition of this coefficient needs an additional adjustment. The introduction of this modified clustering coefficient refers to the computations made by Fagiolo in [11].

First of all, note that the clustering coefficient for a binary undirected network without self-loops can be rewritten in the following way:

$$C(i) = \frac{\frac{1}{2}\sum_{j\neq i}\sum_{h\neq(i,j)} a_{ij}a_{ih}a_{jh}}{\frac{1}{2}k_i(k_i-1)} = \frac{(A^3)_{ii}}{k_i(k_i-1)}$$

where $(A^3)_{ii}$ is the ii entry of A^3.

When generalizing it to a weighted undirected network, a higher weight is considered as a hint of a stronger relationship between two nodes. To account for weights, every weight w_{ij} is normalized by $w_{max} = \max_{i,j} w_{ij}$, so $\tilde{w}_{ij} = \frac{w_{ij}}{w_{max}}$ and the geometric mean is used as follows.

Definition 2.4.9. Let G_w be a weighted undirected network with adjacency matrix W, the clustering coefficient of a node i is defined as follows:

$$C_w(i) = \frac{\frac{1}{2}\sum_{j\neq i}\sum_{h\notin\{i,j\}} \tilde{w}_{ij}^{1/3}\tilde{w}_{ih}^{1/3}\tilde{w}_{jh}^{1/3}}{\frac{1}{2}k_i(k_i-1)} = \frac{(\tilde{W}^{[1/3]})^3_{ii}}{k_i(k_i-1)}$$

where $\tilde{W}^{[1/k]} = \{\tilde{w}_{ij}^{1/k}\}$ is the matrix obtained from $\tilde{W}$ by taking the k-th root of each entry and k_i is the number of connections of node i.

If G_w is not weighted then $C_w(i) = C(i)$. Note that it accounts for the weight of all links forming the triangle and it is invariant to weight permutation in a triangle.

Now, we extend the definition of $C(i)$ to the direct case and finally, we are going to merge the two generalizations to obtain the measure that will be used. First of all, we can introduce some notation:

- $k_i^{in/out}$ is the in/out-degree of node i.
- $k_i^{tot} = k_i^{in} + k_i^{out}$ is the total degree of node i.
- The number of bilateral edges between i and its neighbours, so the total count

of vertices j such that there is an edge from i to j and vice versa:

$$k_i^{\leftrightarrow} = \sum_{j \neq i} 1_{w_{ij} \neq 0} 1_{w_{ji} \neq 0}$$

The first extension that can be made is to consider all possible directed triangles formed by each node regardless of the directions of their links. Each product $a_{ij}a_{ih}a_{jh}$ is related to one specific triangle among the 8 possible combinations.

Definition 2.4.10. Given a directed network G_d, the clustering coefficient of i is defined as the ratio between all directed triangles formed around i, called t_i^D, and the number of all possible triangles that i could have had, called T_i^D:

$$C_d(i) = \frac{t_i^D}{T_i^D}$$

It is possible to write this ratio as:

$$C_d(i) = \frac{\frac{1}{2}\sum_{j \neq i}\sum_{h \neq i,j}(a_{ij} + a_{ji})(a_{ih} + a_{hi})(a_{jh} + a_{hj})}{k_i^{tot}(k_i^{tot} - 1) - 2k_i^{\leftrightarrow}} = \frac{(A + A^T)^3_{ii}}{2(k_i^{tot}(k_i^{tot} - 1) - 2k_i^{\leftrightarrow})}$$

Remark 2.4.11. The equivalence above actually holds, indeed

$$T_i^D = k_i^{tot}(k_i^{tot} - 1) - 2k_i^{\leftrightarrow}$$

Node i can be connected with at most $\frac{k_i^{tot}(k_i^{tot}-1)}{2}$ pairs of neighbours. With each couple, i can create up to 2 triangles, since the edge between the couple can be oriented in 2 ways. This computation actually counts as different nodes all the vertices j with bilateral edges with i. It is correct to do so because one time they are counted as in-neighbour of i and the second time as out-neighbour and having a different direction on that link, they can generate different triangles with other nodes. However, this number counts also false triangles formed by i, j as in-neighbour of i and j as out-neighbour of i, which clearly are not possible. For each of these, we have falsely counted 2 triangles, so we need to subtract $2k_i^{\leftrightarrow}$ to solve the issue.

Note that if A is symmetric, the measure reduces to the undirected case having $C_d(i) = C(i)$.

Merging the two extensions presented above, the measure for the directed weighted case can be defined as follows.

Definition 2.4.12. Let G_{dw} be a directed weighted network, then the clustering coefficient of i is given by:

$$C_{dw}(i) = \frac{t_i^{DW}}{T_i^{DW}} = \frac{[\tilde{W}^{[1/3]} + (\tilde{W}^T)^{[1/3]}]^3_{ii}}{2(k_i^{tot}(k_i^{tot} - 1) - 2k_i^{\leftrightarrow})}$$

If the network is unweighted, then $C_{dw}(i) = C_d(i)$. If W is symmetric then $C_{dw}(i) = C_w(i)$.

In general, we expect the clustering coefficient of the weighted directed network to be lower than in the undirected case, since it computes the number of triangles present over all possible combinations.

2.4.4 Distances

The diameter or the average shortest path length for the directed case can be obtained easily from the undirected one, presented in Subsection 2.3.3. Nonetheless, as explained in Subsection 2.4.2, we expect to have more than one strongly connected component. Thus, it is not possible to go from every node to another one with a path, making impossible to compute d_m and $\langle d \rangle$.

2.4.5 Strength correlation

Studying the correlation of the degree of neighbours can help us study how nodes relate to each other depending on their popularity. As explained for the undirected case in Subsection 2.3.4, on some occasions we may expect a certain pattern to appear. Considering the direction and the weights, the study has to be generalized to relations between in and out-strength.

Strength correlation function

The analyses will be the same as done before but divided for correlation between out-strength and out-strength, out and in, in and out, in and in. Thus, we will compute now 4 strength correlation functions in the same way as before where, for instance, for the out-in case we have:

$$k_{oi}(k) = \sum_{k'} k' P(k'|k)$$

where now k represents the out-strength of node and k' the in-strength of its out-neighbours. For every property of the source node, the function counts the average property of the target which is an out-neighbour of the source. Consider, for instance, the following simple network:

$$A \rightarrow B$$

Then, $k_{ii}(0) = 1$ while k_{ii} cannot be computed on 1 because B has in-degree 1, but has no out-neighbour.

Correlation coefficients

As done before in Subsection 2.3.4, for every combination, we can compute the Spearman coefficient or the Pearson coefficient by considering strength instead of the degree and out-neighbours instead of neighbours. Note that in this setting the sum of row k of E is different than the sum of column k, since E is asymmetric. Then, in this setting, in a neutral network we expect $e_{ij} = \left(\sum_k e_{ik}\right)\left(\sum_k e_{kj}\right)$.

2.4.6 Characteristics correlation

The study of relations among characteristics of the nodes can be done in the directed and weighted case too with just a few adjustments. Consider as in Subsection 2.3.5, the matrix $M = \{m_{ij}\}_{ij}$ where m_{ij} is the total number of links from a node with feature i to nodes with feature j (the sum of the weights of the links).

The normalization, similarly to before, can be made in the following way: each entry is divided by the number of out-links of nodes with feature i plus the number of in-links of nodes with feature j and multiplied by 2. So that the new entries are

$$m^*_{ij} = \frac{2m_{ij}}{\sum_k m_{ik} + \sum_k m_{kj}}$$

Note that now M^* is asymmetric. The assortativity can then be studied by visualizing the plot of M^*.

The general feature assortativity of the network can be assessed via the Pearson correlation coefficient, which is computed as before with slight refinements as done for the strength.

2.5 Time

The methods presented in the previous sections allow to study the interactions between nodes without considering any temporal aspects. However, considering just the static network is limiting. Human interaction has shown to be influenced by past contacts [40] and in general many more patterns and properties of the human behaviour can be discovered introducing the temporal components.

For every node we are going to consider the sequence of event times $\{t_i\}$ where t_i is the time of the ith event and $\tau_i = t_{i+1} - t_i$ is the inter-event time (IET).

The first interesting quantity that can be studied is the distribution of IETs. Ranging different datasets, such as mobile phone call, short messages or email sequences, Karsai et al. in [21] showed that IETs present a power-law behaviour.

2.5.1 Burstiness

The classical definition of burstiness was introduced by Goh et al. in [14].

Definition 2.5.1. Consider an inter-event time sequence $\{\tau_i\}_{i=1}^n$, then the burstiness coefficient is given by

$$B = \frac{\sigma - m}{\sigma + m} = \frac{CV - 1}{CV + 1}$$

where the coefficient of variation is defined as the ratio between the standard deviation and the mean of the sequence, so $CV = \frac{\sigma}{m}$.

For specific inter-event time distribution the measure has a precise value.

- If the sequence is periodic then $B = -1$ because $\sigma = 0$.
- A Poisson process induces an exponential distribution for τ, then $B = 0$ because $\sigma = m$.

- For heavy-tailed distributions $\sigma >> m$, then B is high. In an extreme case $B = 1$, which is considered as an extremely bursty sequence.

Another measure that can be computed is the local variation coefficient, originally introduced in [35].

Definition 2.5.2. The local variation coefficient of $\{\tau_i\}_{i=1}^n$ is given by

$$LV = \frac{3}{n-1}\sum_{i=1}^{n-1}\left(\frac{\tau_i - \tau_{i+1}}{\tau_i + \tau_{i+1}}\right)^2$$

This measure is less sensitive to outliers, however, a power law distribution for the inter-event times will generate, by definition, very high values with a small but relevant probability. For this reason, it is not problematic that B might be influenced by outliers. Again, LV has precise values for specific time sequences.

- $LV = 0$ for a periodic sequence.
- $LV = 1$ for a Poisson process.
- A bursty sequence yields to a large value of LV.

In general $0 \leq LV < 3$. As noted by [24], differently from CV, LV is affected by correlation between τ's.

2.5.2 Temporal correlation

It may be the case that an event induces a bursty sequence of other events. For instance, a message sent in a group chat may generate a discussion which is going to influence successive IETs. For this reason, to perform a more precise and complete analysis of the temporal aspects, correlation has to be investigated too.

Memory coefficient

In [14], together with B, the memory coefficient M is introduced too.

Definition 2.5.3. The memory coefficient of an inter-event time series $\{\tau_i\}_{i=1}^n$ is defined as the correlation of consecutive time IETs:

$$M = \frac{\sum_{i=1}^{n-1}(\tau_i - m_1)(\tau_{i+1} - m_2)}{\sqrt{\sum_{i=1}^{n-1}(\tau_i - m_1)^2 \sum_{i=2}^{n}(\tau_i - m_2)^2}}$$

where $m_1 = \frac{1}{n-1}\sum_{i=1}^{n-1}\tau_i$ and $m_2 = \frac{1}{n-1}\sum_{i=2}^{n}\tau_i$.

A positive value of M suggests a positive correlation between consecutive IETs. As seen in many real datasets of human activities [14], the memory coefficient is slightly higher than what is expected for a Poisson process.

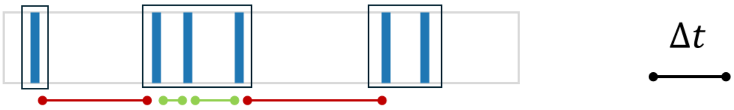


Figure 2.3: Burst trains for Δt on the right. Events are represented with straight blue lines, red and green intervals corresponds, respectively, to greater and lower IETs than Δt. In the example there are 3 trains with size 1, 3 and 2.

Autocorrelation

In the previous subsection, we have seen one measure of correlation of inter-event times. Another way to study them is to focus on the autocorrelation of a specific process [20]. Consider the following process:

$$x(t) = \sum_{i=1}^{n+1} \delta_{t,t_i}$$

where $x(t)$ counts the number of events which happened at time t. Then the autocorrelation function with delay time t_d is defined as:

$$A(t_d) = \frac{\langle x(t)x(t+t_d)\rangle_t - \langle x(t)\rangle_t^2}{\langle x(t)^2\rangle_t - \langle x(t)\rangle_t^2}$$

If the time series presents temporal correlations, then $A(t_d)$ typically decays as a power law $A(t_d) \sim t_d^{-\gamma}$. Even if this measure is used in many studies, it presents some problematic behaviours. Indeed, it is influenced by broad shape distributions of IETs which are common in many real datasets. As shown in [21], sampling τ's from independent power laws will generate a power law decay of $A(t_d)$, thus suggesting the presence of correlation which does not exist. More specifically, the dependence between the power law exponent of the distribution of τ and γ has been studied and proven [20]. The limits of this measure can be overcome using the count of burst train sizes, a measure proposed by Karsai et al. in [21].

Burst train size

Definition 2.5.4. Consider a sequence of events and an interval time Δt. A burst train is a sequence of events such that the IET between successive events is at most Δt, and those between the first event of the sequence and the preceding event, and between the last event and the consecutive, are larger than Δt.

The number of events in a train is called its size E. For an easier understanding, an example of burst trains can be seen in Figure 2.3.

If the inter-event times are independent, then the distribution $p(E)$ of burst train size follows a geometric distribution:

$$p(E) = \left[\int_0^{\Delta t} f(\tau)d\tau\right]^{E-1} \left[1 - \int_0^{\Delta t} f(\tau)d\tau\right]$$

where $f(\tau)$ is the distribution of the inter-event times. Note that this formula is not influenced by the shape of $f(\tau)$. For a finite Δt:

$$\int_0^{\Delta t} f(\tau) d\tau = a < 1$$

So then $p(E)$ satisfies the following:

$$p(E) = a^{E-1}(1-a)$$

As a consequence, for any independent inter-event sequence, $p(E)$ decays exponentially even if $f(\tau)$ has a broad tail. Any difference in the shape of $p(E)$ can be taken as an indicator of temporal correlation of inter-event times.

In many real cases, $p(E) \propto E^{-\beta}$ with $\beta \in [2.3, 4.1]$. Thus, burst trains constituted by many events are more likely to appear than in uncorrelated sequences of τ. Correlation implies that short τ tends to follow after another short one to make the burst size larger. Moreover, $p(E)$ shows a robust behaviour with respect to different Δt, which is why its dependence is omitted by the notation.

Memory function

As proposed in [21], starting from $p(E)$, it is possible to study another function which is called the memory function.

Definition 2.5.5. The memory function $p(n)$ of a time series is defined as the probability that having already n events in a burst train, the next event is going to be part of that same burst train. Thus, it can be written as:

$$p(n) = \frac{\sum_{E=n+1}^{\infty} P(E)}{\sum_{E=n}^{\infty} P(E)} = 1 - \frac{P(E=n)}{\sum_{E=n}^{\infty} P(E)}$$

We are going to estimate each $P(E)$ as the ratio between the number of trains of size E over the total amount of trains.

Assuming that $P(E) \sim E^{-\beta}$, we expect

$$p(n) \sim \left(\frac{n}{n+1}\right)^{\nu} \quad \text{with} \quad \beta = \nu + 1$$

indeed:

$$p(n) = \frac{\sum_{E=n+1}^{\infty} P(E)}{\sum_{E=n}^{\infty} P(E)} \approx \frac{\int_{n+1}^{\infty} Dx^{-\beta} dx}{\int_{n}^{\infty} Dx^{-\beta} dx} = \frac{[x^{-\beta+1}]_{n+1}^{\infty}}{[x^{-\beta+1}]_{n}^{\infty}} = \left(\frac{n}{n+1}\right)^{\beta-1}$$

which suggests $\beta = \nu + 1$. It is possible to show that also the vice versa holds, so assuming the shape above for $p(n)$ it can be proven that $P(E)$ follows a powerlaw with exponent β [21]. By studying $p(n)$ and $p(E)$, it is possible to discover not only the potential correlation of the IETs, but also the shape of the memory function which regulates the formation of burst trains. This information can then be used to develop a model which resembles the burst train creation process under these laws.

2.5.3 Deseasonalization

Starting from burstiness to temporal correlation, one may argue that these phenomena are naturally induced by the circadian rhythms of human life. Indeed, simply following daily rhythm would force short IETs to follow short ones, with a small but still relevant percentage of longer ones caused by nighttime. In [23], they suggested that these phenomena were the cause of the power-law shape of IETs. To address this query, a deseasonalization procedure can be applied to discover whether daily or weekly rhythms are the only causes of these behaviours.

The procedure that will be used here was introduced by Jo et al. in [19] and in [18]. Let $n_i(t)$ the number of events at time t of node i. The number of events at time t for a set of vertices A is then given by

$$n_A(t) = \sum_{i \in A} n_i(t)$$

Let T be the period, in this thesis we will just consider 1 day period. Then the event rate $\rho_{A,T}(t)$ with $0 < t < T$ is computed as

$$\rho_{A,T}(t) = \frac{T}{s_A} \sum_{k=0}^{\lfloor T_f / T \rfloor} n_A(t + kT) \quad \text{where} \quad s_A = \sum_{t=0}^{T_f} n_A(t)$$

where T_f is the time of the last event. For $t \geq 0$, $\rho_{A,T}(t) = \rho_{A,T}(t + kT)$ for any $k \geq 0$. Finally, the rescaled time $t^*(t)$ is defined as

$$t^*(t) = \sum_{0 \leq t' < t} \rho_{A,T}(t')$$

Under this rescaling, the time is dilated at periods of high activity and contracted for low activity. Finally, let t_j the time of the jth event in the ith node, then the rescaled inter-event time τ_j^* is given by the following:

$$\tau_j^* = t^*(t_{j+i}) - t^*(t_j) = \sum_{t_j \leq t' < t_{j+1}} \rho_{A,T}(t')$$

Now, every statistics computed above, such as burstiness or temporal correlation, can be recomputed on the rescaled time series. The results will show whether circadian patterns played an actual role in those phenomena or not. Note that, fixing different T will result in removing different patterns: with the choice of T as 1 day, daily patterns are removed. Moreover, the activity level of nodes is usually broadly distributed, thus dividing the set of nodes into groups of activity level leads to a better and more precise rescaling.

CHAPTER 3

Results

This chapter focuses on the analyses of the forwarding phenomenon on Telegram which is viewed under two perspectives: network structure and timing.

In the first part, of great interest, will be the study of friend of my friend dynamic, i.e. clustering, the detection of potential communities and the underlying bond of these groups, the language assortativity and the analysis of the centrality of nodes. While, in the second one, the focus will switch to the inter-event times analyzing potential correlation, seasonality effects and other properties.

3.1 Undirected and unweighted

We are going to build an undirected and unweighted network where nodes correspond to chats. In this setting, between chat A and chat B there is a link if, at some point in time, one of them has forwarded content from the other. In the dataset, there are some chats that have been forwarded from, but whose messages were not collected due to the stop in the snowballing procedure. The study has been performed both on a full network and a reduced one. As a full network, we refer to the structure obtained by considering every chat whose content was forwarded, even if its messages were not retrieved. Clearly, this procedure creates a big network with around $3.5 \cdot 10^5$ vertices, but the vast majority of them are without additional properties. More importantly, for this vast majority of nodes, no messages are retrieved, thus this network misses a lot of real information. For instance, let A be a channel in the dataset that forwarded a message from channel B which is not in the dataset. Considering the full network, we would have a link between A and B, but it may happen that B has forwarded content of another channel in the dataset C, but this information would not be at our disposal, since the messages of B are not retrieved. In light of this example, it is clear how a network built in this way would draw a false picture of Telegram. This is the reason why a reduced one is considered. Note that the same explanation and choice hold for the directed case.

The reduced network is, on the other hand, the graph built as explained above keeping only the nodes whose messages were retrieved, so such that messages and other properties are given. In this way, the true underlying relationships between two nodes can be detected. Indeed if there is no link between A and B, then for sure

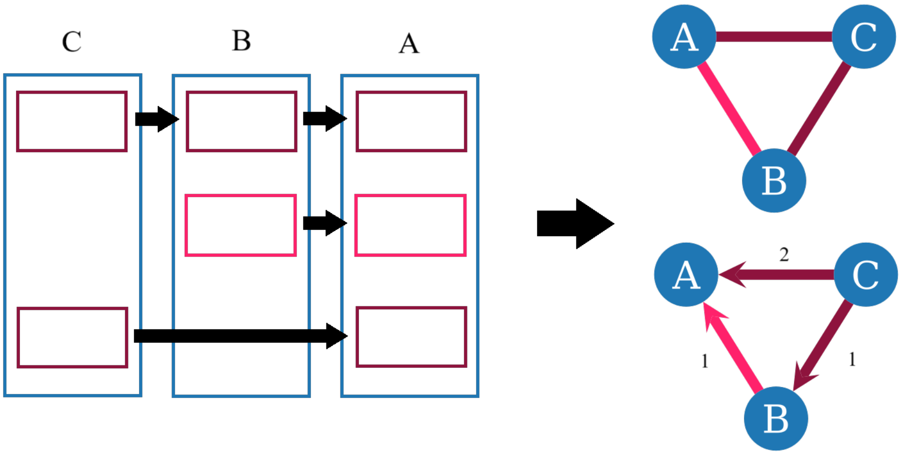


Figure 3.1: The interactions between channels A, B, C are plotted with the corresponding networks: both undirected and directed and weighted. Each blue box represents a chat, the boxes inside are messages coloured depending on in which chat they were sent first. The black arrow indicates that a message has been forwarded from one chat to another. On the right, the corresponding networks are shown.

they have never forwarded anything from one another. Additionally, if someone has forwarded a message from themselves, it would correspond to a self-loop, which is not informative for the aim of the study, thus any self-loop is discarded. After removing these edges, some nodes are left without any link, creating isolated entities which again are not interesting for the study since they do not provide any knowledge of the flow of information. Thus, they are discarded too. The process to build the network is illustrated in Figure 3.1.

From the implementation point of view, the following packages has been used to manipulate the network: networkx [17], networkit [37] and graph tool [33]. The last two have an underlying C++ implementation, which ensures a quick execution of the code, and they provide additional features not present in the first package, such as degree preserving randomization or stochastic block model inference.

3.1.1 General statistics

The number of nodes N present in the network is 29 609, while the number of edges L is 472 163. Note how the number of nodes is slightly lower than the total number of chats collected, but it is due to the fact that we discarded self-loops and isolated vertices. The graph is sparse since the number of edges $L << L_{max}$ where $L_{max} = \frac{N(N-1)}{2}$ which is the maximum number of links that could be present in a network with N vertices. In particular $L/L_{MAX} \approx 1.1 \cdot 10^{-3}$.

In the considered network there is one connected component, so every node is reachable from another one.

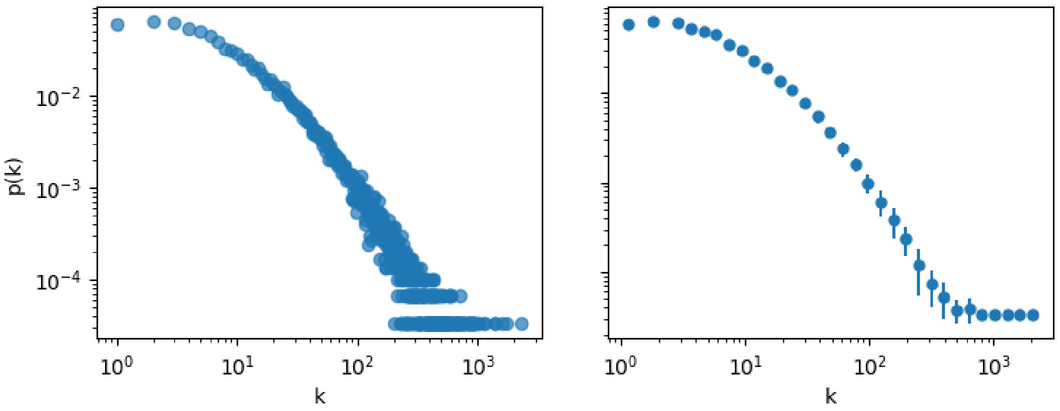


Figure 3.2: Log-log plot of the PMF of the degree: a) Scatter plot of the distribution, note that for high degrees a plateau appears. b) Log-binning of the plot on the left, where the errorbars correspond to the std of each bin.

3.1.2 Degree

The maximum degree is 2 299 and the average degree is $\langle K \rangle = \sum_i \frac{k_i}{N} = \frac{2L}{N} = 31.89$. In many real networks, such as film actors, telephone calls or protein interactions network [27], the degree distribution follows a power law distribution. In Figure 3.2, the degree probability mass function (PMF) is plotted. For high degrees, a plateau appears, which can be explained by the intrinsic limit of the degree of a node. At high degrees, there is just a small number of nodes or no nodes with such degree, and this forms the plateaus. The distribution however does not resemble a perfect power-law, which should appear as a straight line, but it shows a scale-free behaviour. Recall that, a power-law-shaped degree distribution was predicted by the Barabasi model, while for the Erdős-Renyi the binomial distribution would have been expected.

3.1.3 Degree centrality

For the Telegram network, the degree centrality $DC \approx 7.66 \cdot 10^{-2}$.

Type	**Sample Size**	**Mean**	**Std**
Barabasi-Albert	100	$1.82 \cdot 10^{-2}$	$3.14 \cdot 10^{-3}$
Erdős-Renyi	45	$8.5 \cdot 10^{-3}$	$5.78 \cdot 10^{-5}$

Table 3.1: Degree centrality of networks generated with known models: Barabasi-Albert and Erdős-Renyi. The two models generate networks with smaller DC.

In Table 3.1, the value of DC for networks generated from known models is shown. From this measure we can understand that Telegram is structured in hubs and less relevant nodes. Indeed, it has a DC higher than classical models.

3.1.4 Clustering coefficient

In Telegram, it is natural to think that a channel A, which is interacting with another one B (by forwarding from or vice versa), is more likely to interact with a node C which is already linked with B. For instance, an admin of channel A reads the content of channel B and forwards some messages from it, at the same time in the chat of B are present some messages of channel C, thus A is reading the content of C via B. It should therefore be likely for admin A to forward from C.

This intuition is reaffirmed by the value of $C(G) \approx 0.248$ which is much higher than what it is expected from a random model. As mentioned in Subsection 2.3.2, this value is in line with the results of other real-world networks [27].

Type	Sample Size	Mean	Std
Barabasi-Albert	100	$2.3 \cdot 10^{-3}$	$2.16 \cdot 10^{-4}$
Erdős-Renyi	45	$1.08 \cdot 10^{-3}$	$1.22 \cdot 10^{-5}$
Degree-preserving randomized	100	$2.28 \cdot 10^{-2}$	$3.52 \cdot 10^{-4}$

Table 3.2: Clustering coefficient of networks generated with known models: Barabasi-Albert, Erdős-Renyi and degree-preserving randomization. $C(G)$ is lower than in Telegram in all of the three cases. In particular, the degree distribution alone cannot explain the high value of the coefficient.

As can be seen in Table 3.2, the three comparative models generate networks with a much lower $C(G)$. Moreover, both Barabasi and Erdős-Renyi models would predict $C(i)$ to be constant with respect to the degree k_i. Regarding the whole network, the latter predicts $C(G) = \frac{\langle K \rangle}{N-1} = 1.08 \cdot 10^{-3}$, while the first one $C(G) \sim \frac{(\ln N)^2}{N} = 3.58 \cdot 10^{-3}$.

In Figure 3.3, the average coefficient mapped against the degree is plotted. The y-value is computed in the following way: for every degree k, consider all nodes i with $k_i = k$, then the average clustering coefficient is the average local clustering coefficient among these nodes, so among vertices with the same degree. It is possible to see that, contrary to the mentioned models, $C(k)$ decreases with respect to the degree k. To check whether this behaviour is simply caused by the fat-tailed distribution of degrees, the same plot is shown for a degree-preserving randomized network. This plot suggests that the degree distribution itself plays no role in the local clustering coefficient: the orange function is constant and lower than the blue one. This is fundamental in understanding whether having a scale-free degree distribution leads to a high clustering coefficient per se. The independence on the degree distribution is proved also by the clustering coefficient of 100 randomized networks, which produced an average clustering coefficient of $2.28 \cdot 10^{-2}$.

We can summarize the results as follows:

- The local clustering coefficient is much higher than the prediction of Erdős-Renyi and Barabasi-Albert models;
- It is not independent on the degree;
- Its value is not a solely consequence of the degree distribution, which has been kept constant in the randomization.

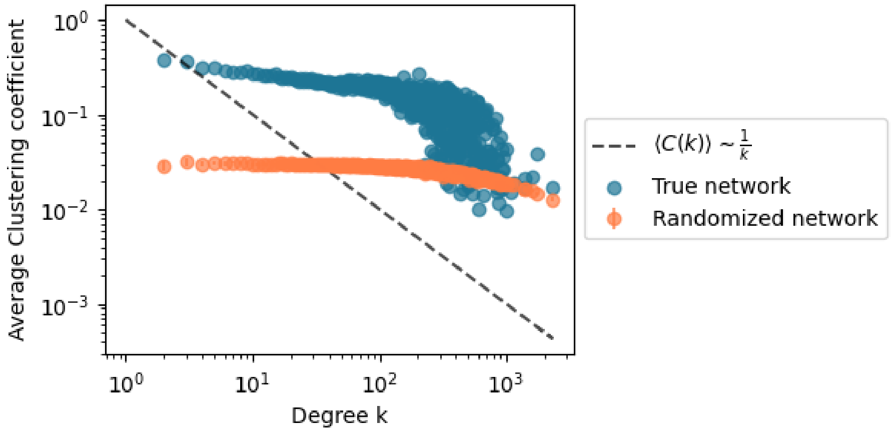


Figure 3.3: The average local clustering coefficient among nodes with the same degree is represented against the degree. The Telegram results are in blue, while the degree-preserving randomized one are in orange. The latter is plotted using an error bar where the points stand for the average coefficient per degree averaged among 10 simulated randomized networks, while the bars correspond to the standard deviation between simulations.

As said above, $C(k)$ decreases with respect to k and this suggests the presence of hierarchy inside the data, a property that both Erdős-Renyi model and Barabasi-Albert ignore. The idea is that small nodes have a high clustering coefficient because they are part of denser communities, while hubs have small $C(i)$ because they connect different communities. Moreover, the average clustering coefficient for the randomized graph decreases a bit but it seems to be constant overall.

3.1.5 Distances

For the Telegram network we have $\langle d\rangle \approx 4.00$, so on average with 4 links we are able to go from one chat to another one. This value is quite small, in line with other real networks [27], and since $C(G)$ is high too, this suggests the presence of a small-world behaviour.

Again, it is interesting to compare the value obtained with the one predicted by other models. More precisely, for a Erdős-Renyi model $\langle d\rangle \sim \frac{\ln N}{\ln\langle K\rangle} \approx 2.97$, while for a Barabasi-Albert $\langle d\rangle \sim \frac{\ln N}{\ln\ln N} \approx 2.33$. The computed $\langle d\rangle$ is of the same order of magnitude of the one obtained from graphs generated by these models.

For Telegram the diameter is 9, meaning that via at most 9 forwarded messages it is possible to reach a chat from another arbitrary one. Both $\langle d\rangle$ and d_m are small compared to the size of the network, reaffirming the tendency of real networks of having small distances.

3.1.6 Degree correlation

In a social network like Telegram, one may expect to detect an assortative behaviour. This would mean that connected chats have a similar number of contacts. To in-

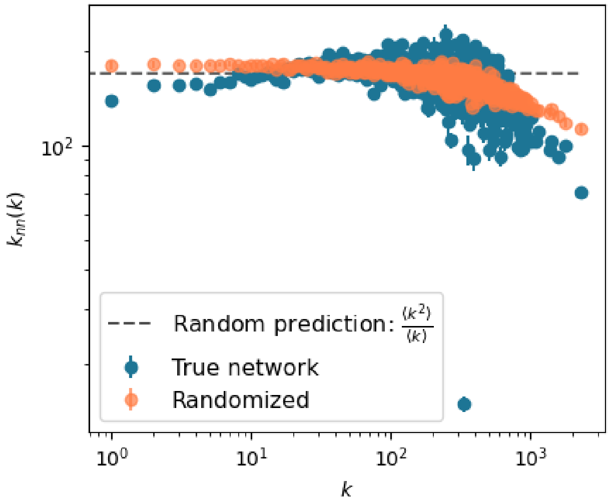


Figure 3.4: The degree correlation function of, respectively, the Telegram network (blue), the degree preserving randomized network (orange). In black, the prediction for a neutral network.

spect the presence of degree correlation, $k_{nn}(k)$ is computed and plotted in Figure 3.4. Recall from Section 2.3.4 that networks may appear as disassortative due to a structural limit caused by the fat-tailed degree distribution. In the Telegram case, we have seen that the degree distribution has indeed such a shape and the structural disassortativity threshold is $k_s(N) \approx 972 < k_{max}$. To check if this phenomenon occurs, we plot $k_{nn}(k)$ for a degree-preserving randomized graph. In the plot, we can visualize the prediction for a neutral random graph too which is $k_{nn}(k) = \frac{\langle K^2 \rangle}{\langle K \rangle}$.

From the plot, the network seems to be neutral, except for a slight decrease for high k, which is present also in the randomized version, indicating structural disassortativity. Note also that the plotted functions are similar to the prediction for a neutral network.

Finally, although they present some drawbacks, the correlation coefficients are computed too. The Pearson coefficient is $r = -0.056$ and for 100 randomized degree-preserving graphs the average is $r_r = -0.042$, respecting the prediction that a randomized has to be neutral. Clearly, the value of r is suggesting linear independence since $|r| \approx 0$. Similarly for 100 Barabasi-Albert models where on average $-2.57 \cdot 10^{-2}$ and for 45 Erdős-Renyi we have $-3.27 \cdot 10^{-4}$.

The Spearman coefficient is instead $r_s = -0.861$, while for the randomized it is on average -0.554. This suggests a monotonic relationship in the function which should not surprise us. Indeed, looking at the plot in Figure 3.4, it is possible to detect a slight decrease. Note also that the randomized value is slightly towards -1 suggesting a decreasing relationship too, which is in line with the remark regarding structural disassortativity.

Overall, the Telegram network is neutral and features, at high degrees, structural disassortativity.

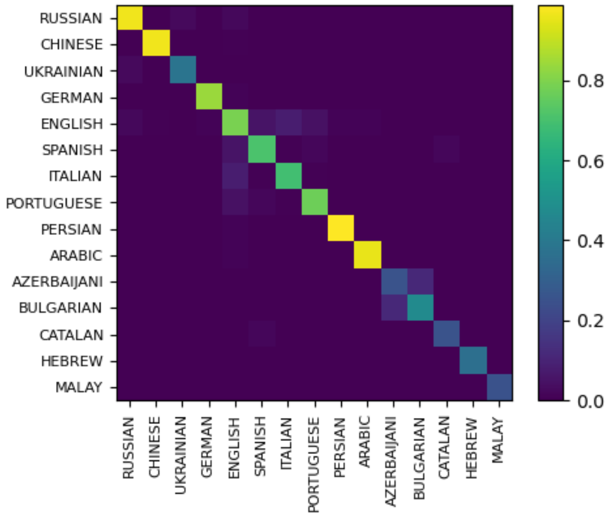


Figure 3.5: Extraction of the original heatmap plot for the language assortativity. The values plotted were selected based on an higher value on the diagonal. Note that, in this subset of languages are present the most popular ones across nodes.

3.1.7 Language

Language is a key component of human relationships, intuitively we expect people speaking the same idiom to interact more than with speakers of different languages. It becomes then natural to investigate whether this pattern can be found in Telegram and if it can be considered a driving force of inter-channel relationships.

To address this task the measures introduced in Subsection 2.3.5 are used as language correlation measures. A reduced M^* is plotted in Figure 3.5 where the languages with the highest value on the diagonal are shown. While, in Figure 3.6, M^* is plotted applying the log on every entry. From the figures, a positive correlation between chats' languages is detected, especially for languages more present in the dataset. Moreover, languages spoken in geographically close regions tend to interact with each other. For instance, Central European languages, such as German, Spanish, Italian, Portuguese and English, interact a lot with each other and the same happens between Russian and Ukrainian too. Interestingly, the chats where no language was recognized, correspondent to unknown language in the plot, seem to speak with every other idioms indistinctly.

Finally, as done for the degree, it is possible to sum up the results by computing the assortativity coefficient. Recall that a coefficient close to 1 suggests assortativity and for Telegram we have $r_L = 0.906$, while for 100 degree-preserving randomized ones on average we have $-9.72 \cdot 10^{-5}$, denoting a clear assortative pattern of the network with respect to languages, which is completely destroyed by randomizing the edge placement.

3.1.8 Communities

To recognize potential communities, the SBM inference approach is applied. After a hierarchical recognition, the highest levels are studied. In particular, the last ones

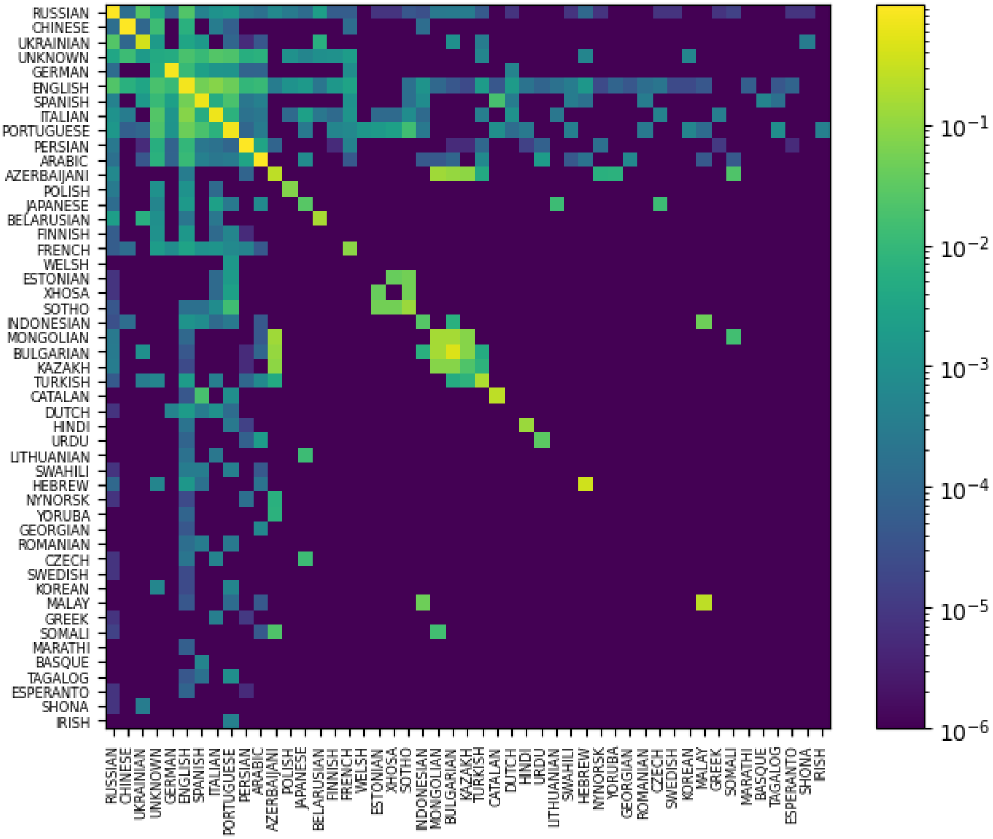


Figure 3.6: Heatmap of M^* on the log values. The zero entries are modified by adding 10^{-6}. A clear positive correlation is present, detected by the high values on the diagonal. It is possible to see other interactions between language with similar cultural background.

consist, respectively, of 7, 3 and 2 communities detected. We have seen how chats tend to interact with those who share the same language, consequently, it is natural to ask whether these communities respect an idiom division or not. The answer can be found in Figure 3.7. Communities are strictly related to the language spoken inside of it and, more specifically, geographical division seems to be predominant in the group formation. This last remark can be explained by the fact that nearby countries may have similar topics to discuss. Additionally, individuals are more likely to be able to talk and understand the language of a nearby country rather than the idiom of a more distant one. As noted in the previous subsection, the chats with no language recognized speak indifferently with others. This tendency is reaffirmed in communities, where the unknown chats split into all groups recognized almost uniformly.

3.1.9 Summary

Throughout the section, it has been possible to see that the Telegram network is characterized by a scale-free behaviour of the degree distribution. It has the small world property, which corresponds to a high clustering and a small average shortest path length. The clustering is high and is not solely induced by the degree distribution. Moreover, a chat interacts with another one (being cited or citing) independently of their degrees. Finally, it is assortative with respect to the language and shows a community structure partially along language lines.

3.2 Directed and weighted

The general setting of the network that will be considered in this section is the same as in the previous one. Differently from before, from channel B to channel A, there is a link if, at some point in time, A has forwarded a message from B. To the link is associated a weight which corresponds to the number of times A has forwarded content from B. Again, both the full network and the reduced one have been analyzed, but similarly to before the most reliable information is obtained from the reduced one. Self-loops are discarded again for the same reason i.e. they do not provide any flow of information between nodes. An example of chat interactions from which this network is built can be seen in Figure 3.1.

3.2.1 General statistics

The number of nodes N present in the network is, as before, 29 609, while the number of edges L is 501 897. In the studied network there are 10 741 strongly connected components with a major one of size 18 578. The total number of events, which corresponds to the sum of edge weights, is 7 500 509. The distribution of the latter, which can be seen at Figure 3.8, seems to have a power-law shape.

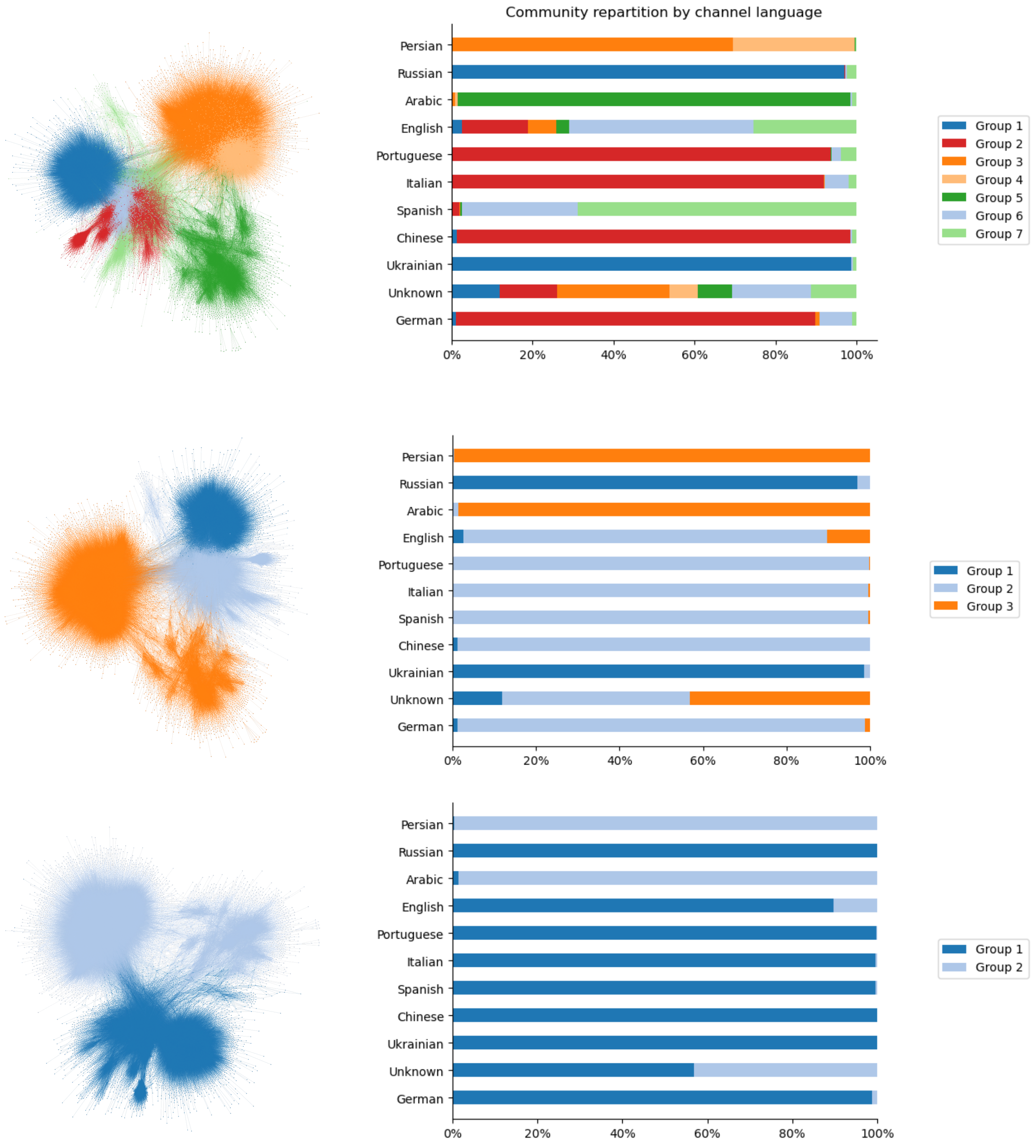


Figure 3.7: Community detection from SBM inference: respectively the first row corresponds to 7 communities detected, the second one to 3 and the last to 2. On the left the network is divided by colour into the detected groups. On the right it is shown the group membership across the 11 most spoken languages.

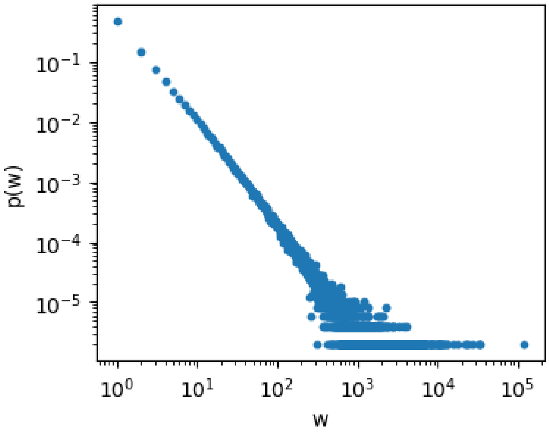


Figure 3.8: Distribution of the weights.

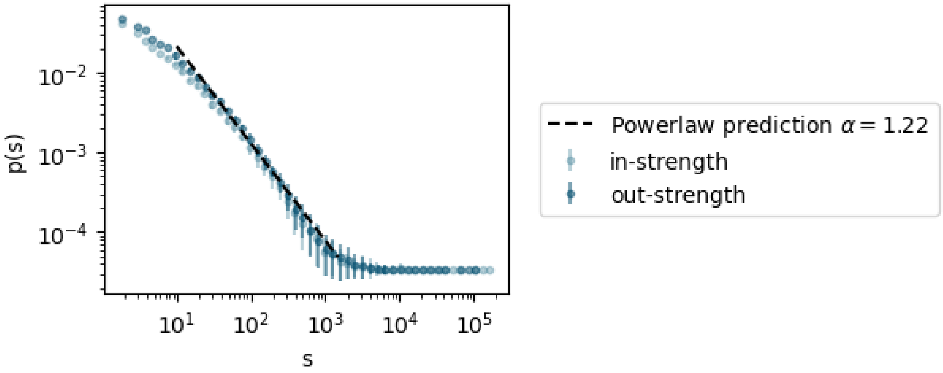


Figure 3.9: The log binning of the PMF of in and out-strength. The straight dashed line is the power law fit for the in-strength values with exponent $\alpha = 1.22$ fitted in a reduced window of values: $[10, 1500]$.

3.2.2 Strength

The average in-strength s_i^{in} and out-strength s_i^{out} are the same:

$$\hat{s} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ji} \approx 253.32$$

By plotting their PMF in Figure 3.9, they seem to follow a power-law behaviour with the usual effect on the tail due to the intrinsic limit in the strength value. The in-strength PMF is fitted as a power-law with exponent $\alpha = 1.22$ in a reduced window of values: $[10, 1500]$.

We may expect that channels with a higher number of participants in them are more cited than others because their content is exposed to more people. A slight monotonic tendency can be seen in the data as shown in Figure 3.10.

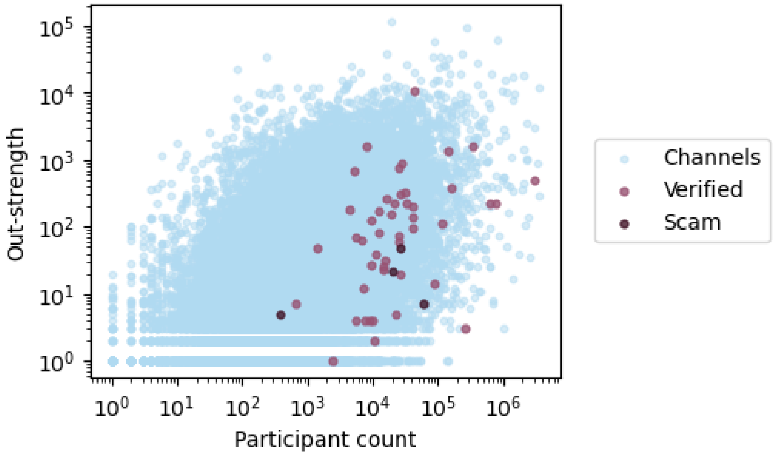


Figure 3.10: Comparison between the out-strength and the number of participants of channels. Points are coloured depending on some particular attribute, such as whether they were flagged as scam or not. A monotonic relation seems to exist, while nothing relevant is detected for verified and scam channels.

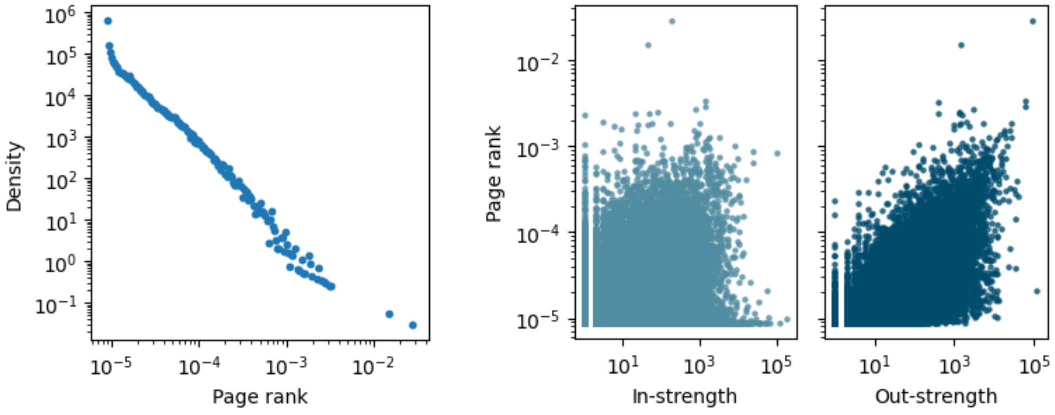
3.2.3 PageRank

In the Telegram network, the direction of the links is chosen to follow the flow of information. Thus, the PageRank would define central nodes as chats where a lot of information arrives, which does not capture the true relevance of a vertex in this setting. For this reason, it makes more sense to compute this measure on the network with reversed links. When a message is forwarded from a channel, from the text itself it is possible to directly go to that channel. Considering the reverse network, we will then have a completely analogous case to the WWW network. In this way, the most central nodes are going to be the one from which information comes.

In Figure 3.11, the PageRank density is shown. As seen for many other distributions in this thesis, the density resembles the one of a power-law, suggesting the presence of a small but consistent number of nodes which are much central than others. Two chats have a high PageRank compared to the others, these are Persian channels with a high number of participants which were created in September 2015. One is the channel of the Khat-e Hizbollah magazine and the other is the channel of the Ayatollah Khamenei Office Information base. The two are thus reasonable chats to have high PageRank since they are actual real-life sources of information and they share it in their chat. Having considered the reversed network, the PageRank should monotonically depend on the out-strength, while no clear dependence is expected for the in-strength. All these predictions are confirmed by looking at Figure 3.11 where the centrality measure is compared to the strengths.

3.2.4 Clustering coefficient

Applying the Definition 2.4.12 to the Telegram network, we expect the clustering coefficient of the network to be lower than in the undirected case, since it computes



(a) (b)

Figure 3.11: PageRank analyses: (a) Density of the PageRank. (b) PageRank compared to the in and out-strength. In the second plot, as predicted, it is possible to see a monotonic dependence between the two variables.

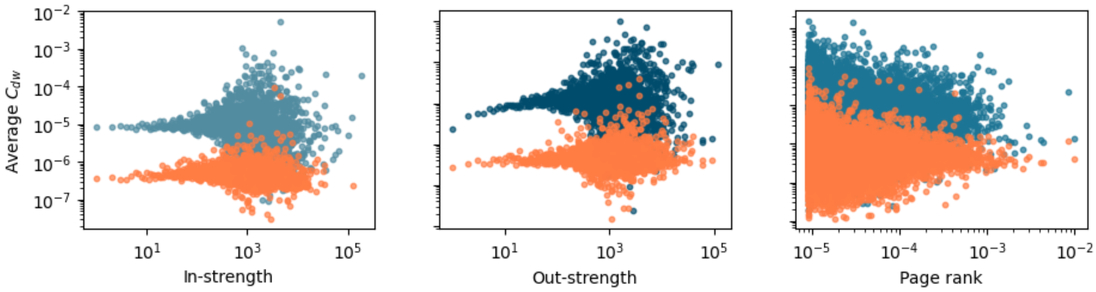


Figure 3.12: Clustering coefficient of nodes with the same in-strength, out-strength, PageRank respectively. In blue, the data from the Telegram network is plotted, while the randomized values are in orange.

the number of triangles present over all possible combinations and in the directed case many more combinations are available. In this case $C_{dw}(G) = 1.06 \cdot 10^{-5}$, while for 10 degree-preserving randomization on average we have $C_{dw}(G_{rand}) = 4.45 \cdot 10^{-7}$. This network still preserves high clustering after adding the direction and weight to the links. In Figure 3.12, we can see the relationship between the clustering coefficient and some centrality measures. In these cases, it seems that there is no clear dependence of the coefficient with respect to any these centrality measures.

Again, the clustering is higher than in the randomized case, thus this high value is not explainable simply by the strength distribution. We can conclude that the Telegram network has a high clustering both in the undirected and in the directed case.

3.2.5 Strength correlation

In the undirected setting, the network appeared to be neutral, so the connection between two nodes is independent on their degrees. We now compute the function

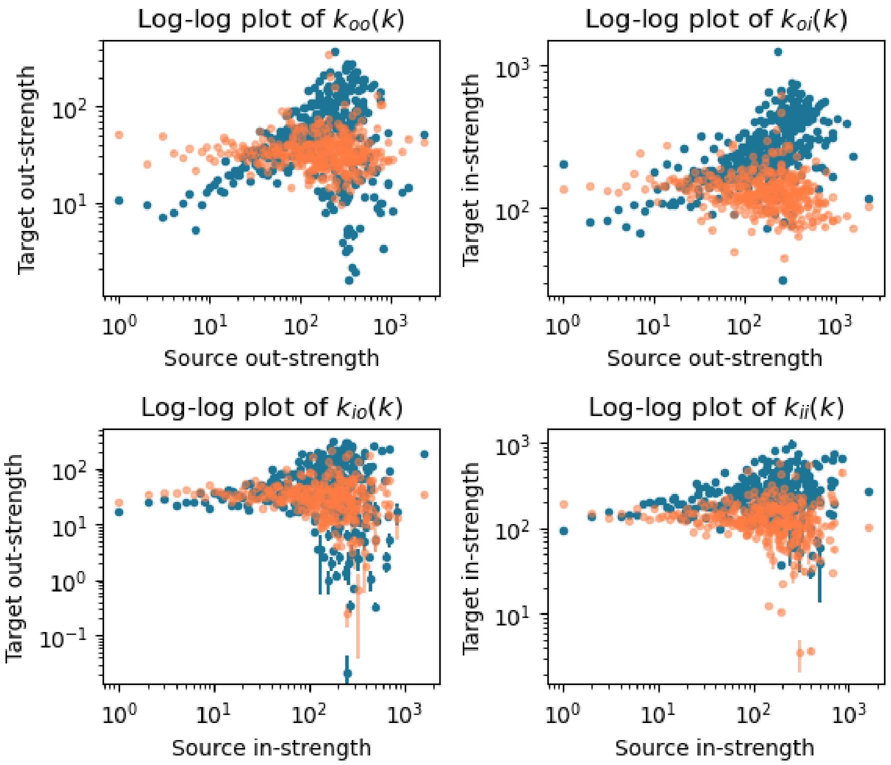


Figure 3.13: Strength correlation function of all 4 combinations. Again, in blue the results from the dataset, while in orange for the randomized network.

for all 4 combinations of strength correlations: out-out, out-in, in-out and in-in.

To explain what assortativity means in this case, first we need to characterize nodes with high/small out-strength and high/small in-strength. Nodes with high out-strength are nodes whose messages are forwarded by either many chats, by some chats many times or both; while nodes with high in-strength are chats that forward a lot of messages from one or many chats.

Thus, assortativity in this setting means that nodes which are cited a lot/less by others are more likely to have as out-neighbour (i.e. be 'forwarded' by) a chat which forwards a lot/less. Note that this may appear as something obvious: if a node has a high out-degree then the out-neighbors are more likely to have high in-degrees, but, in an assortative case, this happens more than what we expect in a randomized network.

Looking at the out-out plot in Figure 3.13, before a threshold (around $2 \cdot 10^2$), it seems to have an assortative behaviour, but for higher values of the source out-strength, it turns out to be disassortative. It could be a hint of structural disassortativity, however the randomized network does not suffer from this. Note that, in the out-in correlation plot, the function seems to suggest assortativity for the Telegram network, while for the randomized is constant. For the in-out, the function seems to have a major drop for high degrees, which to some extent also the randomized has. This might be a suggestion of structural disassortativity for high strengths. Regarding in-out and in-in correlation, the network is neutral. Finally, to measure the monotonic relation between the variables, we compute the Spearman coefficients for every combination which are shown in Table 3.3.

Correlation type	**Spearman coefficient**	
	Network	**Random**
out-out	0.205	-0.0871
out-in	0.585	-0.295
in-out	0.0286	-0.212
in-in	0.195	-0.0896

Table 3.3: Spearman correlation coefficients for the Telegram network and randomized one.

From the table above and the analysis of the plots, we can say that there is an assortative behaviour for the out-in combination and it cannot be explained simply from the distribution of weights. The Spearman coefficient is not able to capture the behaviour of the out-out combination, because the average out-strength of the targets seems to be increasing and then decreasing with respect to the out-strength of the source.

3.2.6 Language

The previous undirected network manifested a strong assortativity of languages which can be checked also in this more complete setting.

As before, a reduced M^* is plotted in Figure 3.14, while the log values of M^* are represented in Figure 3.15. Again, an assortative behaviour between languages can be seen looking at the plot.

The Pearson coefficient in this case is $r_L = 0.940$, while for 10 randomized networks we have on average $-3.27 \cdot 10^{-4}$, denoting again a clear assortative pattern of the network with respect to languages which is destroyed by the randomization. Similar conclusions can be drawn by looking at the Figures 3.14 and 3.15 where the matrix has high values on the diagonal. In the second plot, it is possible to investigate how different languages relate to each other. For instance, Central European idioms speak a lot to each other and the same happens for Russian and Ukrainian.

3.2.7 Communities

As done in Subsection 3.1.8, it is possible to investigate the upper levels of the hierarchical SBM recognition and to study the language distribution across communities, visible in Figure 3.16. In this recognition, the number of groups are respectively 7, 4 and 2. Again the community detection follows a geographical subdivision. Differently from the undirected study, now Arabic is part of the European group instead of the Persian one. This may have happened because European and Arabic chats do not share many links, but they are strong. The other two big groups made of Russian and Ukrainian and the European one are unchanged.

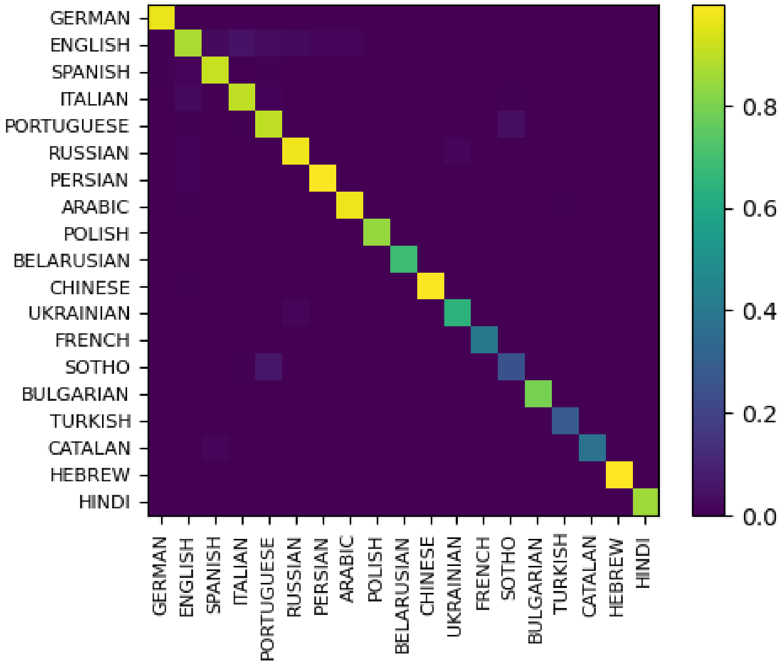


Figure 3.14: Extraction of the original heatmap plot for the language assortativity. The values plotted were selected based on a higher value on the diagonal.

3.2.8 Summary

Adding direction and weight to the links, many properties observed before are found again in this new setting. In particular, in and out-strengths have a scale-free behaviour, the clustering is higher than in the randomized case and the interaction between chats is assortative with respect to the language. Similar communities were recognized with the difference of Arabic chats which share strong ties with Central European speakers.

3.3 Time

In the previous sections the interactions between nodes is studied without considering any temporal aspects. However, considering just the static network is limiting. By introducing the temporal component, we can broaden the study of human dynamics by relating their behaviours to the timing of their interactions too. Thus, in this section, the temporal component of the dataset is studied. We are going to focus on the behaviour of inter-forwards timing. Of particular interest will be the computation of burstiness, temporal correlations and seasonality. All these measures will allow us to characterize the behaviour of Telegram chats in forwarding content.

The number of chats which have ever forwarded at least one message is 22 650. We will refer to event of a node or of a chat as the forwarding of a message. For every node, we are going to consider the sequence of event times $\{t_i\}$ where t_i is the time of the ith event and $\tau_i = t_{i+1} - t_i$ is the inter-event time (IET). Every time is considered in seconds, where $t = 0$ is assigned to the first event of the network. The

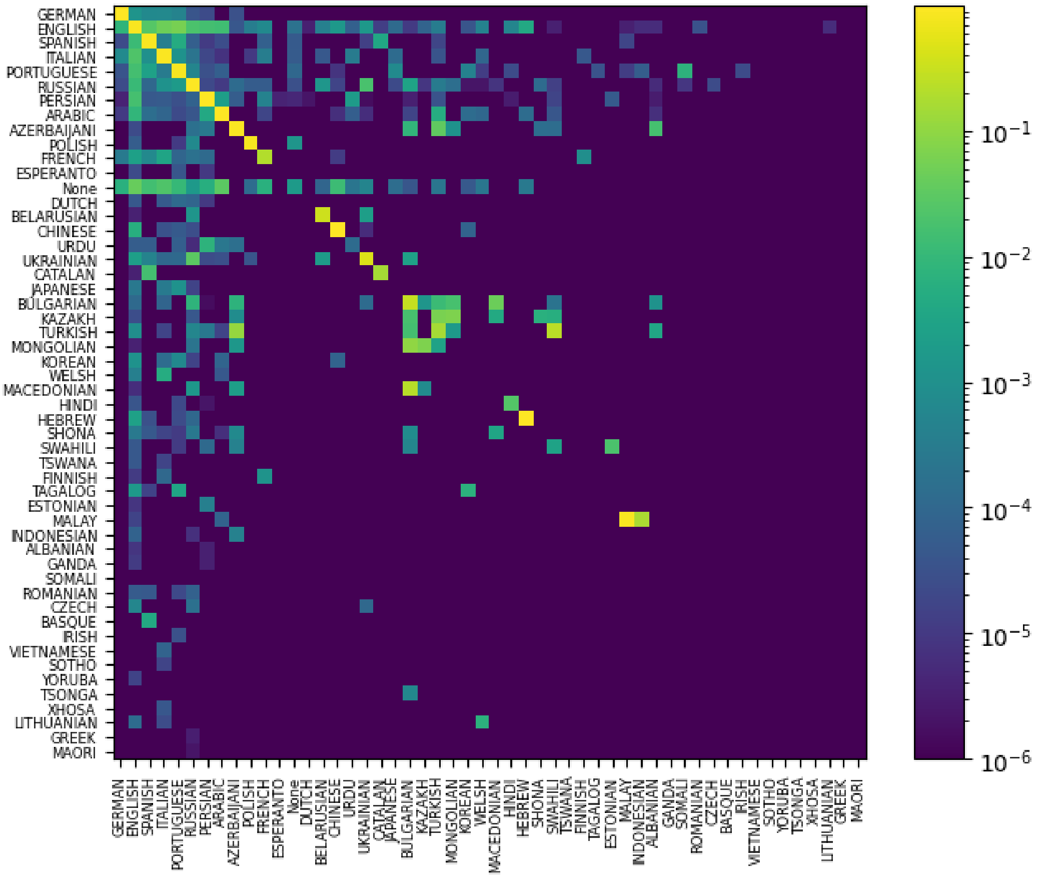


Figure 3.15: Heatmap of M^* on the log values. The zero entries are modified by adding 10^{-6}. A clear positive correlation is present, detected by high values on the diagonal. It is possible to see other interactions between languages with similar cultural backgrounds.

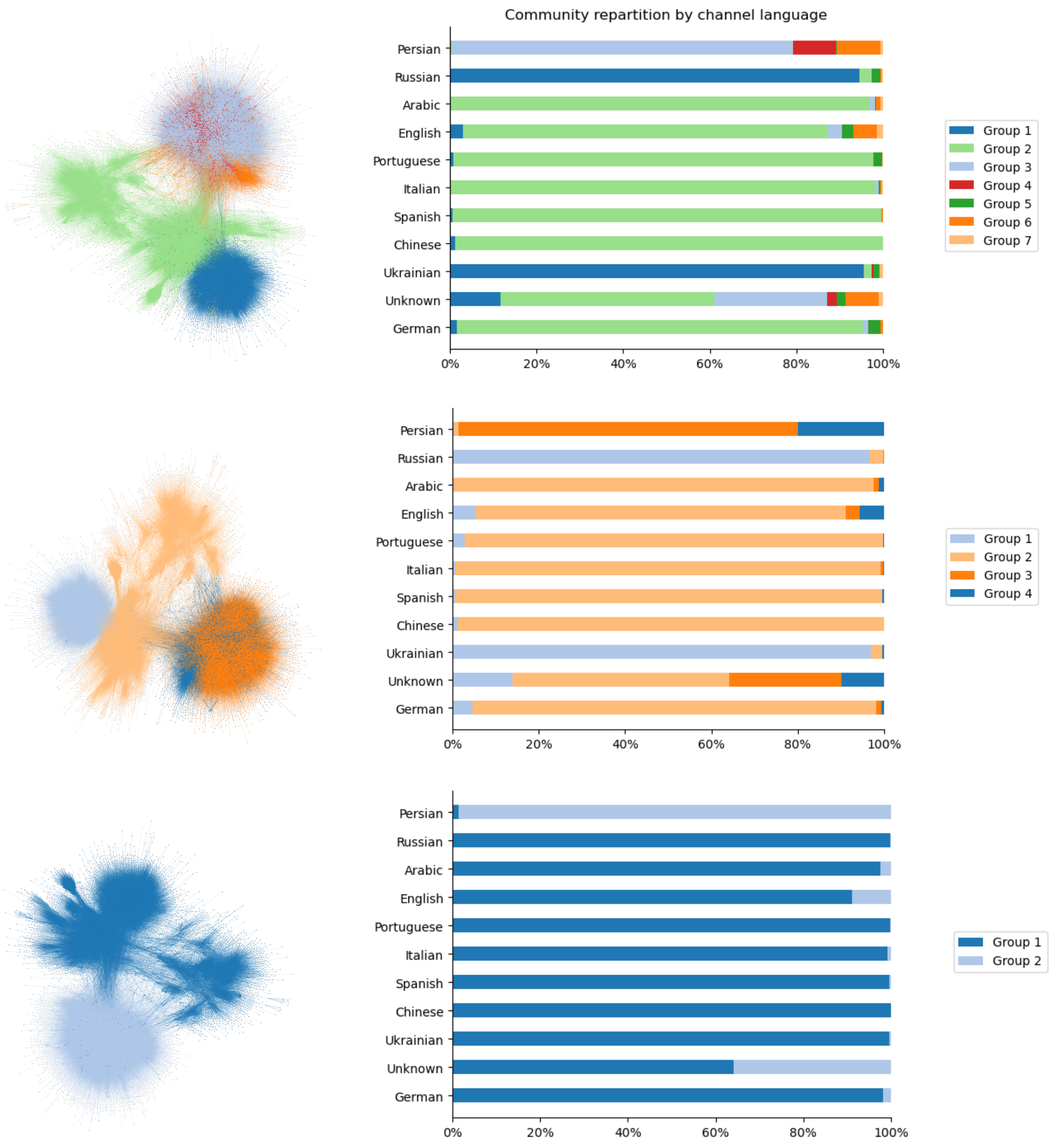


Figure 3.16: Respectively the first row corresponds to 7 communities detected, the second one to 4 and the last to 2. On the left the network is divided by colour into the detected groups. On the right it is shown the group membership across the 11 most spoken languages.

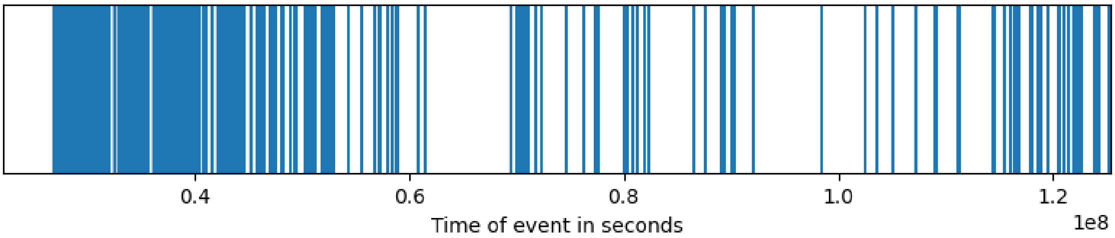


Figure 3.17: The event sequence of a specific chat over the total time window. Every event is represented with a straight blue line. Short inter-event times follow one another forming blue vertical bands, while longer ones follow each other, creating white bands in succession.

total time window spans 4 years from 2015 to 2019, which corresponds to around 10^8 seconds. A sequence of events for a specific node is represented in Figure 3.17. Already from this figure, it is possible to see how short IETs tend to follow short ones, which is detectable from the blue vertical bands.

Looking at the probability density function of IETs in Figure 3.18, it is possible to note that they follow a scale-free distribution. In particular, the distribution seems to follow a power-law with 2 different regimes. The exponents are estimated as $\alpha_1 = 1.00$ and $\alpha_2 = 1.69$. In the second regime, the slope is steeper than in the previous one. The density of τ seems to follow:

$$f(\tau) = \begin{cases} C\tau^{-1.00} & \text{if } 1 \leq \tau \leq 10^5 \\ D\tau^{-1.69} & \text{if } \tau > 10^5 \end{cases}$$

where C and D are the normalization constants. The switch of regimes happens around 10^5 seconds which roughly corresponds to 1 day, which is a natural timescale on which to observe a change in behaviour. Investigating the distribution of other real datasets [21], such as mobile phone call, short messages or email sequences, the IETs distribution resembles a power-law with the presence of 2 regimes where the switch point is again around 1 day.

3.3.1 Burstiness

Human activities are known to be bursty, so dominated by sequences of short inter-event times and very few long ones. This behaviour is often associated with the intrinsic bursty nature of humans caused by circadian patterns. Following a daily pattern, it seems natural that events will happen shortly after each other with longer pauses corresponding to nighttime. Nonetheless, in many cases such as the Short Message or Mobile Phone call dataset [18], Jo et al. have shown that this behaviour remains after performing a de-seasonalization procedure. Telegram is a social network too, thus we expect similar results. First of all, the coefficient B is computed, then in Section 3.3.3 a de-seasonalization procedure is applied.

B has been computed for every node's IETs sequence, and the distribution of such coefficients is plotted in Figure 3.19. The distribution is centered around a value bigger than 0, precisely $\bar{B} = 0.304$, suggesting that the underlying process is not

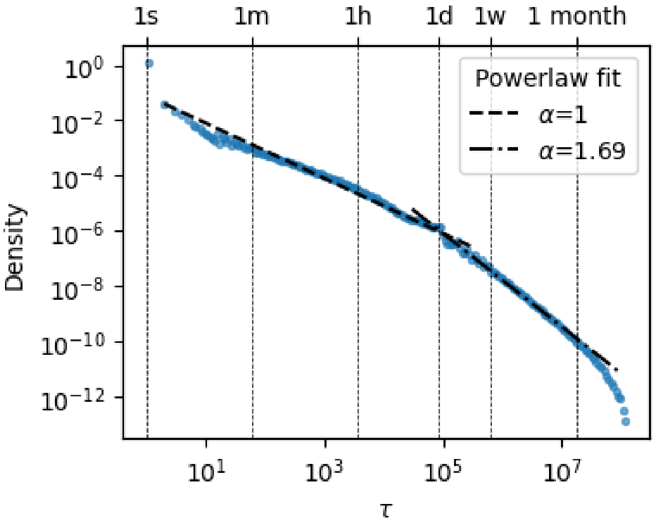


Figure 3.18: The probability density function function of IETs. The vertical lines indicate respectively 1 second, 1 minute, 1 hour, 1 day, 1 week and 1 month. Every $\tau = 0$ has been considered as 1, which explains the high value of the first point. There are two regimes with a transition point at around 1 day.

Poissonian. Moreover, the curve is shifted towards the right. In this computation, in order to have at least 2 IETs, all the event sequences with at least 3 samples are considered, but for too short series the value of B cannot be considered reliable. Intuitively, these small series might be the explanation behind a distribution of B not so close to 1 as expected from a scale-free distributed τ. This intuition is confirmed by looking at the second plot of Figure 3.19, where just the 1 000 series with most events are studied, showing a distribution much more shifted towards $B = 1$.

Considering the series of all inter-event times, the overall burstiness is $B = 0.812$, suggesting a high level of it, which is in line with the values of other human activities [14].

The LV distribution is centered around 1.45, confirming the presence of burstiness in the time sequence. Its density is plotted in Figure 3.20.

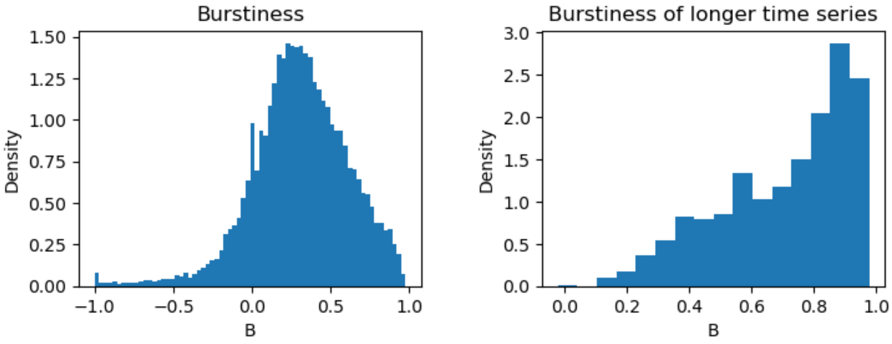
3.3.2 Temporal correlation

To study the temporal correlation of the time sequence, as suggested in Subsection 2.5.2, the memory coefficient M and the burst train sizes are analyzed.

Memory coefficient

As done for the other measures, M is computed for every chat with at least 3 events. However, its distribution has an odd behaviour at the tails with high peaks at the boundaries. This phenomenon is caused by short time series which do not provide a reliable value of M. For this reason, it is more informative to consider chats with at least 7 forwarded messages.

The corresponding distribution is plotted in Figure 3.21 and it is centered around



(a)

(b)

Figure 3.19: Distribution of the burstiness coefficient B in two different cases: a) For every inter-event time sequence. b) For the 1000 time sequences with most events. Both are shifted towards a positive value, denoting the presence of burstiness. The right one shows that longer sequences have a more bursty behaviour.

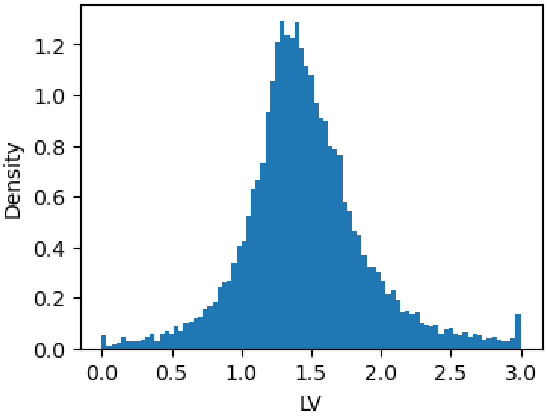


Figure 3.20: Distribution of LV computed for every active chat with at least 3 events.

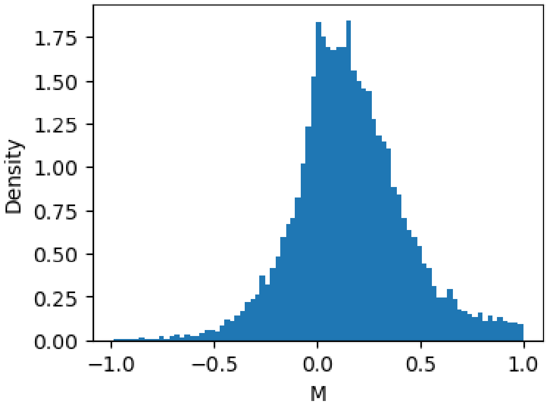
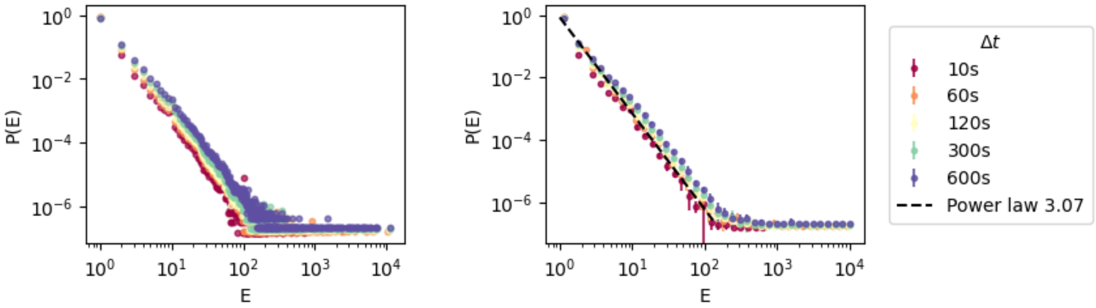


Figure 3.21: Distribution of M in chats with at least 7 messages forwarded.



(a)

(b)

Figure 3.22: Burst train size distribution: a) Distribution of $p(E)$ for different Δt. b) Log-binning of the distribution of $p(E)$, in black the fitted power-law. In both, there are plateaus caused by the intrinsic limit on the burst train sizes.

0.161. This suggests a small positive correlation between successive IETs. As seen in many real datasets of human activities [14], the memory coefficient is slightly higher than 0, which is the value expected for a Poisson process.

Burst train size

As seen in Subsection 2.5.2, if there is some type of correlation in the IETs sequence then we should detect it by studying the distribution of burst train sizes. Burst train sizes are computed for every chat for different Δt and then the results are considered together to plot the distribution. The Δt considered are 10 seconds, 1, 2, 5 and 10 minutes. The distribution, shown in Figure 3.22, resembles a power law with fitted exponent $\beta = 3.07$ for $\Delta t = 120s$. Varying Δt across its whole range of possible values, $p(E)$ shows a robust behaviour as visible in Figure 3.23. The shapes resemble again a power-law but with slightly different slopes. The power-law behaviour of $p(E)$ suggests that inter-event times are not independent and that there is some type of correlation between them. In particular, from the meaning of burst train size, it seems that short IETs come in succession.

It is possible to explore whether inhomogeneities in the in-strength of nodes can

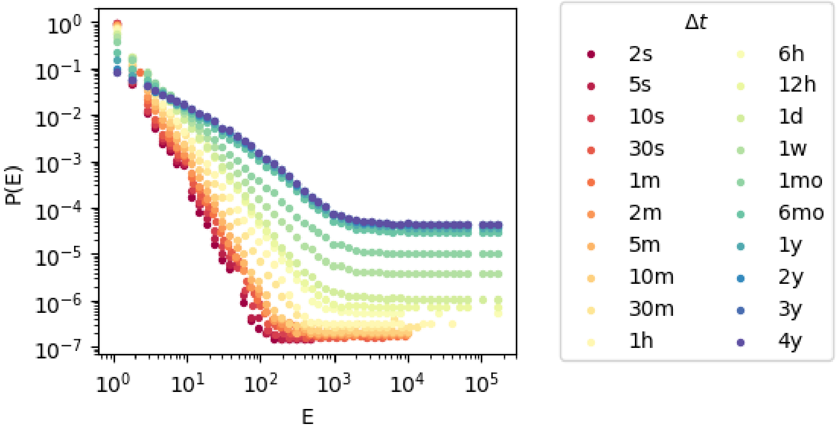


Figure 3.23: Log-binning of $p(E)$ for a wide range of Δt. The power-law looking shape is robust with respect to Δt. As Δt approaches 4 years, the distribution converges to the one of the in-strength.

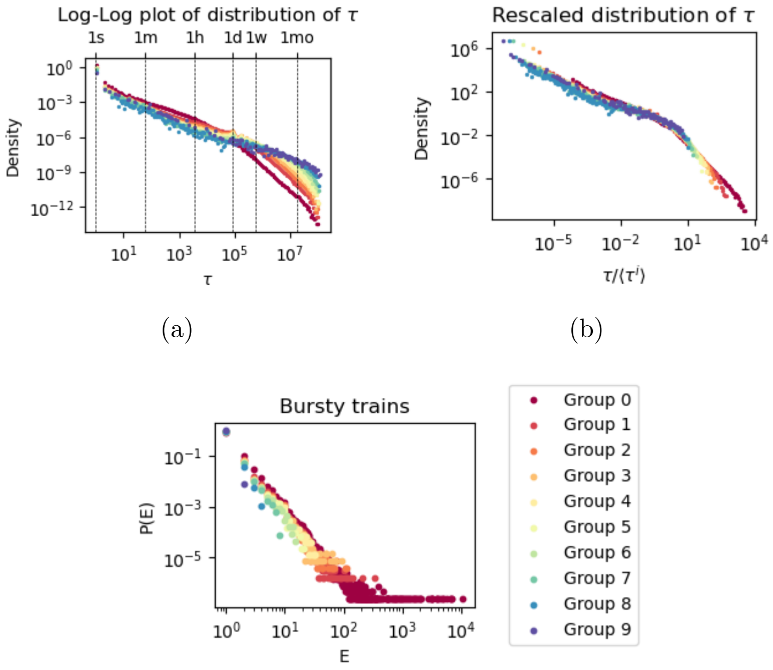
play a role in the burst train sizes and in the distribution of inter-event times. To check this, the chats are segmented by in-strength in groups of equal size, and the distribution of previous measures is recomputed per class. Concretely, chats are divided into 10 groups of 2 265 elements each based on their activity. Some basic statistics of the groups are summarized in Table 3.4.

In Figure 3.24, the distribution of inter-event times divided per group is plotted. The tails of the distributions and group 0, which is the group with the most events, constitute the biggest differences across the functions. Regarding the tail, dissimilarities can be caused by the following tendency. Chats with low activity levels have longer inactive periods between bursts which induces a long tail in $p(\tau)$ with a delayed drop. On the contrary, highly active chats have shorter τ which implies a smaller frequency of long τ. This actually explains the two discrepancies at the same time, indeed group 0 is made by nodes with higher levels of activity. In particular, the amount of activity of other groups is comparable, but this does not hold for group 0. In that set, the in-strength of some nodes is of some order of magnitude higher than the one of other groups. This will cause extreme behaviours of group 0 with respect to others and it is consistent with what we see in the plot. The red line has indeed higher frequencies for smaller τ and much lower for bigger ones with respect to all other lines.

Interestingly, scaling the times with respect to the average τ of each group, the distributions turn out very similar across groups. Finally, no difference between groups is seen in the burst train size distribution. Clearly, the burst train size for groups with lower activity is shorter, but this is a difference induced by the different intrinsic limits in the train sizes.

Number of events	Group 0	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Group 9
Min	579	206	98	52	28	16	8	4	2	1
Max	184 876	578	205	98	52	28	16	8	4	2
Mean	2 666.64	345.55	144.24	72.61	39.18	21.45	11.7	6.06	2.86	1.19

Table 3.4: Statistics of the number of events per chat across different activity bins.



(a)

(b)

(c)

Figure 3.24: Analyses on different groups: a) Distribution of inter-event times of chats. b) Re-scaled plot. c) $p(E)$ for $\Delta t = 120s$.

It can be concluded that having considered all nodes together as one group has not influenced neither the detected correlation nor the recognition of the distribution of inter-event times as scale-free. This result justifies the choice of a time model with the same parameters for every simulated chat, as we will see in more details in the next chapter.

Memory function

In Telegram, the memory function computes the probability that a chat will forward another message within a Δt time frame after it executed n events in the actual burst train. In the previous section, we have seen that $p(E) \sim E^{-\beta}$ with $\beta = 3.07$ and, according to the computations in Subsection 2.5.2 regarding $p(n)$, we expect the memory function to satisfy the prediction $p(n) = \left(\frac{n}{n+1}\right)^{\beta-1}$.

As can be seen in Figure 3.25, the memory function satisfies indeed the predictions. The fit is computed for train sizes corresponding to different Δt using a nonlinear least square procedure. The fitted values for ν are shown in Table 3.5.

Δt	**10s**	**60s**	**120s**	**300s**	**600s**
ν	2.17	2.14	2.10	2.11	1.87

Table 3.5: Fitted exponent ν of the memory function for different Δt.

The estimation changes a bit depending on Δt but all of them range around values such that $\beta = \nu + 1$.

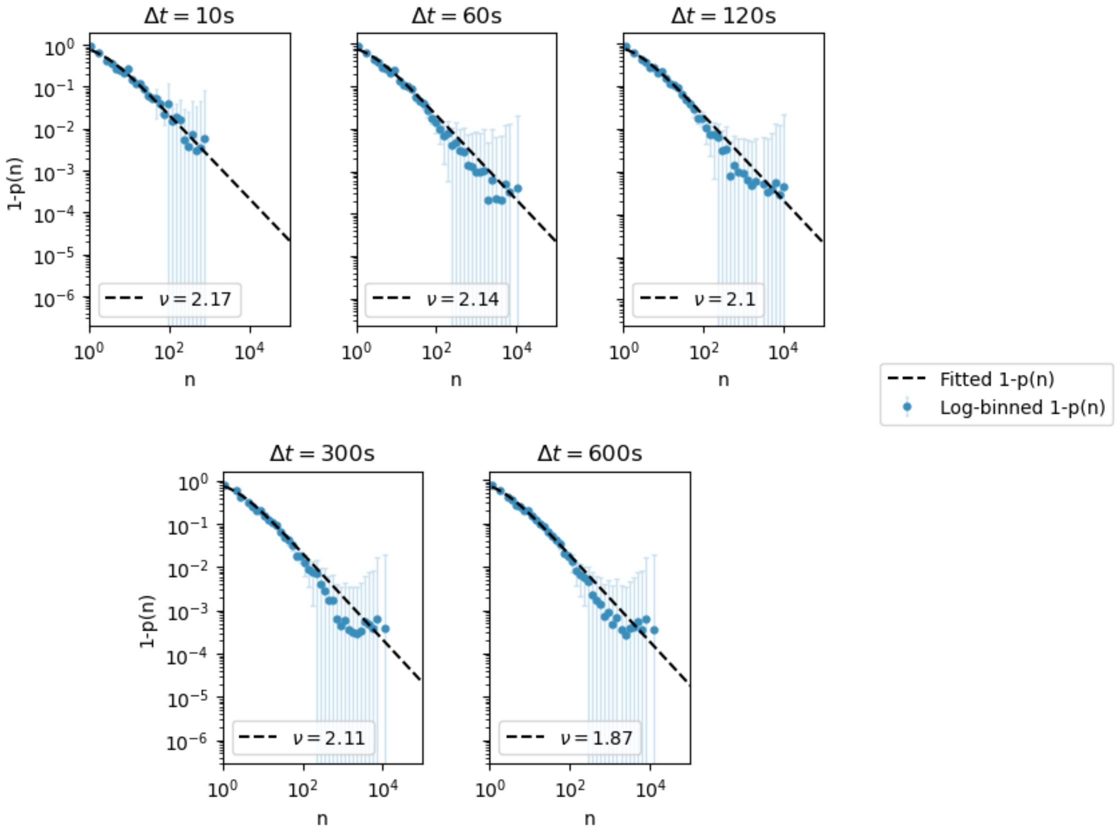


Figure 3.25: Plot of the complementary memory function $1 - p(n)$. The log-binning of the dataset estimation of $1 - p(n)$ is plotted using the blue dots and the standard deviation of the log-binning is plotted as light blue lines. The black dashed line is the fit of the function $1 - \left(\frac{n}{n+1}\right)^{\nu}$. Data seems to fit well to the predicted function. With different Δt considered, the best fitting ν varies a bit, always ranging around $\beta - 1$.

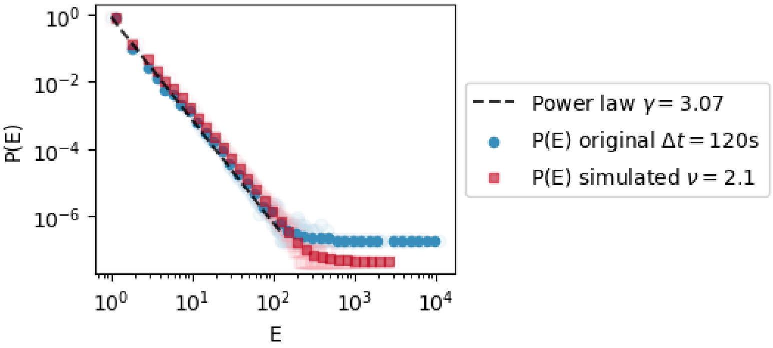


Figure 3.26: After simulating burst trains under $p(n)$ with $\nu = 2.1$, the log-binned $p(E)$ is plotted as red squares. The blue dots represent $p(E)$ of Telegram with $\Delta t = 120s$ and the dashed line is the power-law fit from Telegram data.

As said before, assuming a specific formula for the memory function, it is possible to prove that $p(E) \sim E^{-\beta}$. Simulating a process with $p(n) = \left(\frac{n}{n+1}\right)^{\nu}$ with $\nu = 2.1$, the obtained $p(E)$ of the simulated burst trains follows a power-law which is very close to $p(E)$ of the original data, as can be seen in Figure 3.26. This result suggests that $p(E)$ can be reproduced starting from a burst train process based on $p(n)$. Note that again $\nu + 1 = \beta$ approximately.

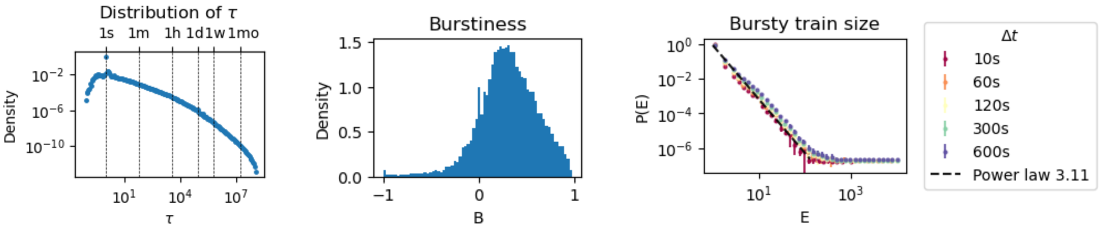
Both the good fit with data and the correspondence between the shape of $p(E)$ and $p(n)$ hint that the memory function is a driving force of the Telegram network of forwarded messages. These suggestions will then be used to develop a model for the inter-event times.

3.3.3 Deseasonalization

To check whether the obtained results are induced by circadian patterns, the deseasonalization procedure presented in Subsection 2.5.3 is performed.

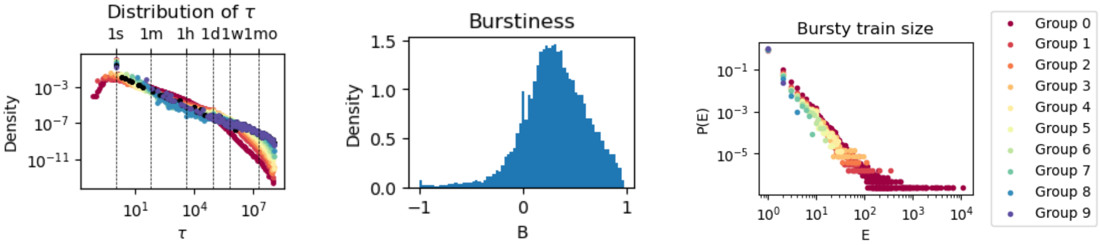
One group

In this subsection, consider A as the set of all chats and T as a 1 day period. The distribution of IETs resembles a power-law with approximately two regimes. The distribution of B is similar to the one before deseasoning and it is centered again around 0.304. The value of B for the overall series is 0.812. The distribution of burst train sizes follows a power-law behaviour with a fitted exponent very close to the previous one: $\beta = 3.11$. Overall, the aim of this study was to show that the qualitative behaviour of the series remained the same after deseasoning. From what we can see in Figure 3.27, circadian patterns play no main role in burstiness and temporal correlation. Thus, in Telegram, humans have these behaviours independently on the daily rhythms they follow.



(a) (b) (c)

Figure 3.27: Statistics of the deseasonalized inter-event times series: a) Distribution of τ's. b) Distribution of B. c) Distribution of bursty train sizes.



(a) (b) (c)

Figure 3.28: Statistics of the deseasonalized inter-event times series divided per activity groups: a) Distribution of τ's. b) Distribution of B. c) Distribution of bursty train sizes.

Activity based groups

Here, we will perform a deseasoning based on activity groups. In particular, recall the division in groups of Table 3.4, where we had 10 groups of chats based on their activity. The procedure is then carried out for each group individually and the period T is again 1 day.

As can be seen in Figure 3.28, the IETs distribution resembles again a power-law. Note that, in that plot, the black dots represent the same distribution but for the original time series. It is then immediate to see that just a few black dots are visible while the majority are covered by the colored ones. The burstiness is again shifted towards the right, centered again approximately around 0.304. Finally, the burst train size divided per activity shows a power-law behaviour.

3.3.4 Reinforcement

From scientific collaborations to Twitter mentions [40], humans tend to create a friend circle and reinforce those ties rather than creating new ones. In particular, human relationships seem to suffer from an intrinsic limit in the number of connections a person has. This translates into a law for which the higher the number of contacts one has, the less likely it is to form new ties. It is possible to encapsulate this idea as done by Ubaldi et al. in [40]. In the Telegram network, we refer to reinforcement process as the tendency of a channel to forward from a chat it has

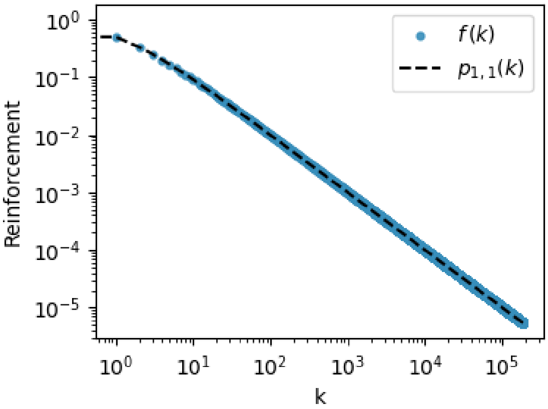


Figure 3.29: In blue the estimated reinforcement function from data. In black the fitted function with $b, c = 1$.

already forwarded from in the past, rather than from a new one. It is possible to define this process via the function $f(k)$ which represents the probability to create a new link rather than reinforce an old one, given that we have already contacted k other nodes. This function, which will be called reinforcement function, can be estimated from data as follows:

$$f(k) = \frac{n(k)}{e(k)}$$

where

- $n(k)$ is the number of times that a chat with k contacts forwards a message from a chat it has never interacted with.
- $e(k)$ is the total number of forwards performed by chats when they have k contacts.

These two quantities are computed for all nodes and then summed together.

In [40], they suggest to fit this function as

$$p_{c,b}(k) = \left(1 + \frac{k}{c}\right)^{-b}$$

Using nonlinear least squares approximation, c and b are estimated from data as 1. The function $p_{1,1}(k)$, plotted in Figure 3.29, fits perfectly the data. This suggests that the reinforcement process is indeed a driving force of Telegram relations.

3.3.5 Summary

Throughout this section, it has been shown, in different ways, how inter-event times have fat tails and show temporal correlation. These phenomena, which intuitively could be explained by the circadianity of human activities, are instead intrinsic to the behaviour of people. Moreover, the underlying process seems to be driven by the memory function, which generates a series of bursts and moments of calmness, and by the reinforcement of old ties.

3.4 Stationarity

To investigate more deeply the structure of the network, it is possible to repeat the analyses above considering a subset of the total time window. First, since the dataset ranges across 4 years, the data has been split into 4 parts depending on the year. The network and time analyses have been repeated showing consistent results throughout the years with just a few differences. In particular, the Russian language appears in the dataset in the last 2 years and the burstiness is slightly higher in the first two years. Then the dataset has been considered before and after a relevant event that could have induced a change in the structure of Telegram: the introduction of discussion chats. These chats are groups in which all the participants can comment on what has been sent in the main channel. In particular, many of these chats automatically forward content from the original channel. Their introduction implied a spike in the number of forwards and it could have changed the patterns observed for this action. Indeed, the forwards made in these chats resemble more the phenomenon of sending messages rather than forwarding them and it may present different characteristics. From qualitative analyses, the network properties observed before are found again in this setting. Regarding the temporal aspects instead, the burstiness seems to decrease after the introduction of discussion chats but still has high levels, while the correlation between τ's remains untouched. The results of this analysis suggest that the network is stationary with respect to time. Thus, it is reasonable to develop a model with parameters which do not depend on the age of the network.

CHAPTER 4

Model

In the previous chapters, it has been possible to characterize the Telegram network under different aspects. Both from a temporal and topological perspective, the dataset has shown to follow some specific patterns, which can be seen as the driving force of the dynamic or as a consequence of the underlying mechanisms that regulate the forwards. Those results are used to develop a model for Telegram data, which consists of two parts: one for the temporal aspect and the other for the network structure. In a first step, the timing at which chats forward content is simulated with a time model. Then, based on the simulated inter-event times (IETs), the network model is deployed to choose from which chats the forwards are made. After presenting both, the results of various simulations are shown and commented on.

4.1 Time

From the temporal point of view, the dynamic of the chats is bursty and correlated. Moreover, it is driven by the memory function $p(n)$. The idea behind the time model comes from [21] of Karsai et al. where they propose a model to reproduce both correlation and fat-tailed distribution of inter-event times. The structure of this model is summarized in Figure 4.1. A chat can be in state A, which is a quiescent state, or in state B, which is an excited state. Once it reaches a state, it has to perform an event there, i.e. forward a message, under the laws of that state. After performing an event, it is possible to stay in that state or move depending on some probabilities. This model can be adapted better to Telegram data by introducing the birth date of every chat.

The model consists in the following steps:

- At first, the state is chosen randomly and the event time is set to the birth date of the chat.
- If the current state is A, then an event is performed there. The counter of actual inter-event time t is set to 1 at first. Then, at every time step, an event is generated with probability $1 - f_A = 1 - \left(\frac{t}{t+1}\right)^{u_A}$, so that the current t is the inter-event time from the previous event. Otherwise, we increment t by one and we go back making a random draw until an event is performed. After an

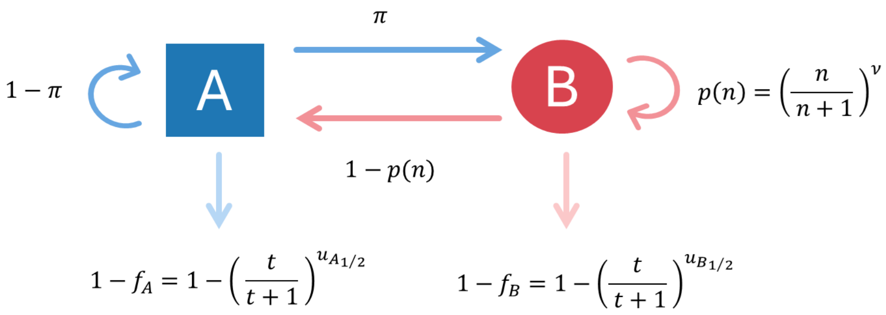

π
$1-\pi$
A
B
$p(n) = \left(\frac{n}{n+1}\right)^{\nu}$
$1-p(n)$
$1 - f_A = 1 - \left(\frac{t}{t+1}\right)^{u_{A1/2}}$
$1 - f_B = 1 - \left(\frac{t}{t+1}\right)^{u_{B1/2}}$

Figure 4.1: Model of the event times. A is the quiescent state, B is the excited state.

event has been performed, with probability π we go to state B, otherwise, we stay in A.

- If the current state is B, then an event is performed there as in A, but the timing is regulated by $f_B = \left(\frac{t}{t+1}\right)^{u_B}$. Let n be the number of consecutive events that happened in state B since the last time we entered that state. Then with probability $p(n) = \left(\frac{n}{n+1}\right)^{\nu}$, which is the memory function depending on ν introduced in Definition 2.5.5, we stay in state B, otherwise, we switch to state A.

- We continue until the simulation time span is reached.

This model can generate heavy-tailed distributed τ's and temporal correlation. The presence of the memory process, which regulates the transition from state B, introduces correlation between the IETs generated and, thanks to the observations in Subsection 2.5.2, we expect the model to recreate long bursty trains with a fat-tailed $p(E)$. The assumption behind the choice of f_A and f_B is that the more time a chat has waited to forward a message, the longer the admins are going to wait. Once we are in a certain state, the probability of having an inter-event time τ is:

$$p_{A/B}(\tau) = f_{A/B}(1) \cdot \ldots \cdot f_{A/B}(\tau - 1) \cdot (1 - f_{A/B}(\tau)) = \tau^{-u_{A/B}} - (\tau + 1)^{-u_{A/B}} \quad (1)$$

which is a mixture of power-laws and has a heavy tail, as shown in Figure 4.2. The distribution of the τ simulated by the model is not going to follow this exact distribution, because it would be the result of the combination between the one of state A with the one of state B. The combination between the two is difficult to predict since it also depends on π and $p(n)$. Nonetheless, we can expect to see a heavy tail distribution.

In Telegram, the IET distribution follows 2 regimes, as seen in Figure 3.18, shifting around 1 day, precisely $\tau_s = 10^5$. To account for that, the model above is modified by varying the value of u_A and u_B depending on the threshold of the regime. In particular, it is possible to do so by replacing u_A and u_B with 2 parameters each. In this way, if the current IET in a state is below the threshold, that we set to τ_s, then $u_A = u_{A_1}$ while if it is above $u_A = u_{A_2}$. To change the value continuously, for

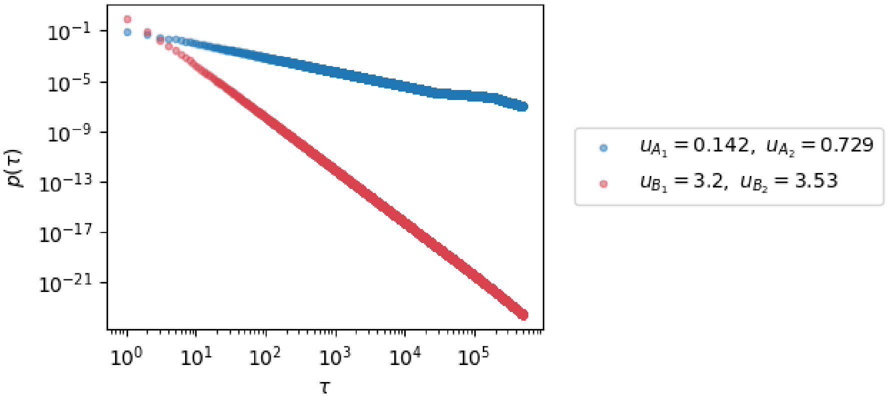


Figure 4.2: The probability mass function of IETs computed in equation (1) is plotted. For each function we have 2 parameters according to the observation regarding the two regimes of Telegram IETs distribution. In blue $u_{A_1} = 0.142,\ u_{A_2} = 0.729$ and in red $u_{B_1} = 3.2,\ u_{B_2} = 3.53$. The parameters used are the one selected for the time model.

a range around the threshold, we vary u_A linearly:

$$u_A(t) = \begin{cases} u_{A_1} & \text{if } t < t_1 \\ \frac{u_{A_2}-u_{A_1}}{t_2-t_1} \cdot t + \left(u_{A_1} - \frac{u_{A_2}-u_{A_1}}{t_2-t_1} \cdot t_1\right) & \text{if } t_1 \leq t \leq t_2 \\ u_{A_2} & \text{if } t > t_2 \end{cases}$$

and analogously for u_B.

The parameters of the model are then u_{A_1}, u_{A_2}, u_{B_1}, u_{B_2}, π, ν. Technically, t_1 and t_2 are parameters too, but they are chosen to be around the threshold. In this case, they are fixed as $t_1 = 10^5 - 7 \cdot 10^4$ and $t_2 = 2 \cdot 10^5$. In a more general framework, they can be interpreted as parameters too. As seen in Subsection 3.3.2, it is reasonable to set $\nu = 2.1$. As suggested by the paper, u_A is responsible for the exponent of the power law of simulated τ's, in particular $u_A + 1 = \alpha$. Thus, the region of possible values of u_{A_1} and u_{A_2} will be reduced according to this additional information. Moreover, since A is the calm state, then $u_A < u_B$ and since the slope of the second regime appeared steeper, then $u_{A_1} < u_{A_2}$ and $u_{B_1} < u_{B_2}$. The domain of definition of every parameter is set as follows:

$$u_{A_1} \in [10^{-4}, 0.5],\ u_{A_2} \in [0.5, 0.8],\ u_{B_1} \in [2, 7], u_{B_2} \in [2, 8],\ \pi \in [0.08, 1]$$

These regions have been selected based on the remarks above and exploratory analyses.

To fit these parameters, the method of Bayesian optimization presented in Subsection 2.1.2 is used to minimize the difference between the IETs density distribution of the data and the same function obtained from the model. Consider the log-binning function of the IETs density distribution, then a simulation of the model is run with some parameters for 5 hypothetical chats and the two log-binned distribution functions of τ are compared. Note that to do this comparison the same log-bins are applied and any bin where no data or model observation is present is not considered. The loss function is computed as follows:

$$L = \frac{1}{n_b} \sum_i \left[\log(y_i^{data}) - \log(y_i^{model})\right]^2$$

where i stands for the ith log bin, y_i is the log-binned density and n_b is the number of log-bins. The difference between logarithms is used in order to give all bins the same weight and not overly consider the first ones. The parameters are found using Bayesian optimization with 1000 initial points and 100 exploratory ones per iteration. The procedure stopped when the value L on the current best parameters has been stable for at least 4 steps. To do so, it has been used the python package presented in [28] with constrained optimization [13]. The values found are:

$$u_{A_1} = 0.142,\ u_{A_2} = 0.729,\ u_{B_1} = 3.20,\ u_{B_2} = 3.53,\ \pi = 0.157$$

Note that the parameters are the same for every chat and constant with respect to time. This choice is consistent with the results of Sections 3.3.2 and 3.4, where it has been shown that the correlation and heavy tail behaviour of IETs is independent on the activity level and age of a chat.

As we will see in the following, the approach above may not be able to reproduce correctly the in-strength distribution of the directed network. To address this issue, it is possible to add a limit to the number of events that each chat can produce. To do so, a simulation of the in-strengths is sampled from truncated power-law, which is a power law distribution with an exponential cutoff. This choice is made to have limited values of the in-strength. Then, each chat is assigned an in-strength value according to a specific order. First, the birth dates of the nodes are arranged in ascending order, from the youngest to the oldest. Then, the sampled in-strength values are arranged in descending order, with the largest ones assigned to the oldest nodes and the smallest ones to the youngest nodes. Finally, the time model runs until either the event time reaches the maximum value or the total number of events performed matches the associated in-strength value. For simplicity, we are going to call this version of the time model constrained and the first one unconstrained.

The two models are run for every active chat of Telegram, i.e. that has forwarded a message in the total period, which are 22 650. The simulations will give, for every simulated chat, a sequence of times at which the node forwards a message. On top of these simulations, a network model is needed to reproduce the interactions between chats.

4.2 Topology

The network both as undirected and directed has shown different interesting behaviours. They are both characterized by scale-free distribution of the strengths, a high clustering and language assortativity. In addition, also degree correlation has shown particular results with a neutral pattern in the undirected case and some assortativity in the directed one. The process behind Telegram's forwards seems to be driven then by some major causes: clustering, language assortativity and reinforcement of old ties.

The model that will be considered to recreate the topology of the network is inspired from Laurent et al. [22] which is based on 3 mechanisms to develop the network: node deletion, reinforcement process and closure. The closure mechanisms are able to recreate two different human tendencies. The first, called focal closure, regards the propensity to choose a new chat to forward from, recreating the formation of links between nodes with shared interests or features. Instead, the second, known as triadic closure, is connected to the tendency to contact a friend of a friend, which implies high clustering values. The node deletion phenomenon is not present in the Telegram dataset, thus it is discarded. Note that this model was thought for undirected networks, while here it is adapted to create directed links. Among the other main mechanisms, another one is introduced which regards language assortativity. The nodes will be divided into groups, which in this case are based on languages, and each group will be assigned an assortative coefficient which regulates how much chats with the same attribute interact together. Additionally, every chat is associated with its real creation date and at every step newly created nodes enter the network, making its size increase over time.

At each time step, only active nodes are considered. The active nodes are nodes who are going to forward a message at that time step and this information is provided by the time model. These nodes can forward content from non active chats too. Let n_i be the number of contacts of node i up to the considered time step and p_{ji} be the ratio between the number of times i has forwarded content from j and the total number of forwards of i. The flowchart of the model is represented in Figure 4.3 and the steps of the model can be summarized as it follows.

- Let i be an active node at current time step t, then with probability $\frac{n_i}{n_i+1}$ it is going to forward a message from one of its old tie, otherwise it forwards from another chat, forming a new tie.

- In the first case, among its old ties, i forwards from node j with probability p_{ji}.

- In the second case i has to choose a new node to connect with. At this point, apart from particular cases, chat i tries to forward from a friend of its friends. Thus, a neighbour j of i is selected according to p_{ji}, then, with probability p_{kj}, a neighbour k of j is sampled. Now, with probability p_{TC}, i follows the triadic closure mechanism, so it forwards from k to close the triad, otherwise, it follows focal closure. In the latter case, i chooses a random node it has never contacted before. If possible, with probability p_{SL}, i selects at random a node which speaks the same language, otherwise it chooses from all available chats.

Some specific cases which modify the flow of the model are the following. If i or the selected j have no neighbour, or if the only neighbour of j is i then i directly follows focal closure mechanism. If a node is considered to be active from the time model, but it cannot forward from anyone because it is the only node in the network, then it is considered inactive. Finally, if all the nodes in the network at some point are neighbours of i, then, when performing focal closure, neighbors are not discarded.

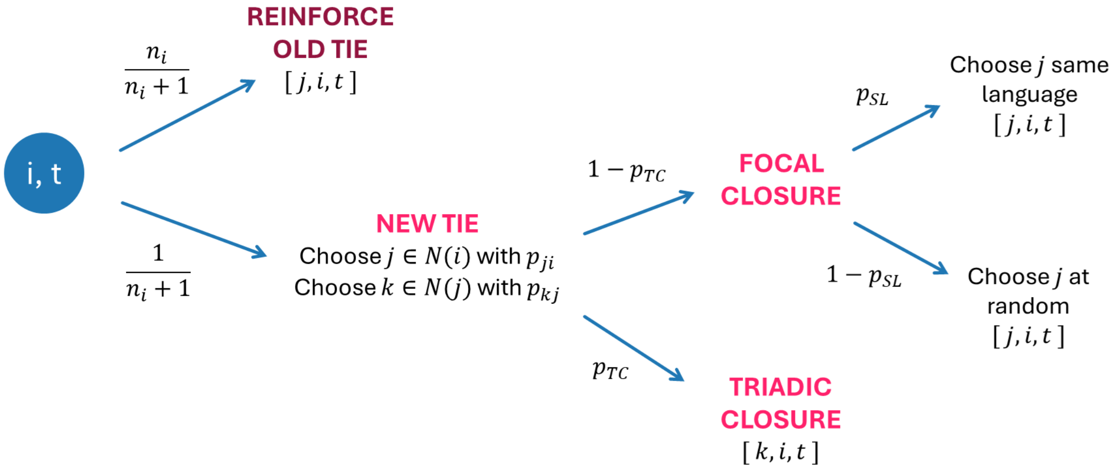


Figure 4.3: Topological model where n_i is the number of contacts of i until that time step, p_{ji} is the ratio of the number of times i has forwarded content from j to the total number of forwards made by i. The parameters p_{TC} and p_{SL} are to be set. $[j, i, t]$ refers to a forward from j to i at time t.

The parameters are p_{TC}, and p_{SL} which varies across groups. The first one is set to 0.9 after exploratory and qualitative analyses. This parameter affects the value of the clustering coefficient in the network, thus it can be chosen accordingly to obtain the desired C. The second one influences how a chat forwards from another one which speaks the same language. This parameter can be the same for every language, but, as seen in Figure 3.6, there is heterogeneity between languages, thus it is reasonable to fix one parameter value per group. The quantity to be estimated is the probability of a chat speaking a certain language to make a new contact with a node speaking the same idiom, we will refer to it as the assortative value. Consider the matrix M of Subsection 2.3.5, then these probabilities can be estimated by the diagonal of the matrix with entries $\frac{m_{ij}}{\sum_k m_{ik}}$. These entries, plotted in Figure a) of 4.4, have values between $[0, 1]$, with a peak at 0 and a smaller one near 1. Thus, a sensible modeling choice is to assume that the distribution of the assortative value is a $Beta(b_1, b_2)$. The mean and variance of this distribution are given by:

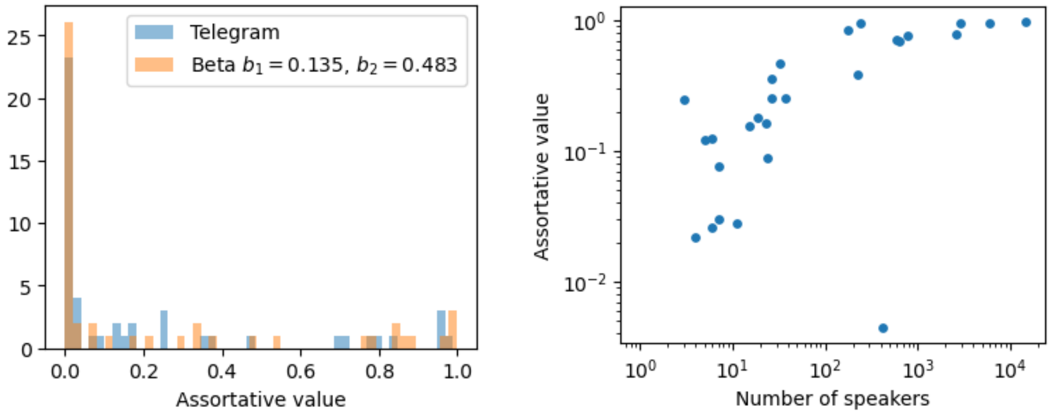
$$E[X] = \frac{b_1}{b_1 + b_2}$$

$$V(X) = \frac{b_1 b_2}{(b_1 + b_2)^2 (b_1 + b_2 + 1)}$$

Using the method of moments, we can estimate b_1 and b_2 using the sample mean estimate $\bar{x}$ and sample variance estimate $\bar{v}$, obtaining:

$$\hat{b}_1 = 0.135, \ \hat{b}_2 = 0.483.$$

A comparison between the sample of the assortative value and one obtained from a $Beta(\hat{b}_1, \hat{b}_2)$ is shown in Figure 4.4. Qualitatively, the two samples seem to have a similar behaviour. Since within the scope of this model, we only seek a qualitative



(a)

(b)

Figure 4.4: Analysis for the assignment of assortative values p_{SL}: a) In blue is represented the distribution of the assortative value in Telegram, and in orange the distribution for a sample of $Beta(0.135, 0.483)$. The two samples seem to have a similar distribution. b) The assortative value of each language in Telegram compared to the number of chats speaking that language. There appears to exist a monotonic relationship between the two.

reproduction of language assortativity, the assortative coefficient of every language is simulated with a sample from that $Beta$. Each idiom is assigned a sample from $p_{SL} \sim Beta(0.135, 0.483)$ depending on how many chats speak that language. Indeed, looking at Figure 4.4, it is possible to see that the assortative value depends almost monotonically on the number of chats speaking that idiom. For this reason, the beta samples are ordered and each value is assigned to a language according to the rank of the latter.

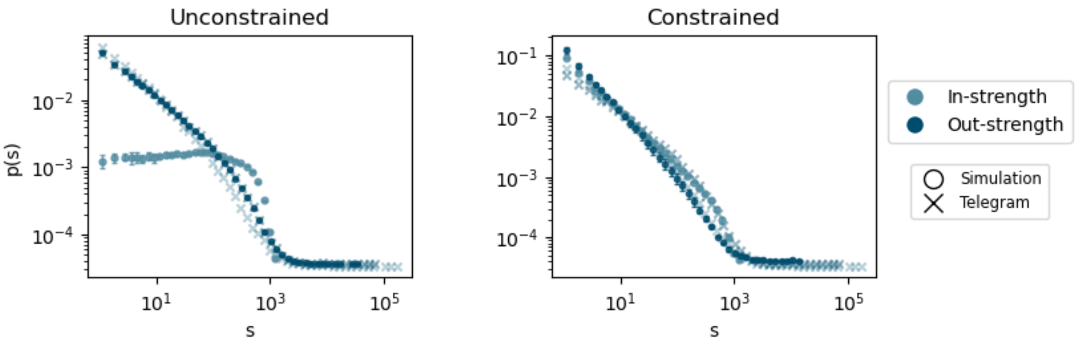
This model can now be run on the data generated by the time model. Each time a chat is active, the topological model returns from which chat the forward is made, basing its decision on reinforcement of old ties, triadic and focal closure and language homophily. Now, it is possible to use both models to simulate the Telegram network and analyze the results.

4.3 Simulations

This section is divided into two parts with the presentation of the results obtained from 10 simulations of the unconstrained and constrained models.

4.3.1 Unconstrained model

Overall, the model is able to reproduce the number of events of Telegram in the same time period, the scale-free behaviour of the degree and of the out-strength, the high value of the clustering coefficient both in the undirected and in the directed, the neutrality of the undirected network and the assortativity with respect to languages. Some properties averaged across the simulations are presented in Table 4.1. For



(a)

(b)

Figure 4.5: Log-binning of the PMF of in and out-strength of Telegram and the two models: a) Unconstrained b) Constrained. Telegram data is represented with crosses with lighter colours, while the simulations' results are indicated with errorbars. In the first one, $p(s^{in})$ has a flat shape, while $p(s^{out})$ resembles the one of Telegram. In the second one, both resemble the Telegram shapes.

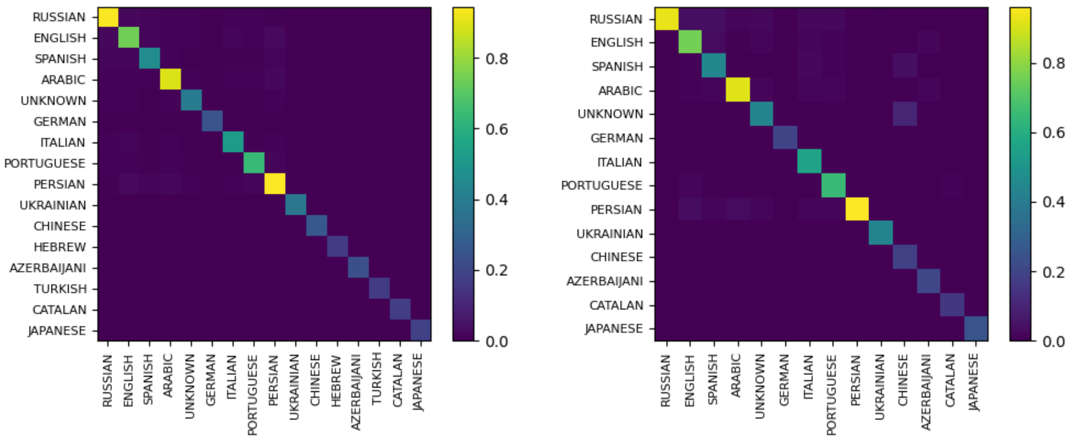
the majority of measures, the results are in line with the Telegram ones, however, the clustering of the directed and weighted network is much higher. Thanks to triadic closure, the average clustering coefficient with respect to the degree and to the strength is higher than in the randomized case. In the directed network, the simulations show assortativity for the out-out combination and neutrality in the other cases. As expected, IETs have a similar distribution and show temporal correlation, plotted in Figure 4.8, with the burst train size distribution which has a similar shape to the original one. However, it shows some limitations too. The in-strength distribution, plotted in Figure 4.5(a), appears flat, indicating a behaviour similar to a uniform distribution. This could have been expected since the number of events is given by the time model, which is the same for every chat, thus the in-strength should be similar across nodes. The only difference between chats is the birth date, which creates the effect of a uniform distribution, rather than a distribution centered around a specific value. For this reason, the constrained model is introduced which has an additional limitation on the in-strength.

4.3.2 Constrained model

This model achieves the desired scale-free shape for the in-strength distribution (Figure 4.5(b)) and all the other desired properties apart from one: the number of events. In Telegram, the total number of forwarded messages was around 7.5 million, however, this model produces around 2.8 million. The major drop obtained is clearly due to the in-strength limitation introduced.

As in the previous case, as can be seen in Table 4.1, C_{dw} is higher than the Telegram one, while the other measures are in line with the Telegram data. Again, C is higher than in the randomized networks. The shape of $k_{nn}(k)$ suggests assortativity in the out-out case and neutrality for other cases.

Since the topological model is the same both in the constrained and uncon-



(a) (b)

Figure 4.6: Extraction of the heatmap plot of the language assortativity matrix of a constrained model simulation: a) Undirected network b) Directed and weighted network. The language assortativity does not change between unconstrained and constrained setting, since the topological model is the same. The plots resemble the ones from Telegram, represented in Figures 3.5 and 3.14. The model is able to reproduce both assortativity, especially for popular idioms, and heterogeneity.

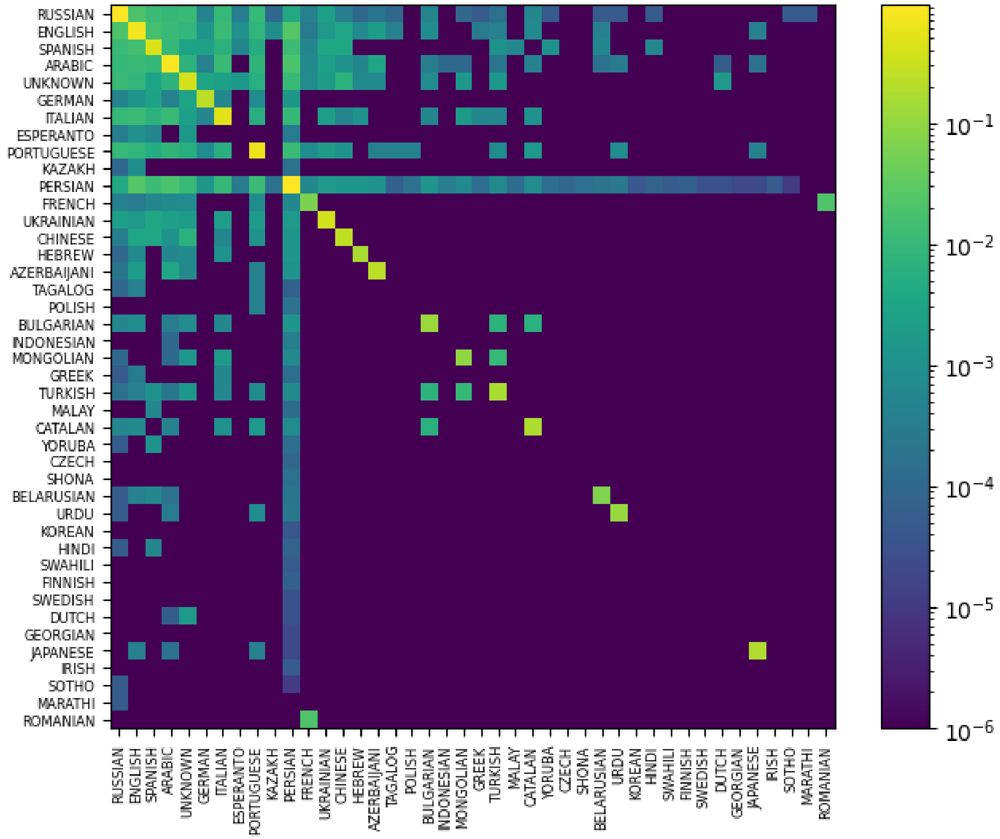


Figure 4.7: Log plot of the language assortativity matrix for the undirected network of one constrained model simulation.

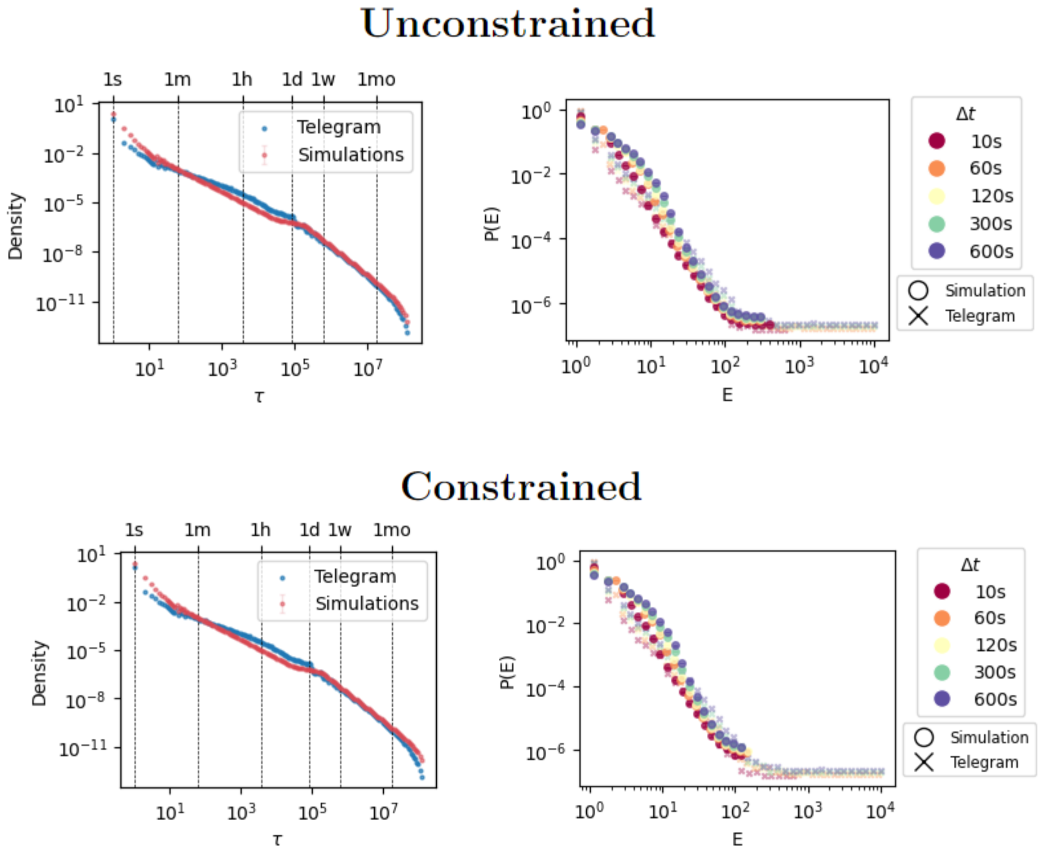


Figure 4.8: Temporal analyses of the 10 simulations of the unconstrained model. On the left, log-binning of the pdf of IETs of Telegram (blue) and of the simulations (red). The latter is averaged across simulations and the errorbars are the standard deviations. The simulations show a consistent distribution of τ which is very close to the Telegram one. On the right, PMF of the bursty train sizes in Telegram (cross) and in the simulations (circle). The simulations plot is averaged across the 10 samples. Both models are able to recreate the shape of $p(E)$ but fail in reproducing the long tail.

Metric	**Unconstrained**		**Constrained**		**Telegram**
	Mean	**Std**	**Mean**	**Std**	
N	$2.720 \cdot 10^4$	$0.001 \cdot 10^4$	$2.415 \cdot 10^4$	$0.635 \cdot 10^4$	$2.961 \cdot 10^4$
L	$3.24 \cdot 10^5$	$0.03 \cdot 10^5$	$1.66 \cdot 10^5$	$0.05 \cdot 10^5$	$4.72 \cdot 10^5$
C_u	0.236	0.009	0.274	0.007	0.248
r	-0.100	0.010	-0.093	0.015	-0.056
r_L	0.875	0.080	0.821	0.067	0.906
$\sum w_{ij}$	$7.57 \cdot 10^6$	$0.02 \cdot 10^6$	$2.80 \cdot 10^6$	$0.14 \cdot 10^6$	$7.50 \cdot 10^6$
C_{dw}	$3.18 \cdot 10^{-3}$	$0.51 \cdot 10^{-3}$	$2.34 \cdot 10^{-3}$	$0.22 \cdot 10^{-3}$	$1.06 \cdot 10^{-5}$
r_L^d	0.876	0.080	0.822	0.067	0.940

Table 4.1: Values of some measures computed on the Telegram network and on 10 simulations of the unconstrained and constrained model.

strained case, the language assortativity results are the same. In Figures 4.6 and 4.7, it is possible to see that the model recreates both assortativity and heterogeneity resembling the data from Telegram.

With both the constrained and unconstrained models, the IETs distribution shows an additional regime before 10 seconds which was present also in Telegram plots. This new regime is caused by the fact that $p(\tau)$ of state A is lower than the one of B, and before $\tau = 10$ the two functions intersect. It is possible to check this by looking at Figure 4.2. Correlation is correctly reproduced thanks to the memory function, however, both models, especially the constrained one, fail to reproduce the longer part of the tail of $p(E)$. This is due to the problem in reproducing correctly the in-strength. In the constrained model simulations, the number of total events is much lower than in Telegram, thus every chat forwards less messages which implies an intrinsic reduction of the maximum size of every burst train. The same happens for the unconstrained, where the total number of events is similar to the Telegram case, but fails to reproduce the in-strength distribution. In particular, the maximum in-strength of the chats in the unconstrained simulations is of the order of 10^4, while in Telegram the maximum is of the order of 10^5. This, clearly, influences the maximum size that burst trains can have.

By examining Definition 2.4.12, it is possible to note that the in and out-strength influence the value of C_{dw} too, as the weights are normalized by a larger value in Telegram compared to the simulations data. This may explain the higher values of C_{dw} in both models.

Finally, the reinforcement process is present in the simulated data as was expected by the definition of the topological model.

Both models are able to reproduce the majority of the Telegram properties, from static network measures to timing. At the same time, they fail to reproduce exactly every aspect, in particular, the clustering in the directed case, the in-strength and the number of total events. Nonetheless, the approach presented looks promising and for future works it may be possible to investigate which aspects are to be modified in order to recreate every tendency.

CHAPTER 5

Conclusion

Throughout this thesis, the phenomenon of forwards on Telegram has been deeply studied. To better understand the mechanisms behind it, network science has been deployed to capture the relationship between chats. At the same time, to produce a complete analysis, temporal aspects have been considered too. Telegram shows similar properties to other human temporal networks. It has been shown the presence of small-world dynamics, language assortativity and the tendency to reinforce old ties. Chats aggregate into communities based on the languages spoken, with groups typically forming around languages from specific geographical regions, such as Central Europe, Russian and Ukrainian-speaking areas, or Persian speaking regions. Interestingly, analyzing the network structure in different time windows, it features the same properties, thus we can conclude that the network is stationary with respect to its age. Regarding the timings of forwards, time sequences are bursty and correlated. Bursty trains of events are shown to be generated by a memory process. Furthermore, after performing a deseasoning of the time series, we have seen that the causes of these behaviours are not daily patterns.

From the results of the analyses, two models have been developed: unconstrained and constrained. These models are composed of two parts. One models the timing at which each chat forwards a message, while the other simulates from which chat the forward is made. The temporal and topological components are inspired by the models presented in [21] and in [22], adapted to successfully recreate Telegram's mechanisms. Both the unconstrained and the constrained models show promising results being able to reproduce the majority of the features of the social media. The second one was introduced to solve the in-strength problems of the unconstrained one, but it failed to reproduce the total number of events performed. Even if the models are not perfect in reproducing Telegram data, they are able to capture the majority of specific patterns seen in the social media. This suggests that the mechanisms on which the models are based, such as triadic and focal closure, reinforcement of old ties, language homophily and burst trains memory, are indeed driving forces of the forwarding phenomenon.

Future works may start from the structure of the models proposed in this thesis and test new aspects to introduce to obtain, for instance, an in-strength distribution more similar to the Telegram one. Then, these models can be used to study diffusion processes on the Telegram network with the freedom of varying the number of chats, the time window and the parameters to discover factors that may facilitate or not the spreading. For instance, a study regarding misinformation can be made on top of the simulations. As done in [8], the authors tested the effects of mitigation strategies for disinformation on networks generated by known models and on real-world networks too. Moreover, another aspect that can be introduced is content recognition of the messages sent in a chat, to retrieve the topic of discussion of each channel. In this thesis indeed, the content has not been taken into account, however, as done for languages, it may be interesting to study whether chats aggregate into communities based on the topic they discuss. Another direction could be to study whether the structure of the network or the temporal dynamics change with respect to the main topic of the channels.

Bibliography

[1] Jeff Alstott, Ed Bullmore, and Dietmar Plenz. "powerlaw: A Python Package for Analysis of Heavy-Tailed Distributions". In: *PLoS ONE* 9.1 (Jan. 2014). Ed. by Fabio Rapallo, e85777. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0085777.

[2] Albert-László Barabási and Réka Albert. "Emergence of Scaling in Random Networks". In: *Science* 286.5439 (1999), pp. 509–512. DOI: 10.1126/science.286.5439.509.

[3] Albert-László Barabási and Márton Pósfai. *Network science.* Cambridge: Cambridge University Press, 2016. ISBN: 9781107076266 1107076269.

[4] Jason Baumgartner, Savvas Zannettou, Megan Squire, and Jeremy Blackburn. "The Pushshift Telegram Dataset". In: *Proceedings of the International AAAI Conference on Web and Social Media* 14.1 (May 2020), pp. 840–847. DOI: 10.1609/icwsm.v14i1.7348.

[5] Fabrício Benevenuto and Philipe Melo. "Misinformation Campaigns through WhatsApp and Telegram in Presidential Elections in Brazil". In: *Communications of the ACM* 67.8 (2024), pp. 72–77.

[6] Stefano Bonaccorsi. *Stochastic Processes.* 2021.

[7] Eric Brochu, Vlad M Cora, and Nando De Freitas. "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning". In: *arXiv preprint arXiv:1012.2599* (2010).

[8] David J Butts, Sam A Bollman, and Michael S Murillo. "Mathematical modeling of disinformation and effectiveness of mitigation policies". In: *Scientific Reports* 13.1 (2023), p. 18735.

[9] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. "Power-Law Distributions in Empirical Data". In: *SIAM Review* 51.4 (2009), pp. 661–703. DOI: 10.1137/070710111.

[10] Michele Coscia. *The Atlas for the Aspiring Network Scientist.* 2021. DOI: 10.48550/arXiv.2101.00863.

[11] Giorgio Fagiolo. "Clustering in complex directed networks". In: *Phys. Rev. E* 76 (2 Aug. 2007), p. 026107. DOI: 10.1103/PhysRevE.76.026107.

[12] Philipe de Freitas Melo, Carolina Coimbra Vieira, Kiran Garimella, Pedro O. S. Vaz de Melo, and Fabrício Benevenuto. "Can WhatsApp Counter Misinformation by Limiting Message Forwarding?" In: *Complex Networks and Their Applications VIII*. Ed. by Hocine Cherifi, Sabrina Gaito, José Fernendo Mendes, Esteban Moro, and Luis Mateus Rocha. Cham: Springer International Publishing, 2020, pp. 372–384.

[13] Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. "Bayesian optimization with inequality constraints." In: *ICML*. Vol. 2014. 2014, pp. 937–945.

[14] K.-I. Goh and A.-L. Barabási. "Burstiness and memory in complex systems". In: *Europhysics Letters* 81.4 (Jan. 2008), p. 48002. DOI: 10.1209/0295-5075/81/48002.

[15] P Grindrod and A Bovet. "Organization and evolution of the uk far-right network on telegram". In: *Applied Network Science* 7 (2022).

[16] S Gupta, RM Anderson, and RM May. "Networks of sexual contacts: implications for the pattern of spread of HIV". In: *AIDS (London, England)* 3.12 (Dec. 1989), pp. 807–817. ISSN: 0269-9370.

[17] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX". In: *Proceedings of the 7th Python in Science Conference*. Ed. by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, 2008, pp. 11–15.

[18] Hang-Hyun Jo, Márton Karsai, János Kertész, Kimmo Kaski, et al. "Circadian pattern and burstiness in human communication activity". In: *New J Phys* 14.1 (2012), p. 013055.

[19] Hang-Hyun Jo, Márton Karsai, János Kertész, and Kimmo Kaski. "Circadian pattern and burstiness in mobile phone communication". In: *New Journal of Physics* 14.1 (Jan. 2012), p. 013055. DOI: 10.1088/1367-2630/14/1/013055.

[20] Márton Karsai, Hang-Hyun Jo, and Kimmo Kaski. *Bursty Human Dynamics*. Springer International Publishing, 2018. ISBN: 9783319685403. DOI: 10.1007/978-3-319-68540-3.

[21] Márton Karsai, Kimmo Kaski, Albert-László Barabási, and János Kertész. "Universal features of correlated bursty behaviour". In: *Scientific reports* 2.1 (2012), p. 397.

[22] Guillaume Laurent, Jari Saramäki, and Márton Karsai. "From calls to communities: a model for time-varying social networks". In: *The European Physical Journal B* 88.11 (Nov. 2015). ISSN: 1434-6036. DOI: 10.1140/epjb/e2015-60481-x.

[23] R Dean Malmgren, Daniel B Stouffer, Adilson E Motter, and Luís AN Amaral. "A Poissonian explanation for heavy tails in e-mail communication". In: *Proceedings of the National Academy of Sciences* 105.47 (2008), pp. 18153–18158.

[24] N. Masuda and R. Lambiotte. *Guide To Temporal Networks, A (Second Edition).* Series On Complexity Science. World Scientific Publishing Company, 2020. ISBN: 9781786349170.

[25] Stanley Milgram. “The small world problem”. In: *Psychology today* 2.1 (1967), pp. 60–67.

[26] M. E. J. Newman. “Mixing patterns in networks”. In: *Phys. Rev. E* 67 (2 Feb. 2003), p. 026126. DOI: `10.1103/PhysRevE.67.026126`.

[27] M. E. J. Newman. “The Structure and Function of Complex Networks”. In: *SIAM Review* 45.2 (2003), pp. 167–256. DOI: `10.1137/S003614450342480`.

[28] Fernando Nogueira. *Bayesian Optimization: Open source constrained global optimization tool for Python.* 2014–.

[29] Alfonso de Paz, Manuel Suárez, Santiago Palmero, Sara Degli-Esposti, and David Arroyo. “Following negationists on Twitter and Telegram: application of NCD to the analysis of multiplatform misinformation dynamics”. In: *International Conference on Ubiquitous Computing and Ambient Intelligence.* Springer. 2022, pp. 1110–1116.

[30] Tiago P. Peixoto. “Bayesian Stochastic Blockmodeling”. In: *Advances in Network Clustering and Blockmodeling.* John Wiley & Sons, Ltd, 2019. Chap. 11, pp. 289–332. ISBN: 9781119483298. DOI: `https://doi.org/10.1002/9781119483298.ch11`.

[31] Tiago P. Peixoto. *Descriptive vs. Inferential Community Detection in Networks: Pitfalls, Myths and Half-Truths.* Elements in the Structure and Dynamics of Complex Networks. Cambridge University Press, 2023.

[32] Tiago P. Peixoto. “Hierarchical Block Structures and High-Resolution Model Selection in Large Networks”. In: *Phys. Rev. X* 4 (1 Mar. 2014), p. 011047. DOI: `10.1103/PhysRevX.4.011047`.

[33] Tiago P. Peixoto. “The graph-tool python library”. In: *figshare* (2014). DOI: `10.6084/m9.figshare.1164194`.

[34] Nicola Perra, Bruno Gonçalves, Romualdo Pastor-Satorras, and Alessandro Vespignani. “Activity driven modeling of time varying networks”. In: *Scientific reports* 2.1 (2012), p. 469.

[35] Shigeru Shinomoto, Keisetsu Shima, and Jun Tanji. “Differences in Spiking Patterns Among Cortical Neurons”. In: *Neural Computation* 15.12 (2003), pp. 2823–2842. DOI: `10.1162/089976603322518759`.

[36] Peter M. Stahl. *Lingua.* 2022. URL: `https://github.com/pemistahl/lingua-py` (visited on 09/09/2024).

[37] Christian L Staudt, Aleksejs Sazonovs, and Henning Meyerhenke. “NetworKit: A tool suite for large-scale complex network analysis”. In: *Network Science* 4.4 (2016), pp. 508–530.

[38] *Telegram APIs.* URL: `https://core.telegram.org/api` (visited on 06/06/2024).

[39] *Telegram evolution.* URL: `https://telegram.org/evolution` (visited on 06/06/2024).

[40] Enrico Ubaldi, Nicola Perra, Márton Karsai, Alessandro Vezzani, et al. "Asymptotic theory of time-varying social networks with heterogeneous activity and tie allocation". In: *Scientific Reports* 6.1 (Oct. 2016). ISSN: 2045-2322. DOI: 10.1038/srep35724.

[41] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world'networks". In: *nature* 393.6684 (1998), pp. 440–442.

Ringraziamenti

Vorrei ringraziare il mio relatore, il professore Claudio Agostinelli, per avermi aiutata e guidata nella stesura della tesi. Ringrazio inoltre Thomas, Riccardo e l'unità Chub di FBK per l'accoglienza, l'aiuto e i consigli che mi hanno dato in questi mesi.

Voglio ringraziare i miei genitori Lucia e Sergio per avermi sostenuta, ascoltata e supportata in questi anni.

Ringrazio i miei meravigliosi fratelli Alice, Andrea e Alessia per avermi sempre offerto ascolto e sostegno. Specialmente Alessia per tutte le risate e i momenti divertenti condivisi assieme.

Ringrazio i miei fantastici nipoti Irene e Gregorio, per tutto l'amore, i sorrisi e le risate che mi hanno donato da quando sono nati.

Ringrazio i miei nonni Bice e Danilo per tutto l'affetto che mi hanno sempre dimostrato.

Ringrazio Laura, compagna in tutto e per tutto, per avermi supportata e sopportata sempre.

Ringrazio la mia amica Rebecca per le nostre lunghe e confortanti chiaccherate.

Ringrazio i miei amici Paco, Sara, Angela, Elizabeth, Daniele e Giovanni.

Ringrazio i miei compagni di università per la collaborazione in anni di lezioni.