SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS DEPUIS 1257

Habilitation à diriger des recherches de Sorbonne Université

Spécialité
Informatique

Présentée par
Lionel Tabourier

From a static to a dynamic analysis of complex networks

Soutenue le 24 septembre 2018

Composition du jury :

M. Étienne Birmelé, Professeur, Université Paris-Descartes ..... Examinateur
Mme Vittoria Colizza, DR INSERM, Sorbonne Université ..... Examinateur
M. Marco Fiore, Chercheur CNR-IEIIT ..... Rapporteur
M. Petter Holme, Professeur Tokyo Institute of Technology ..... Rapporteur
Mme Christine Largeron, Professeur Université Jean Monnet ..... Rapporteur
Mme Clémence Magnien, DR CNRS, Sorbonne Université ..... Examinateur
Mme Céline Robardet, Professeur INSA Lyon ..... Examinateur

Contents

Remerciements

Je remercie d'abord vivement Marco Fiore, Petter Holme et Christine Largeron d'avoir accepté d'être rapporteurs de mon mémoire d'habilitation; ainsi qu'Étienne Birmelé, Vittoria Colizza et Céline Robardet pour leur participation au jury. Je suis très heureux d'avoir pu composer un jury d'une telle qualité, qui reflète la variété des thèmes abordés dans ce mémoire.

Par ailleurs, j'aimerais remercier tout particulièrement Matthieu Latapy et Clémence Magnien : j'ai énormément progressé à leurs côtés au cours de ces dernières années. Par leurs conseils et leur soutien, ils m'ont permis d'avancer scientifiquement tout en me donnant une grande liberté dans mes travaux de recherche et je leur en suis très reconnaissant.

Ce travail doit aussi beaucoup à mes collaborateurs, cités dans ce manuscrit. J'ai eu la chance de travailler avec des gens que j'apprécie aussi bien sur le plan humain que scientifique et j'aimerais les remercier en cette occasion. Ces remerciements s'adressent plus généralement à mes collègues passés et présents, notamment ceux de l'équipe Complex Networks. J'ai appris de chacun d'eux et les années que j'ai passées dans cette équipe ont été très enrichissantes.

Enfin, en dehors du monde la recherche, j'aimerais remercier ceux et celles qui m'ont soutenu pendant la rédaction de ce manuscrit et plus généralement au cours de ces dernières années, tout particulièrement mes parents, Josiane et Pierre, qui, parmi bien d'autres choses, m'ont communiqué leur goût de la connaissance.

Chapter 1

Introduction

Physical contacts between individuals, social interactions, economic transactions, or computers exchanging packets, all these different systems have in common to be a set of interacting objects without being coordinated by a central brain. Therefore, the structure of interactions results from decentralized processes, which are often unknown. Since the 90s, it has been pointed out that graph representations of such systems exhibited common properties, allowing to use transversal methods to describe them and understand their underlying mechanisms. These studies then evolved into a unified field of research, which is called complex networks analysis.

Because of the simplicity of graph representations, as well as the rich body of knowledge accumulated in graph theory and algorithmics, describing interaction data with graphs has led to substantial successes. However, increasing access to online datasets highlighted the need to take into account the intrinsically dynamic aspect of interaction data. This observation appears regularly and in various contexts in my research works. An important part of this manuscript is dedicated to characterizing how such interaction data may be described by tools which account for its dynamic aspects.

This thesis summarizes the most salient aspects of my previous and current research works. Given the format constraints of the manuscript, I will not go into the technical details. Instead, I choose to describe the general ideas that derive from these works and new questions that they bring about.

Scientific motivations and approach

The access to scientific resources is wider than it has ever been, yet the inflation of scientific production also makes it harder to find the most relevant information. It certainly contributes to the difficulty of getting a broad view of a scientific field. My interest for complex networks analysis was raised by the fact that it proposes a language to reveal resemblances between different systems, thus bringing new concepts and tools to existing fields of research. Getting a big picture which relates to many different fields of applications is a very appealing idea in my opinion, and I try to apply the same intentions in my research work.

In retrospect, I realize that bringing together different domains is a costly approach: it demands to understand the vocabulary of a new field, its motivations, questions which are considered important, etc. It also leads to mistakes, among which re-exploring old ideas or following a research path which is known to be misleading. Nevertheless, I believe that it is essential in order to balance hyper-specialization and create bridges between communities. My main scientific interest, which is developed throughout this manuscript, is to contribute to such interdisciplinary connections.

Position in the field

The interaction data that is the focus of this manuscript is either static or dynamic. In the first case, it is a finite collection of pairs (u, v) denoting an interaction[1] between nodes u and v. It allows to represent the data as a graph $G = (V, E)$, where V is a set of nodes and $E = V \times V$ is a set of edges among these nodes. In the dynamic case, data is usually represented as a finite set of triplets (u, v, t), meaning that nodes u and v interact at time t. Several terms have been coined to designate such data and/or the formalisms which allow to manipulate it. Among them, dynamic or temporal networks [29], time-varying graphs [15], link streams [36]. Depending on the dataset considered, the representation may be enriched to account for some specificities, for example links may be directed and/or weighted. In such case, it is explicitly mentioned in the text.

To specify how my works relate to the field of complex networks analysis, I use a two-dimension scheme, with two components covering a large part of existing works:

- The first dimension is the object of research. It is either the structure of the inter-

[1]Some datasets considered in the following describe relations rather than interactions, but for convenience I only use the later term.

action dataset itself, that is to say essentially who interacts with whom and when. Or it is the dynamics of a process occurring on this interaction network. I summarize this dichotomy by the terms *structure* (or topology) and *processes*. But when considering data which is dynamic in itself, the frontier is vague.

- The second dimension concerns what we aim at achieving. Three kinds of activities are considered here: *describing*, *predicting* and *modeling*.

Of course, structuring the field in this way is arbitrary and other choices are possible. In particular, we will see that describing spreading processes may be considered to a certain extent as describing the very structure of interactions. Yet, it allows to represent in a simplified way my contributions to the field in the following double-entry table.

	Describe	Predict	Model
Structure		chapter 3	chapter 4
Processes	chapter 2		

Organization of the manuscript

The approach developed is fundamental, in the sense that my intention is to build transversal methods, which can be applied to improve our understanding of various systems. Throughout this manuscript, I illustrate this methodological work with different applications mainly stemming from the analysis of human behaviors: communication networks, contact networks, etc. The manuscript is organized according to the following outline:

- **Describe the spreading dynamics with cascades** (Chapter 2). Different types of dynamic processes are studied on networks, among which synchronization, routing, consensus search, etc. Here, we restrict ourselves to spreading phenomena. More precisely, we represent propagations by simple, deterministic models (very close to the SI and SIR epidemiological models), that we call spreading cascades. Their primary function is to describe in a simplified way the propagation worst case, but I prefer interpreting them as motifs of description of the interaction structure.

- **Recover and predict interactions** (Chapter 3). We are looking for the dataset features which allow to recover or predict interactions which are likely to happen in the system. I mean by recovering that we find interactions that are missing in the data. It aims at identifying characteristics which are correlated and might be

causally related, thereby we attempt at reconstructing the microdynamics of the system. For this purpose, we use data mining and machine learning tools.

- **Model the interaction structure** (Chapter 4). We investigate the question of modeling, that is to say producing a simplified representation of the system in order to recover characteristics observed in real data. The models proposed rely on a few simple properties, and our work build upon the idea that if we emulate the features observed, we have identified key ingredients of the interaction topology.

- In each of the previous chapters, I present some perspectives related to the work presented. Chapter 5 is specifically dedicated to **applicative prospects**, which also correspond to on-going and future research projects.

On the three topics presented in Chapters 2, 3 and 4, I summarize in the introduction of each section my main contributions and the related references in bold, to make the reading easier.

Chapter 2

Describe the spreading dynamics with cascades

Dynamic processes on networks are used to model a large range of phenomena: opinion formation, synchronization of behaviors, mobility of individuals, etc. (see [9] for a review). My work focuses on spreading phenomena. This should be understood in a broad sense, as we use simple and versatile models of these complex processes. Applications of such models range from the propagation of a disease in a population to belief propagation or rumor spreading.

While there is an important literature dedicated to the development of realistic models for specific situations, especially in the field of epidemic spreading (e.g., [20, 7]), the approach assumed here aims at generality. The main focus is not creating a precise model for a specific disease or environment, but rather uncovering the properties of the interaction datasets on which it is implemented. In that sense, the spreading model can be seen as a probe of the interactions structure and dynamics.

As others did [28, 33, 47, 60, 23], I use simple models on interaction data which has been extracted from real-world situations and describe the spreading characteristics. This approach is upstream of the modeling process. It aims at identifying features of the dataset which are important to characterize the propagation and then guide modeling specific problems. We evaluate the impact of a few features: the burstiness of individual activity, the directionality and ordering of the interactions, and the periods of latency between events.

2.1 Features affecting the size and shape of a spreading cascade

In this section, we investigate the impact of the interaction patterns on the size and speed of a propagation. I first define more precisely what this expression means in the context of this work and then the investigation is led on the example of a phone call dataset. It is a well-known fact that interactions among humans (contacts, emails, etc.) are bursty and heterogeneously distributed [32], and phone calls are no exception to this [47]. It has been disputed in what way these features interlace to affect the propagation speed [33].

On this question, I contributed by evaluating how the size and speed of a spreading is affected by the directionality of the interactions, and the correlation between when and with whom individuals interact. I report here works achieved mostly in collaboration with Fernando Peruani [58, 69] and to a lesser extent with Renaud Lambiotte and Jean-Charles Delvenne [35].

2.1.1 Deterministic spreading cascades

The default propagation model that I use in the following is a variation of the classic epidemiological SIR model (see e.g. [9]). In a few words, a node has one of three statuses: *Susceptible*, *Infected* or *Recovered*. At the beginning of a simulation, all nodes have status S except for a seed of infected nodes. A contact between a S and a I changes the status of the S node to I (with probability 1 in our case). Then, I nodes recover to the R state according to a given rule, here the recovery is deterministic: a node is infected for a period of length τ. Finally, when a node has status R, it remains in this state. I will refer to this family of models as *spreading cascades*[1]. An example of such a cascade is represented on Figure 2.1. A cascade starting from a single node can be described as a tree rooted at this seed node, consequently we can define its size σ (the number of nodes) and its depth δ (the maximum distance from the root to the leaves).

Beyond its simplicity, there are several advantages in this definition choice, which explains why this procedure has also been implemented in other works, such as [47] or [60]. First, it allows to control the duration of infection of a node, which is most useful as we aim at investing the spreading phenomena at different timescales. Another important point is that this model can also be understood as the measurement of a motif in the structure of the interaction stream, as it is deterministic.

[1]Other names can be found in the literature: *branching tree* [75], *causality tree* [58], etc.

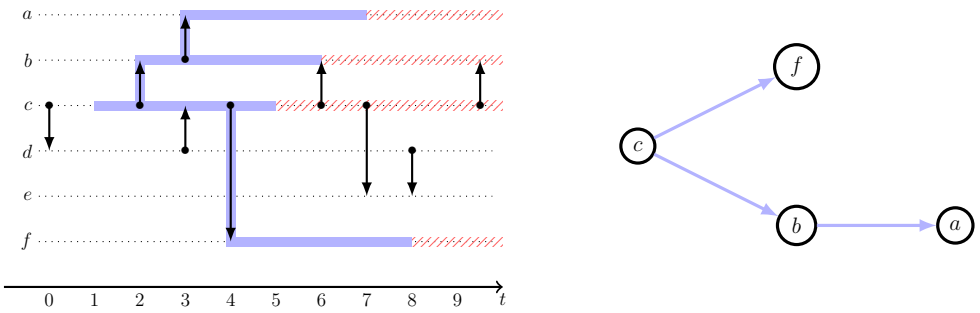


Figure 2.1 – Left: Representation of a spreading cascade starting from node c at time 1 with infection duration $\tau = 4$. Nodes in blue have status I, nodes in hatched red have status R, uncolored nodes are S. Right: Corresponding tree (here, size $\sigma = 4$ and depth $\delta = 2$).

2.1.2 The phone call dataset

Throughout this section, we use a mobile phone call record dataset for illustration purposes. Nodes are anonymized users of a major European mobile phone provider, which metadata (who calls whom and when) has been collected during one month. We isolated the largest connected component of the aggregated network which leads to 1,044,397 users and 13,983,433 phone calls among these users.

In many ways, this dataset exhibits the usual features of a large scale communication network, most notably:

- the degree distribution is heterogeneous: most users have few contacts, a few have many, and the distribution spreads continuously from one extreme to the other;
- the network representation exhibits small-world properties: the clustering coefficient is high (that is to say, much larger than the density of the network), and the average distance between two nodes is a few units;
- individuals have a bursty activity pattern and the collective activity follows the expected regularities: day-night cycles, an average activity roughly similar on week days and diminishing during weekends.

For the sake of brevity, I do not detail further these aspects in this manuscript. More descriptive information can be found in [58].

2.1.3 The need for null models

A recurring question of this thesis is the following: *What is an expected behavior?* A standard answer is comparing to a randomized situation, that is to say a null model. This idea is indeed pervasive in complex networks analysis literature. For example, the abundance of a specific motif in a graph is interpreted in regards to its abundance on a randomized version of this graph [45, 42]. Similarly, a community of nodes is often defined by comparison to some sort of random benchmark, usually a random graph with a similar degree distribution [51], etc.

In general, a randomized model is built from real data by alleviating some constraint. We follow a similar approach on temporal data, using two different models.

- The *time mixing model (tmm)* consists in randomizing timestamps of the dataset. This model breaks the individual bursty activity patterns, but preserve the global periodicities in the dataset (daily, weekly cycles). The topology of interactions (who interacts with whom and how many times) is left unchanged. Given the simplicity of the model, it has been used under many different names in the literature: *time-shuffled* in [33] and [47], *permuted times* in [28], *random dynamic* in [60].
- We proposed the *correlation mixing model (cmm)* [69] to also preserve the individual patterns of activity. It breaks specifically the correlations between who we interact with and when we interact with them. To do so, we shuffle the time labels of the interactions originating from a same caller (or equivalently, we shuffle the destinations of the calls of a same caller), as represented on Figure 2.2. Note that this model can be defined only in the context of directed interactions.

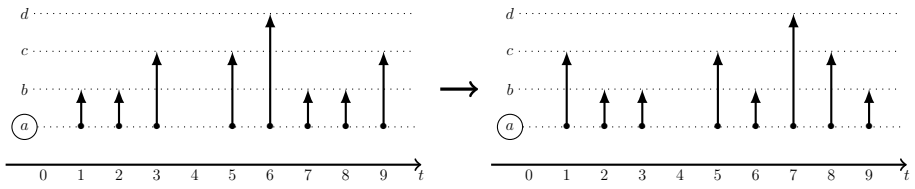


Figure 2.2 – An example of randomization according to the *cmm* on node *a*.

2.1.4 Picturing the spreading at smaller scales

We investigate the shape of the cascades as a function of the parameter τ (duration of infection) and compared the results on real data to the null models mentioned above. We focused on relatively short timescales, that is to say τ does not reach the percolation threshold (this phenomenon will be discussed later). Observing the sizes and depths of the cascades (see Figure 2.3), we can see that real cascades are larger and deeper than those that *tmm* produces. At a short timescale, the spreading is thus faster in the real data structure than in the null model where time labels of links are randomized.

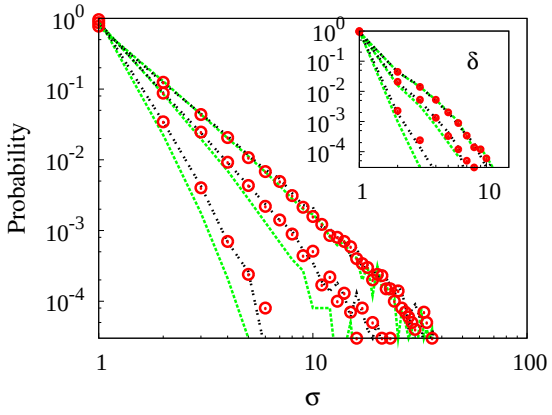


Figure 2.3 – Distribution of the sizes (σ) and depth (δ) in real data (red circles), and according to the *tmm* (green line) and *cmm* (black dotted line). Left group of curves is $\tau = 30$min, middle is $\tau = 3$h and right is $\tau = 12$h.

In [33], Karsai *et al.* observed that the entanglement between the link weights, and the network topology leads to slowing down the spreading. But in that article, the spreading was considered using large scale measurements, for example the time necessary to reach all nodes in the network with a SI propagation. The use of a deterministic SIR model allows to shade and complete these conclusions. We have seen that larger and deeper cascades tend to be over-represented on short scales in real data compared to models where interaction times are randomly shuffled. In other words, at short timescales the spreadings are faster in real data. Once the close neighborhood of a node is reached, we observe that the spreading is "trapped" in this dense area, as commented in the literature [54]. Our conclusion on this point is thus in agreement with what Miritello *et al.*'s established [47]: *"while bursts hinder propagation at large scales, conversations favor local rapid cascades"*.

Notice that these studies were achieved on phone call datasets with SIR-like models. It seems consistent with previous results obtained on the reachability paths of email networks [28]. Other works indicate that other datasets or variants of the model may lead to qualitatively different behaviors. For example, [60] showed on sexual contact data that spreading cascades are faster with real data than with the time mixing model. As underlined in [43], this observation points to the question of the precise features which cause a speed-up or a slow-down of the propagation. In this example, an explanation suggested in [60] is that a user has a longer average inactivity period in real data than in the model. Another possibility is that there are few repeated interactions in this dataset, in strong contrast with phone call data.

Coming back to our study on phone call data, the *cmm* cascades are not significantly different from the real data cascades according to the size and depth measurements. Breaking the correlations between when we call and who we call has effects which are more difficult to detect. To do so, we measure the abundance of temporal motifs: cascades, but also *connected temporal subgraphs* in the sense of [34]. Precisely, we focus on reciprocal interactions (u calls v, then v calls u back in less than τ) and directed cycles (u calls v, then v calls w in less than τ, then w calls u in less than τ). It reveals that some patterns are underestimated by *cmm*, in particular reciprocal interactions and cycles, thus others are overestimated to balance this effect. After a short period – from a few minutes to a few hours, depending on the pattern considered – the effect is not measurable any longer. This shows that the trapping effect aforementioned is not only the consequence of the bursty individual dynamic, but also comes from the correlation between who is involved in the events and when they occur. In phone call datasets, it remains marginal, probably because correlated events are relatively rare, but the approach used could be employed on other directed dynamic networks.

These observations are also consistent with a related work [35]. We showed on toy examples that spreading processes characteristics are affected not only by the fact that the activity on the network is bursty, but also by the ordering of the events. More precisely, this work presents a graph where links are activated according to power-law tailed distributions of inter-event times. It is shown that a spreading process can go from below to above the percolation threshold by tuning the distribution adequately, which corresponds to changing the relative order of link activations. In summary, the heavy tail of the inter-event times distribution is known to have significant effects on dynamic processes [71, 62], but the fact that a certain event takes place before or after some other should not be overlooked, especially when evaluating the importance of a node or a link in the spreading process. This matter will be discussed further in Sec. 2.2.

2.1.5 The importance of directionality

In parallel of the investigations described in the previous section, we explored in what way the directionality of interactions affects the structure of the spreading cascades [58]. Precisely, we investigated analytically and numerically the impact of the link directions on the percolation threshold of the SIR model underlying the cascades. The in- and out-degrees of a node in a mobile phone network are correlated, but the level of correlations is not extremely high (the Pearson coefficient is 0.58 in the phone call dataset). It is thus not obvious to know how this property affects the cascades.

Analytically, we describe the cascade as a branching process. Its properties widely depend on $p(k_i, k_o, \tau)$, which denotes the probability for a node to have a k_i in-degree and a k_o out-degree over a period τ. We make simplifying assumptions in order to derive the characteristics of the process: the in-degree of a node is uncorrelated to the out-degree of its neighbors, the activity of an edge (i.e., the number of times it is activated) is independent from the activity of other edges, and the activity of an edge is independent from the degree of its nodes. Defining the percolation threshold τ_c by the recovery time at which the size of the infection diverges, we are led to the following expression[2]

$$\tau_c = \frac{\langle k_o \rangle_\infty}{\langle \rho \rangle \langle k_i k_o \rangle_\infty} \tag{2.1}$$

where $\langle k_{i/o} \rangle_\infty$ designates the average in- or out-degree of the nodes when aggregated on the whole duration of the dataset, and $\langle \rho \rangle$ is the average activation rate of a link.

We compare this situation to two extreme cases: 1) the in- and out-degree are fully correlated, meaning that for any node $k_i = k_o$, and 2) the in- and out-degree are uncorrelated. The results obtained and the corresponding numerical simulations on real data are reported on Figure 2.4. We first observe that the correlated τ_c threshold is relatively consistent with the simulation results, which suggests that the simplifying assumptions are sufficiently realistic.

The relative order of the thresholds computed was expected: intuitively, the higher the correlation between in- and out-degree, the larger the chance that a *"super-receiver"* (a node with large in-degree) is also a *"super-spreader"* (a node with large out-degree). Consequently, a cascade may easily reach a node that is able to spread fast to a large number of nodes. We also see that taking into account the directionality of the interactions has a significant impact on the characteristics of the cascades, that is visible through the

[2]The details of the computations can be found in [58].

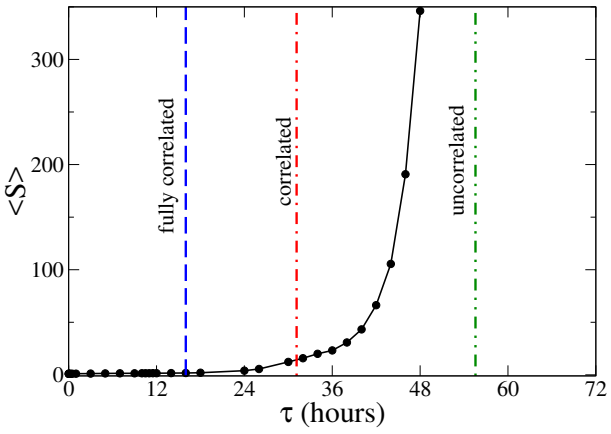


Figure 2.4 – Analytical expressions of the percolation thresholds τ_c (vertical bars) and average size of the cascades in numerical experiments as a function of τ.

threshold computed, as it varies from 14h (fully-correlated) to 32h (correlated) to 55h (uncorrelated).

2.2 Evaluating epidemic risk and control strategies

The previous section mainly focused on fundamental ideas: how features of the dataset impact the propagation processes. In this section, we address similar ideas from the perspective of a specific application. Observing that temporal information has much influence on the shape of a cascade, we wonder how this could be taken into account in epidemic control strategies. In the field of complex networks analysis, the efficiency of control strategies may be tackled as a two faceted problem: one is how do we evaluate that the system is exposed to epidemic risk, and another is how do we define the control strategy. To address these questions, we were granted access to the French cattle trade movement database.

The contribution of this work is to show that both of these aspects can be improved by including temporal information about the patterns of interaction, in order to give a better evaluation of control strategies. It has been achieved in the context of Aurore Payen's PhD, co-supervised with Matthieu Latapy [57].

2.2.1 The BDNI dataset

The BDNI (*Base de Données Nationale d'Identification*) is the implementation in France of the EU decision to record cattle exchanges after the 1996 bovine spongiform encephalopathy crisis. Through the ministry of agriculture, we have access to all cattle trade movements in France from 2005 to 2015. It translates into about 148 million cattle transfers among 300,000 holdings (farms, markets, centers, slaughterhouses, etc.)

I do not describe the dataset in this manuscript, but information about its characteristics can be found in [57]. It is qualitatively similar to the description of years 2005-2009 given in [19] and to a more limited extent to other animal trade movement datasets which have been described in the literature [5, 37].

2.2.2 Evaluating the epidemic impact with cascades

The sensitivity of a system to an epidemic has often been estimated in terms of the number of nodes which can be potentially reached by the infection [5, 19]. According to this view, the measure of the connected components of an undirected network has been proposed as an estimator of the infection potential from an early stage of graph-based analysis [49]. In a directed network, one measures the number of nodes which are reachable from a given seed, that is to say nodes located downstream. From a theoretical point of view, these approaches are interesting, especially as they allow to use the mathematics of percolation processes. But they also describe the impact of an epidemic without considering the temporal ordering of events, which is critical.

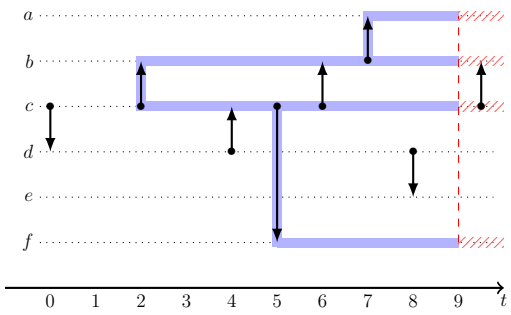


Figure 2.5 – Representation of a spreading cascade starting from node c at time 2, with overall duration $\Delta = 7$.

We are looking for a relevant equivalent when temporal data is available. Computing nodes which are reachable from node u in a directed network is equivalent to running a breadth first search algorithm (BFS) from u, which is in turn equivalent to running a deterministic SI model. So, when looking for an equivalent of a BFS in dynamic networks, we are led to consider deterministic spreading models. Precisely, the model investigated is a deterministic SI model with the additional characteristic that the infection is interrupted after a duration Δ, making this model close to a deterministic SIR model. However, it differs from the one described in section 2.1 as here a node has status I from the moment when it is infected to the end of the cascade. A schematic representation of the process is shown in Figure 2.5. Interestingly, the size of such cascades have been introduced a few years ago under the name of *out-going infection chains* by epidemiologists [53], but formulated in quite a different fashion.

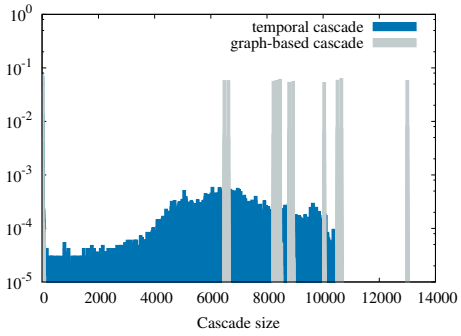


Figure 2.6 – Compared distributions of sizes for static 4 weeks graph snapshots (light gray) compared to 4 weeks cascades in a dynamic network (blue).

Considering the temporal ordering of events has a dramatic effect on the evaluation of a potential outbreak with the size of these cascades. We report in Figure 2.6 the distribution of the cascade sizes compared to the sizes of BFS in static snapshots of a similar duration. The distributions differ qualitatively and quantitatively: first, the maximum size of cascades in dynamic network are by construction smaller than the size of the BFS on the corresponding snapshot (by a factor 1.3 in the case of 4 weeks durations). Also, the sizes of the BFS are distributed according to several modes, which can be expected from the bow-tie structure of real-world directed networks [13]. Indeed, the size of a BFS in a directed graph is given by the number of nodes located downstream to the seed node, so for instance, all BFS starting from nodes located in a strongly connected component reach exactly the same nodes. By contrast, cascade sizes in a dynamic network are continuously distributed. It has been brought to our attention that recent works on

pig trade movements in Germany reveal very similar characteristics [37], and we expect that many other animal trade movements datasets exhibit similar features.

2.2.3 Accounting for speed in evaluation

The size of these cascades is a more reliable evaluation of an epidemic risk than connected components are. However, it would not differentiate a scenario where all reachable nodes are immediately infected from a scenario where the infection remains local during a long period before suddenly breaking out, as represented in Figure 2.7. As a consequence, an adequate evaluation of the risk should also account for the speed of the epidemic. It is all the more important as we have seen in the previous section that the cascade speed is substantially affected by the order of the interactions. Our choice is to consider the *area under the infection curve* (AUIC), which is defined as $\mathcal{A} = \sum_{t=t_0}^{t=t_0+\Delta} I(t)$, where $I(t)$ is the number of infected nodes at time t.

We simulate SI cascades on the BDNI according to the model previously described (with $\Delta = 4$ weeks, 3 months or 1 year). Among the scenarios described in Figure 2.7, we thought that scenario A was the most probable: nodes are quickly infected and when all reachable nodes are reached, we would see a saturation effect. But this not what we observe. A typical cascade exhibits two modes: first the cascade grows very slowly, and it either remains small or it reaches a tipping point and starts growing at a linear rate, but no clear saturation effect is detectable at those timescales. Note that when Δ is several years long, the saturation effect starts to be visible.

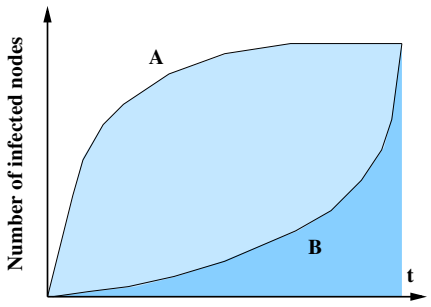


Figure 2.7 – Two possible propagation scenarios (A and B), which lead to the same total number of infected nodes, but to different areas under the infection curves.

In Figure 2.8, we plot the area under the infection curve as a function of the size of the cascade. This scatter plot exhibits two specificities: curves are organized in a

sheaf of parabolas and they are inscribed in a wing-like envelope. We also represent the corresponding scatter plot for a 2-phase model based on the description above. With t_0 the starting time of the cascade, the model is:

- From t_0 to t_w: waiting phase, the number of infected nodes is constant ($I(t) = I(t_0)$), t_w is called the waiting time.
- From t_w to $t_0 + \Delta$: linear growth phase, $I(t) = gt + I(t_0)$, g is the growth rate.

For a given cascade of size $I_f = I(t_0 + \Delta)$, we fit the model and draw the corresponding scatter plot. We observe a similar envelope as the one of the real data. Moreover, for a given growth rate, all points fall indeed on a same parabola. These observations confirm that this basic two-phase description is a good approximation of the shape of a spreading cascade in this dataset.

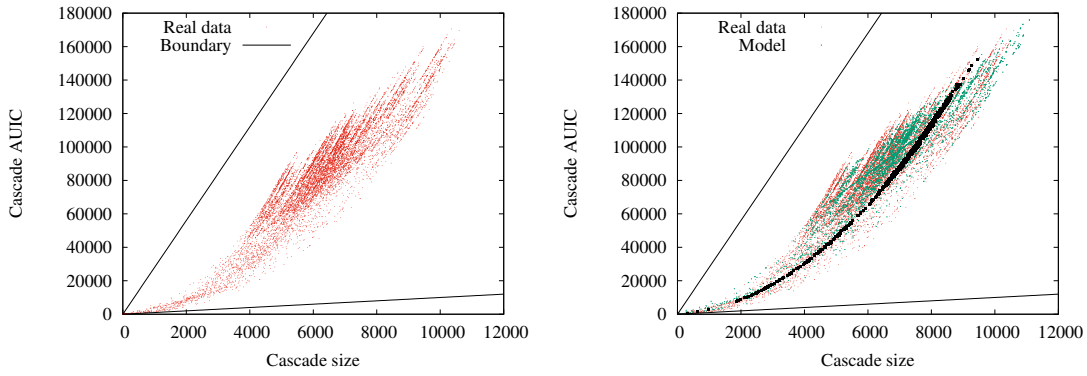


Figure 2.8 – Comparative scatterplots of the cascade sizes and AUIC in the real data and the 2-phase model. The black solid lines represent the theoretical boundaries outside which no cascade can exist, black dots correspond to model cascades with a similar linear growth rate (300 ± 5 infected holdings per day).

For a Δ of 4 weeks, 3 months or 1 year, the two-modes model fits convincingly our measurements. The spreadings do not exhibit an initial superlinear growth, which is often witnessed on other data or on synthetic spreading experiments [10, 60]. Moreover, the timescales at which we observe the BDNI dataset do not allow to observe the saturation effect. Two hypotheses can be made in relation to this observation: either new nodes enter the system, constantly feeding the cascade, or nodes which enter the cascades have been in the system for a long period, but the dynamics is slow in the sense that we are

observing the transient to an eventual stationary state. Additional measurements – not reported here – suggest that the second assumption is more probable in the case of the BDNI.

2.2.4 Monitoring control strategies

Monitoring control strategies often consists in targeting specific nodes (sometimes links) of the system and evaluate how the suppression of this node (link) affects the risk evaluation. A standard targeting method (especially popularized by [56]) consists in ranking nodes according to a centrality measurement, and then suppress them by decreasing centrality. As we consider spreading cascades in a dynamic network, we can refine this method 1) by using the improved risk evaluators previously described (size of cascades, AUIC) and 2) by targeting nodes or links according to temporal features.

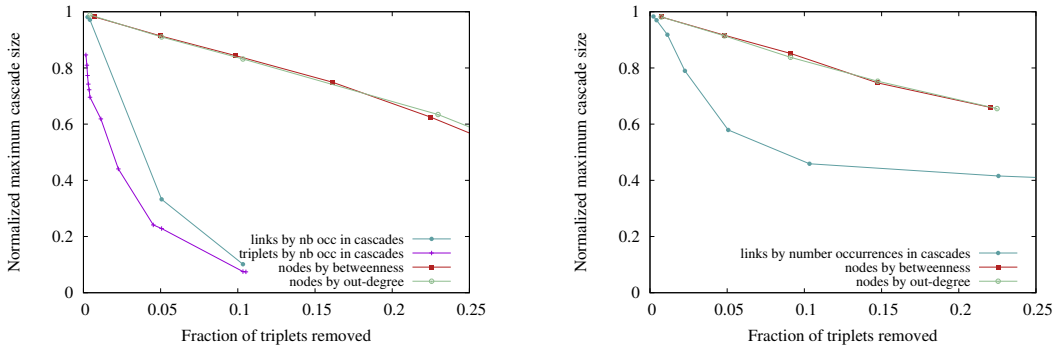


Figure 2.9 – Effect on the maximum cascade sizes of different control strategies. The x-axis corresponds to the fraction of triplets (u, v, t) removed. Left: *a posteriori* analysis context, we use the data of year 2015 both to rank the triplets and to measure the effects on cascade sizes. Right: in a predictive context, we use the data of year 2014 to rank triplets and measure the effects on cascade sizes in 2015.

Another key aspect is that we propose to measure the cost of a targeting strategy in terms of triplets (u, v, t) removed. Indeed, as observed in [43], a triplet is the atom of structure of a dynamic network, which makes it a natural unit in this context. As an illustration, Figure 2.9 shows that an efficient control strategy consists in targeting links which appear in a lot of spreading cascades. It probably stems from the fact that there is an important overlap between large cascades in the dataset. For example, some interaction triplets (t, u, v) belong to 87% of the cascades occurring at t. Consequently,

targeting links upstream of these overlapping regions significantly reduces the cascade sizes.

2.3 Prospects

The questions addressed in this chapter are general to dynamic interaction data. In particular, we have seen on the BDNI case that the spreading process is fundamentally dynamic, in the sense that it does not reach its stationary state. Our current assumption is that it is a consequence of the observation timescale.

This observation may be compared to experiments which have been achieved on the Internet topology at the IP level. Indeed, it has been observed that exploring the structure by radar-like probings lead to the perpetual discovery of new links from an IP address to another [41]. By contrast with our measurements, it seems that in this case new links are always appearing to feed the probe. The system reconfiguration timescale seems to be much shorter than the time necessary to explore a significant part of it, and in that sense, it is not a transient to a stationary state.

Of course, the measurement procedure raised here significantly differs from the experiments that I reported in this chapter. However, it suggests to investigate dynamic interaction datasets in a transversal manner. The next step that I consider is to identify characteristics of the dynamic network (such as node appearance rate, inter-event times distribution, etc.) which would give an indication of the type of dynamic regime that takes place in the system.

Chapter 3

Recover and predict interactions

In Chapter 2, the description of spreading cascades gave us a glimpse of the complex interweaving between the interaction topology and the bursty activity patterns. To have a better understanding of these interacting systems, a long-term objective would be to uncover the mechanisms governing their dynamics. More modestly, identifying which interaction might trigger another, or pointing out correlations between structural and temporal properties of the dynamic network gives insights on candidate mechanisms.

This is the topic of this chapter. Precisely, starting from the problem of recovering and predicting links in a large graph, I will progress towards the question of predicting links in a dynamic framework.

3.1 Using temporal information for predicting links

A standard formulation of the link prediction problem can be found in a famous article by Liben-Nowell and Kleinberg [38]. In their words: *"Given a snapshot of a network at time t, we seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time t' "*. The problem of link recovery is very similar in theory. The main difference stems from the fact that the links to be discovered are missing links at time t (for example because the collection process is not comprehensive). Link recovery does not aim at identifying which patterns cause the occurrence of links in the future, but rather at determining elements of the network structure which are correlated. Throughout this section, these two problems are dealt with in a similar way and when both problems are concerned, I favor the term link prediction for generality

purposes.

Link prediction can be formulated either as an unsupervised or a supervised machine learning problem. As defended in [39], the supervised framework allows for more precise and controlled predictions. In this context, the problem is clearly formalized as a binary classification task, for which a plethora of methods are available. But it also raises specific questions, among which we concentrate particularly on how to scale to large graphs.

My main contribution on this question is the realization of a supervised link prediction method (called RankMerging), which scales to large graphs and allows for a certain degree of comprehension of the mixture of the features. This work is the product of a collaboration with Renaud Lambiotte, Daniel Bernardes and Anne-Sophie Libert. Its results have been described and applied in various contexts in a series of papers [66, 64, 67] and the corresponding code is available at the address: **http://lioneltabourier.fr/program.html.**

3.1.1 The specificities of link prediction in large networks

Link prediction is usually seen as a binary classification task, and can benefit from the important literature on the question. Nevertheless, it also has several specificities, some of which are pointed out in this section.

Class imbalance. Link prediction seen as a classification task suffers from class imbalance. Precisely, any pair of nodes belongs to one of two classes: connected or not, the number of pairs of nodes is $\frac{N\cdot(N-1)}{2}$, with N the number of nodes. Real world networks are often sparse, with a number of links of the order of N. Consequently, the class of unconnected pairs is typically N times larger than the other, which makes the classes imbalanced, especially for large graphs.

Tuning the number of predictions. While it is not necessary in all experimental contexts, a user is often willing to adjust the number of predictions. Indeed, some applications demand to make few but high-precision predictions, while others may have lower precisions but demand a recall value as high as possible. There are several ways to tune the number of predictions in a classification task. We favor a ranking solution to the problem. The idea is that pairs of nodes are ranked according to their supposed probability of being connected, and we use the top T elements of this ranking to predict

T links. Because having a supervised framework seems a crucial point to us in order to make accurate and controlled predictions, we are brought to transform the initial problem into a supervised learning-to-rank task, with large ranking sizes.

Evaluation. Another question is how to evaluate the quality of a link prediction in a large graph? Plenty of evaluators exist in the literature of classification, and the AUC (area under the ROC curve) is often favored to summarize the performance of a prediction method. However, we think (like others, e.g. [74]) that precision and recall are more appropriate to assess the quality of link predictions in large graphs. Indeed, it appears in experimental contexts that the ROC curve is harder to interpret than the precision-recall curve. One may grasp the underlying idea by realizing that the false positive rate is usually very low because of the large number of true negative predictions, making the "interesting part" of the ROC curve limited.

3.1.2 A supervised ranking method to predict links

The general idea of a supervised pair ranking method is the following: we consider various features describing the interaction data, each of them chosen to be correlated to the probability for a pair of nodes to be connected. We build an unsupervised ranking from each of these features – the input rankings – and then combine them in a supervised framework in order to get an improved ranking, which is used for the actual prediction.

Unfortunately, there are few supervised learning-to-rank methods available which scale to large graphs. Indeed, considering networks with $N = 10^6$ nodes, there are roughly 10^{12} candidate pairs to rank, which is hardly manageable in practice. Even when restricting the prediction to relevant pairs (we shall see how to do that), we typically handle rankings with 10^5 to 10^7 items. Yet, most research has focused on improving the prediction accuracy on a relatively small number of items, rather than making the prediction scalable [16]. The main purpose of this work is to design an efficient learning-to-rank method with a linear complexity in the size of the input rankings, thus suited for link prediction in large graphs. Due to length constraints, I only report here the general scheme of *RankMerging*, which is described briefly in [66] and in more details in [64].

Training dataset, test dataset. We first focus on the case of link recovery, which is simpler to describe. To define a supervised method, the user needs labeled data, that is the training dataset. Here, it is a graph (V_1, E_1), where V_1 is the set of vertices, E_1 is the

set of links. A subset $E_1' \subset E_1$ is the set of links to be guessed. These links are used to learn the parameters of the method. The test dataset is used for the actual prediction; it is also a graph (V_2, E_2). A subset $E_2' \subset E_2$ is the actual subset of missing links that the method aims at recovering. Of course for the learning to be efficient, the underlying assumption is that the existence of the links in E_2' can be explained by the structure of $(V_2, E_2 \setminus E_2')$ for the same reasons as the links in E_1' can be explained by the structure of $(V_1, E_1 \setminus E_1')$.

In a link prediction context, the training and test datasets are defined similarly, except for the fact that the links to be guessed correspond to links which appear in the future.

***Rankmerging* training phase.** Suppose that we have α rankings of all the pairs of distinct nodes in $V_1 \times V_1 \setminus (E_1 \setminus E_1')$, each of which have been obtained using features of $(V_1, E_1 \setminus E_1')$. Typical examples of features include indices measuring the similarity of the structural environment of two nodes, such as the number of common neighbors, but also indices measuring the node attributes similarity, etc. (see [1, 40] for examples). The principle of the training phase is to combine these rankings to obtain a new, better one.

To do so, we define a list of pairs for each input ranking. All lists have the same length, which is tuned to obtain the best possible training. Each list gathers the top-ranked available pairs of its ranking and at each step, we count the pairs within each list which are connected, that is to say in E_1'. Then, we select the ranking corresponding to the highest count: we consider that this ranking is the *best* available at this step, and its top-ranked pair is added to the output ranking. Simultaneously, we register the ranking which has been selected at this step. The top-ranked pair selected is removed from the lists that it featured in and it is made unavailable for future selection. The next top-ranked pairs available in each ranking are added to the lists so that their length remains constant. Then, we iterate the process.

At the end of the learning phase, we have an output ranking, which quality can be measured by the number of pairs in E_1' that features in the top T items. Moreover, we have monitored the rankings selected at each step of the process, and this information is used during the test phase. Note that we went through each ranking only once during this process, which makes the complexity of the training phase linear in the ranking sizes.

***Rankmerging* test phase.** First, we generate the unsupervised rankings of all the pairs of distinct nodes in $V_2 \times V_2 \setminus (E_2 \setminus E_2')$ which correspond to the set of classifiers on $(V_2, E_2 \setminus E_2')$. Then, we combine them by using at each step the ranking which had been

selected during the training phase. For example, if at step 1, the ranking corresponding to the number of common neighbors had been selected during the training phase, then the first element of the test phase output ranking is drawn from the top of the number of common neighbors ranking computed on $(V_2, E_2 \setminus E_2')$. Note that in case the test rankings do not have the same size as the learning rankings, we apply a scaling factor when selecting pairs during the test phase. Again, the process complexity is linear in the size of the ranking, making the method scalable to large graphs.

Experimental results and method characteristics. We apply *RankMerging* to link recovery problems on social networks. For this purpose, we used data from a Slovakian OSN: Pokec (8,320,600 links and 1,632,804 nodes)[1] and a scientific collaboration network: DBLP (10,724,828 links and 1,314,050 nodes)[2]. The data is preprocessed in order to simulate a situation where a percentage of links are missing in the network. We limit the unsupervised classifiers to structural index which measure the similarity of the structural environment of two nodes (see [64] for the exact list). To limit the sizes of the rankings (initially larger than 10^8 pairs), we focus on 10^6 pairs which belong to the intersection of the tops of all unsupervised rankings.

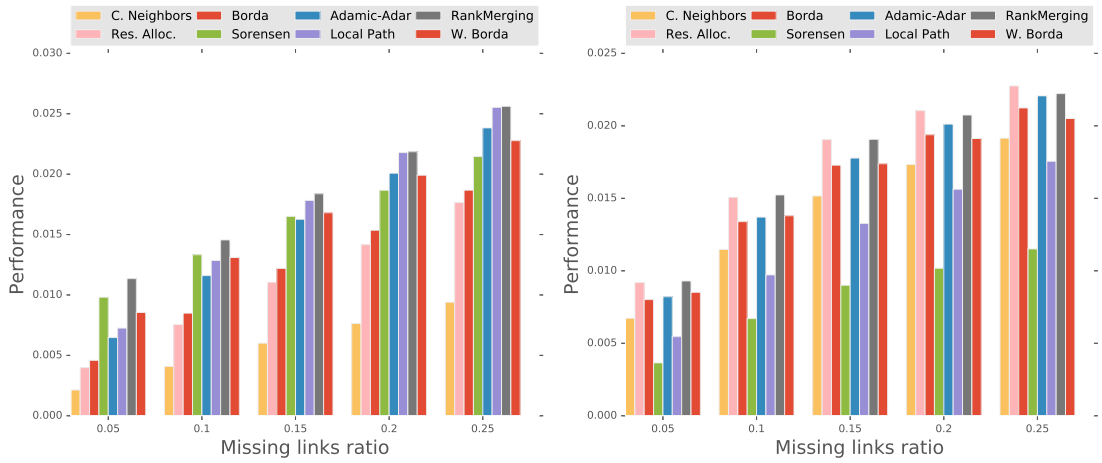


Figure 3.1 – Area under the precision-recall curve for different missing links ratios. Left: DBLP dataset, right: Pokec dataset.

The performances of our method are evaluated in Figure 3.1. They are compared to

[1] available at http://snap.stanford.edu/data/soc-pokec.html
[2] available at http://konect.uni-koblenz.de/networks/dblp_coauthor

the unsupervised rankings which were used as inputs and to the *supervised weighted Borda* method, which is the only supervised learning-to-rank method that we found which scales up to the problem (see [59] for a description of the method). The results are reported for various percentages of missing links, and evaluated using the area under the precision-recall curve.

Most of the time *RankMerging* yields better results than input rankings, and than the *supervised weighted Borda* method. When it does not (for example on the 0.25 missing link ratio experiment on Pokec dataset), the reason lies into overfitting: the best ranking during the training process is not as efficient during the testing phase. In terms of computation times, the experiments reported here are about 10^3 seconds long (training + test, on a standard processor). Such computation times can be neglected when compared to the time needed to generate the input unsupervised rankings (which can reach up to 10^5 seconds with the features that we used).

Complementary investigations – not reported here – allow to describe the usual behavior of the method: it tries to stick to the best performing input ranking during the process, and switch ranking when it considers that there is a better one available. Note also that while it is efficient when predicting a large number of links, the method is not appropriate for a relatively small number of predicted links: as each selection is based on the properties of a list of elements, there may be misestimations of the quality of top-ranked items.

3.1.3 Case study: temporal information for ego-centered networks

In this section, we investigate an implementation of the method described above in the practical context of ego-centered networks to illustrate two aspects of *RankMerging*: the accuracy improvement allowed by a supervised formulation and its level of interpretability. This application is also an opportunity to evaluate the information brought by temporal features to a link recovery problem.

Problem formulation. The problem is the following: we suppose that we have access to the interactions of a node (called *ego-node*) in the network, but we do not have information about the interactions among its neighbors. Such a set-up is frequent for example when data is collected from interviews and it is not possible to collect information from the neighbors of an interviewee. So, with minimal structural information, our goal is to

infer the existence of a link between the neighbors of the ego-node. Thus we rely on temporal information. Precisely, the starting assumption is that much information can be derived from the timing of interactions between an ego-node and its neighbors. To test this hypothesis, we compare the results of predictions using only structural information to predictions which are based on both structural and temporal information.

Data and benchmark. The data that we use for illustration is derived from the phone call dataset described in Sec. 2.1.2. In addition to who calls whom and when, we also use the information of the duration of the phone calls and the text-message communications in this section. The ego-networks are considered independently, and the information available is the number of interactions between an ego-node and each of its neighbors.

As seen in Sec. 3.1.2, pairs of nodes (in this case the pairs of neighbors of ego-nodes) are ranked according to a score, and the top-ranked items are considered as most likely to be connected. Among several purely structural benchmark scores that we have compared, the most efficient one happens to be the ranking provided by the score

$$s(i,j) = \frac{w(e,i) \cdot w(e,j)}{\sum_i w(e,i)} \tag{3.1}$$

where e is the ego-node and $w(x,y)$ designates the weight (i.e., the number of interactions) between x and y.

Temporal features. We designed temporal features that we considered to be relevant in order to detect links among neighbors of an ego-node. I briefly summarize here the intuition underlying some of these features – see [67] for the precise definitions of these scores.

- *Profile-based scores* are built on the idea that one tends to call or text others at particular moments of the day or week, e.g., coworkers during working hours, family and friends on week-ends, etc.
- *Delay-based scores* use the fact that if A calls B just after A called C, there is a higher probability that B and C know each other. We defined several of these scores depending on the timescale considered (1h, 3h, 1 day or 1 week delay).
- The *duration-based score* is based on the simple idea that if i and j are strongly connected to the ego-node, they have a larger probability of being connected to each other, the strength is measured by the aggregated duration of the phone calls.

- The *regularity-based score* is based on a similar intuition as the previous one, except that strength is evaluated by the regularity of the calls (regularity is evaluated using the Fano factor on the phone calls).

Protocol and results. We draw benefit from the fact that ego-networks can be grouped based on their degree k, which substantially improves the learning and thereby the prediction. We define degree classes $k = 2$, $k = 3$, etc. and build a learning and a test set for each of these classes. We suppose that the connections between neighbors are known for 80% of ego-nodes and the prediction is made over the remaining 20% ego-nodes.

The results are presented in the table below, where we observe that the improvements to the benchmark s (see eq. 3.1) of the area under the precision-recall curves are significant, ranging from 15.5 to 51.6 %. It means that temporal patterns do indeed contain information that can be used for link prediction. This is especially true for high degree classes, which comes from the fact that the information is richer on high degree ego-nodes.

ego-node class	Pr-Rc improvement
$k = 2$	+ 15.5%
$k = 3$	+ 18.8%
$k = 4$	+ 19.3%
$k = 5$	+ 21.4%
$k = 6$	+ 22.3%
$k = 7$	+ 22.5%
$k = 8$	+ 25.5%
$k = 9$	+ 25.5%
$k = 10$	+ 28.1%
$k = 11$	+ 30.9%
$k = 12$	+ 26.4%
$k = 13$	+ 33.1%
$k = 14$	+ 36.2%
$k \geq 15$	+ 51.6%

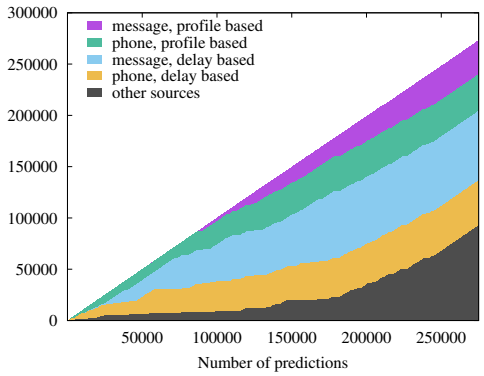


Figure 3.2 – Results of the link prediction via *RankMerging* in the ego-node protocol. Left: area under the precision-recall curve improvement to the benchmark s. Right: contributions of each ranking to the merged ranking at each step of the process, class $k = 8$.

Key information is brought by Figure 3.2 (right): it reveals which ranking is used at which step of the learning process, on the example of class $k = 8$, which is typical (most classes exhibit a similar pattern). *RankMerging* tends to select pairs of nodes using delay-based scores at the beginning of the process. More detailed results (not reported here) show that delay-based scores related to phone calls at short timescales feature in the top of the output rankings: for example, the 1h and 3h delay-based rankings alone are

responsible for about 30% of the links predicted among the top 100,000. Information based on temporal profiles is as well quite largely used. On the other hand, some rankings are nearly not used by the method, revealing that they bring information which is redundant with other scores; that is the case of regularity and duration-based scores on this specific dataset.

3.1.4 Conclusion

This section is a methodological interlude in the global direction of my work. Chronologically, we started from the question *how to draw full benefit from the temporal information when predicting links ?* and were led to devise a method for predicting links using supervised learning-to-rank methods at a large scale. It allowed to bring into light some typical issues of this problem and suggest ways to solve them. It also confirmed the intuition – which is certainly largely shared – that temporal information does bring significant information to the prediction.

3.2 Predicting who interacts with whom and when

We are often tempted, when thinking about link prediction, to represent the situation as a sequence of static pictures of graphs taken at different instants – snapshots of graphs. While it allows to use the powerful formalism of graph theory, it also misses some important points. One of them has been mentioned in the previous section: temporal features of the dataset bring meaningful information to a prediction. Another is the fact that some links have appeared several times in the past, and may appear several times in the future. When considering each link individually, one could think of it as a series of peaks, and collectively as a time series. Past works have brought some light on these aspects: creating hybrid prediction methods which mix time-series analysis and structural link prediction [30, 17], or considering links differently whether they are new or recurring [61]. But instead of tweaking link prediction methods for them to incorporate temporal aspects, the idea defended in this manuscript is that the problem itself should be rethought in another vocabulary.

When temporal information is available, investigating the data as a link stream [36] sheds a different light on how to solve the problem. As mentioned earlier, several formalisms have been developed in order to apprehend interaction data as dynamic networks. ***In this section, I present one way to tackle the problem in the words of the***

link stream formalism, thus emphasizing the duality between the structural and temporal aspects of the problem. The main contribution here is to propose a modular prediction framework, which lays the foundation of link prediction with this formalism. It has been achieved in the context of Thibaud Arnoux's PhD thesis, supervised jointly with Matthieu Latapy [2, 3].

3.2.1 Predicting the activity in a stream

First, let us precise that a link stream is defined here as a triplet (T, V, E) where $T = [\alpha, \omega] \subseteq \mathbb{R}$ is a time interval, V is a set of nodes and $E \subseteq T \times V \otimes V$ is a set of links: $(t, u, v) \in E$ means that u and v interacted at time t. We consider here undirected interactions between pairs of distinct nodes u and v, which we denote by $(u, v) \in V \otimes V$.

The first – most ambitious – formulation of the problem that one could imagine is to predict the stream itself, that is to say predict the exact moment when each interaction occurs. This problem raises many tricky issues. To get an intuition of the origin of the troubles, one may think of it in this way: while predicting links is a difficult task in the sense that class imbalance often causes the rate of good predictions to be low, the task that we define has an additional dimension of prediction to match: time. Consequently, we choose to address a simpler problem, that is predicting the activity of the pairs of nodes in the stream.

Formalization. The *activity* of a pair (u, v) in the stream (T, V, E) is defined as $\mathcal{A}^{uv} = |\{(t, u, v) \in E\}|$, that is to say the number of occurrences of interactions between u and v during the period considered. In practice, we have data during a given period, that is the observation stream $L_o = (T_o, V, E_o)$ and we aim to predict the activity of each pair of nodes $(u, v) \in V \otimes V$ during the prediction period T_p. If $T_o = [A_o, \Omega_o]$ and $T_p = [A_p, \Omega_p]$, then we have $\Omega_o \leq A_p$. For convenience, we choose $\Omega_o = A_p$ in the examples considered. In addition, we define the actual stream $L_a = (T_a, V, E_a)$ which is the stream that actually happens (the ground truth) during the prediction period, so $T_a = T_p$.

One might notice that this task may be formulated with the lexicon of link prediction: we aim at predicting the links that will appear during period T_p and their weights. However, our point here is precisely to explore in what way the intuition is stimulated by formulating this problem with the link stream formalism.

Quality evaluation. Denoting $\mathcal{A}_p^{uv}$ the predicted activity of a pair (u, v) and $\mathcal{A}_a^{uv}$ the actual activity during the prediction period (so the target of the prediction), it is natural to use the difference between these quantities as a quality estimator.

In order to give an interpretation to the quality evaluation, notice that if a link has been predicted while it has not yet occurred, it can be understood as a *false positive* prediction. Similarly, a link which exists in the actual stream while it has not been predicted can be seen as a *false negative* prediction. Events which are predicted and occurred are *true positive* predictions. Consequently, we may denote quantities related to the prediction in the following way:

$$\begin{cases} |tp^{uv}| = \min(\mathcal{A}_a^{uv}, \mathcal{A}_p^{uv}) \\ |fp^{uv}| = \max(\mathcal{A}_p^{uv} - \mathcal{A}_a^{uv}, 0) \\ |fn^{uv}| = \max(\mathcal{A}_a^{uv} - \mathcal{A}_p^{uv}, 0) \end{cases}$$

Considering the prediction on all the pairs of nodes, $|TP| = \sum_{(u,v)\in V\otimes V} |tp^{uv}|$ is the total number of true positive predictions, and we denote similarly $|FN| = \sum_{(u,v)\in V\otimes V} |fn^{uv}|$ and $|FP| = \sum_{(u,v)\in V\otimes V} |fp^{uv}|$. While merely a notation, it also leads to define analogues for other quality evaluators:

- the precision $\mathbf{pr} = \frac{|TP|}{|TP|+|FP|}$ quantifies the fraction of good predictions among the total number of predictions,
- the recall $\mathbf{rc} = \frac{|TP|}{|TP|+|FN|}$ is the fraction of events detected among the total number of events which can be detected.

In the following experiments, we usually employ their harmonic mean, analogous to the F1-score, as the quantity to optimize during the prediction process.

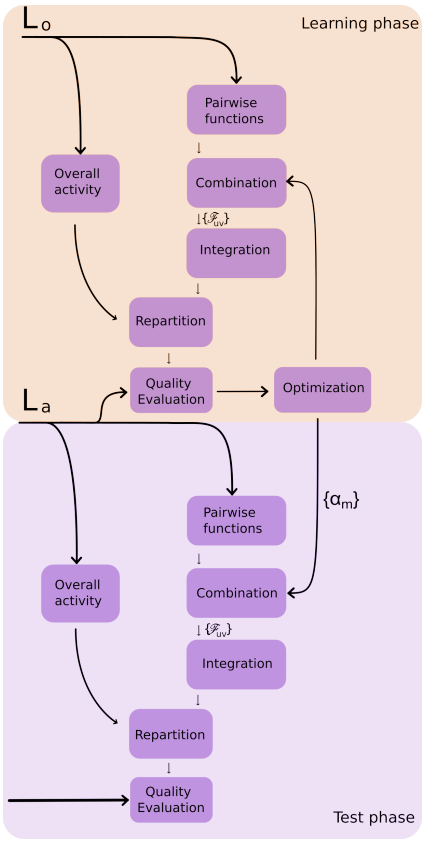
3.2.2 Addressing the problem with pairwise likeliness functions

We propose to address the problem with what we called *pairwise likeliness functions*. These functions are built in order to reflect the propensity for a given pair to interact at a given time. For example, if the connections between two nodes are known to be very regular, then it would be relevant to define the pairwise likeliness function as a periodic function with a period based on the regularity of previous interactions. One benefit of this approach is that features which are commonly used for link prediction or time series prediction tasks can be translated to this representation. For instance, link prediction often calls to structural index, such as the number of common neighbors. Such an index can be expressed as a constant function in this framework.

Finally, each feature k of the observation stream is represented by a pairwise likeliness function, denoted $f_k^{uv}(t)$, which depends on the pair (u, v) considered. Then we combine theses functions in a *combined pairwise likeliness function* $\mathcal{F}^{uv}(t)$. Many combinations can be imagined; here we choose a simple linear combination: $\mathcal{F}^{uv}(t) = \sum_k \alpha_k f_k^{uv}(t)$.

3.2.3 Experimental implementation

Now that we have defined the basic concepts of the prediction framework, we can describe its technical implementation. The workflow which is summarized in the figure below can be seen as a series of interconnected blocks. Indeed, there are different steps in the prediction process, and our choice is to achieve a modular process, so that each module can be improved independently depending on the data. The most important steps of the process are specified below.



Overall activity prediction. The first step of the prediction is to evaluate the overall activity in the stream during the prediction period – that is to say the total number of links which appear – from the activity observed during the observation period.

Feature choice. We define features – that is to say likeliness functions – that reflect different characteristics of the pairs of nodes. Some are built from purely structural information (number of common neighbors, Adamic-Adar index, ...), others from purely temporal information (activity during the last 1000 seconds, ...) and others from hybrid information (structural metrics weighted by the number of interactions). Using a link stream representation also allows to use features which are specific to this formalism, we come back to this point in 3.2.4.

Learning process. The prediction itself consists in distributing the total activity among the different pairs of nodes proportionally to the integral of their combined pairwise likeliness function. The weights α_k of the combination are learned to optimize a quality estimator of the prediction on the observation stream.

Results. Experimental results are obtained on human face-to-face contact datasets. The details of this study can be found in [3]; I only report here a few experiments on a dataset collected during the Infocom 2006 conference in Barcelona[3]. During 3 days, contacts between 98 nodes have been registered using bluetooth devices, and we focus on the trace of the first day of the event.

The experimental implementation reported is a proof of concept, so we make the choice of simplicity when setting the experimental details:

- The overall activity prediction is achieved by extrapolating the activity during the observation period to the prediction period with a linear model.
- For the features choice, we use a variety of structural, temporal and hybrid features.
- Concerning the optimization process, the observation stream is cut in halves and the features defined on the first part are used to predict the activity on its second part. The quality of the prediction is computed using the equivalent of the F1-score defined in Section 3.2.1 and the optimization process is implemented via a standard stochastic gradient descent.

Results are reported in Figure 3.3, where we observe a typical pattern of how weights are distributed among features by the learning process: a mix of features have significant weights, indicating that various sources of information are indeed complementary. We also see that the method tends to favor temporal metrics at the detriment of structural and hybrid metrics.

A closer analysis reveals that 98% of the links predicted are chosen among recurring links, that is to say links that have already occurred in the observation period. In fact, they represent 80% of the links that actually appear. The reason for this is that the prediction of new links is harder in this dataset, hence the learning method tends to give weight to temporal features because they are related to pairs of nodes which have interacted in the past, so recurring links. Actually, this observation was quite expected, as predicting recurring links and predicting new links are often considered as two different prediction tasks. However, we will make suggestions in Sec. 3.2.4 to adapt to this problem, while keeping the framework unchanged.

Throughout the implementation, arbitrary choices have been made: feature selection (i.e., likeliness functions), learning techniques, observation and prediction periods, etc. Some of them are discussed in more details in [2, 3], but the main point of this study is to

[3]Data available at https://crawdad.org/cambridge/haggle/.

Observation Duration	# Links Predicted	# Links Appearing	Precision	Recall	F1-score
1h	8167	8220	0.58	0.59	0.59
2h	16737	14051	0.50	0.60	0.55
3h	22568	20850	0.63	0.70	0.67

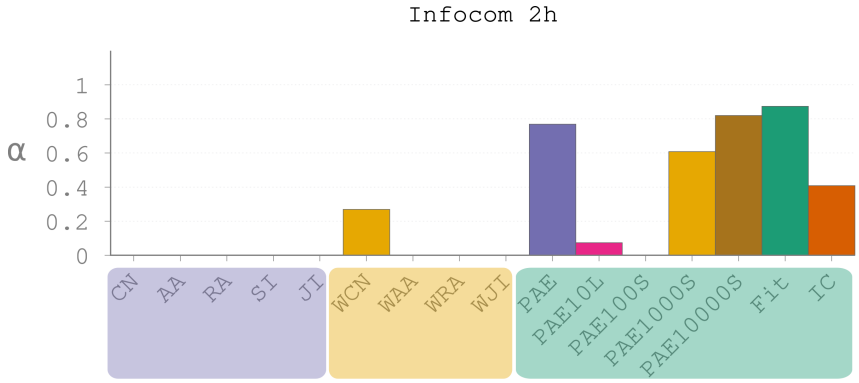


Figure 3.3 – Top: quality evaluation with different experimental settings. Bottom: distribution of weights over the features, leftmost are structural (blue group), middle are hybrid (yellow group), rightmost are temporal (green group).

show the viability of such an approach and its potential for further improvements, which is allowed by its modular structure.

3.2.4 Prospects

Indeed, we believe that there is much to investigate in this line. Among future directions, a few of them are:

- **Adapting benchmark methods.** It is still difficult to assess the full benefit that we can draw from the link stream formulation of the problem, as link prediction methods are not directly comparable to ours. Thus, we should put some effort in adapting existing link prediction methods to this task to measure the improvement brought by a pairwise likeliness function-based formulation of the features.

- **Dividing pairs of nodes into groups.** We have noticed that the method tends to favor the easier prediction of recurring links at the expense of new links. By

dividing pairs of nodes in subgroups, and having an independent learning process for each of these groups, we observe that the weights spread differently over the features depending on the group. For example, new link prediction tends to use structural information, while recurring link prediction favors temporal information. Furthermore, new and recurring links is only one way to divide pairs of nodes, one is tempted to find appropriate subgroups automatically. In that respect, data mining methods such as subgroup discovery [4] may be interesting leads.

- **Feature design.** How should we design the function-based features? Surely, we can draw some inspiration from previous works which attempt at making hybrid prediction models [17]. It is also inviting to turn to features which mix the structural to the temporal aspect of the data representation. For instance, the equivalent of a clique in a link stream has been defined in [72]; we suppose that if a temporal clique has started during the observation period, it may continue during the prediction period. Can we define features that account for this property?

When these goals are met, we should be better armed to address the problem of predicting the stream itself. The long-term purpose of this research line is to question the problem of what should be considered as *normal dynamics*, and dually what is an *unexpected event* in a dynamic network.

Chapter 4

Model the interaction structure

As mentioned previously, when it comes to decide if a phenomenon is expected or not, a usual approach is to define a baseline model, that is to say to generate a sample of random graphs with specific properties. The model is supposed to describe a normal behavior, then one evaluates if what we observe differs significantly from this definition of normality. This question arises when describing the structure of the stream. One generates artificial data according to some assumptions, and if the artificial data have similar characteristics to the real one, then the assumptions may be relevant. In that sense, data modeling has an explanatory dimension, that we investigate in this chapter.

I first tackle the problem of generating artificial graphs in the context of static interaction data and then consider the question of generalizing to dynamic networks.

4.1 Flexible graph generation

Before addressing dynamic data, let us consider graph models of static interaction networks. A well-known illustration of the approach in the context of graphs has been presented in [52], where the authors showed that a simple bipartite configuration model accounts for some properties of a large range of social networks. In other contexts, one would like to have access to more elaborate constraints. For example, Mahadevan *et al.* [42] discussed increasingly constrained network models in order to account for the structure of the Internet at the Autonomous System level. While very interesting in its focus, the generative procedure implemented is flawed: the authors compare the topology of real networks to graph models with predefined 3-nodes correlation distributions,

however there is no known procedure to generate such graphs uniformly, see [63] for a detailed analysis in French. This observation points out the need for devising methods to generate uniformly a sample of graphs, which have structural properties resembling real data.

I present in this section a step toward this direction. Precisely, my main contribution is the realization of a flexible method to generate uniformly graphs satisfying elaborate topological constraints. It is the result of a collaboration with Camille Roth and Jean-Philippe Cointet [68, 65] and it has been used for applications in economics [27].

4.1.1 The uniformly random graph generation issue

Processes to generate random graphs can be classified in two categories: building processes and shuffling processes – sometimes also called respectively bottom-up and top-down null models [23]. Building processes consist in assembling the constituents of the graph according to a given rule. The famous Barabási-Albert or Watts-Strogatz processes and their numerous variants belong to this category. While they give an intuition of the processes at work, the extent of their explanatory power is arguable because they produce graphs that have unrealistic properties. For example, Barabási-Albert model produces a very specific subset of scale-free networks, which are almost tree-like (in the original version of [8]).

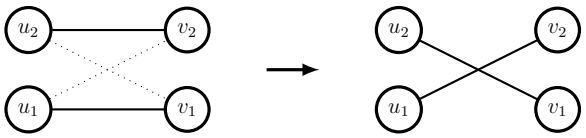
Therefore, we are more interested in generating a random graph which satisfies a precise set of constraints, that we call *target set*. Any graph satisfying the target set of constraints should have the same probability of being produced. If so, the generative process is said to be *uniform*. It is sometimes possible to do so with a building process. For example, the configuration model generates uniformly random graphs with a given degree distribution using a building process [48, 50]. However, when the set of constraints is more elaborate, building methods are often unusable. Consequently, we turn to shuffling processes: the overall idea is to start from any graph satisfying the target set of constraints, then the structure is randomized iteratively until the graph obtained is a random element of the set of interest. We develop a specific design of shuffling method in the following.

4.1.2 Proposed method

Edge switching method. Let us start with a simpler graph generation method, widely used in the literature (e.g. [46, 73]). We suppose that we have at least one graph which satisfies the target set of constraints. The edge-switching method then consists in iterating the following process:

1. select two edges randomly and exchange their ends,
2. if the graph produced satisfies the desired constraints then continue, otherwise exchange the ends back then continue.

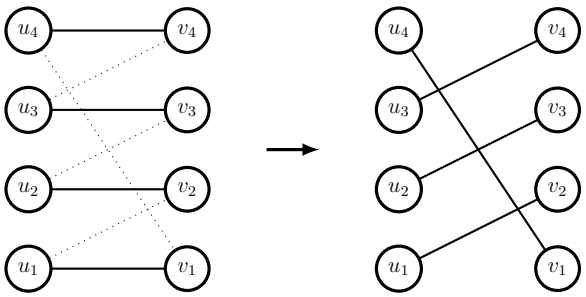
A schematic representation of the process is given on the figure below. This simple markovian process converges to a random element of the set of graphs that can be reached. Moreover, the sampling is uniform if the Markov chain is correctly conceived (see [68] for more details on this point).



In order to assess if the Markov chain has indeed reached its steady state, a common practice is to use an *ad hoc* experimental criterion [25]. For example, the user choose a structural measurement (e.g., the average distance between nodes), and evaluate how its value evolves during the process to estimate when the steady state is reached.

The main limitation of this method comes from the fact that in most cases we don't know if the process reaches the entire desired set of graphs. In other words, the markovian process may not be ergodic. A few cases are proved to be ergodic: for example the set of connected graphs with a given degree distribution [70], but most of the time there is no theoretical result, so we propose to generalize the process in order to circumvent this issue.

Multiple edge switching method. The generalized process that we propose consists in iteratively switching the ends of k edges instead of 2. At each iteration, we select k edges and a random permutation of the ends of the edges is applied, as represented in the figure below. If the modified graph belongs to the desired set, the switch is validated, otherwise we go back to the previous configuration.



It is straightforward to prove that there exists a value k_0 of k for which the process is ergodic. Unfortunately, there is no simple way to know what is the value of k_0 in most cases. Therefore, we proposed to answer this question experimentally: we apply the method for increasing values of k, starting from $k = 2$. Then, we compare the steady states of the Markovian processes by comparing the topology of the graphs obtained. When the steady states cannot be distinguished over several consecutive k values, then we make the assumption that we have reached k_0. The underlying intuition is that we rapidly reach ergodicity by increasing the value of k, which is supported by some experimental results on toy examples (see [68] for more details). An implementation in OCaml of the method with some specific sets of constraints is available at http://lioneltabourier.fr/program.html.

Limitations. I mention here several limitations of the method which should be considered when using it in practice.

- The method described suppose that the degree distribution of the graph is fixed. Indeed, the switching processes do not alter the degree distribution. Note however that in practice, a user often wants to set the degree distribution, as testified by the vast literature on graph generation with a fixed degree distribution.

- There is no theoretical guarantee on how close we are to the ergodic situation. We can only state that we are closer to ergodicity than 2 edges switching method would.

- The more elaborate the constraints, the slower the process. As previously mentioned, we don't know *a priori* the number of steps which is necessary to reach the steady state and it is estimated experimentally using heuristics. This requires to make many iterations to ensure that the steady state is reached. If the target set of constraints is elaborate then the corresponding set of graphs is small and the fraction of successful k-edge switches drops with k, which can lead to slow generation processes.

4.1.3 A case study for explanatory purposes

I describe here a use case of the method which is detailed in [65] (in French) on the family of scientific coauthoring networks. The goal of this study is to find a set of constraints that is sufficient to build realistic coauthoring networks. We suppose that the input is a coauthoring dataset, that is to say a set of articles and their authors, and we generate coauthoring graphs, which nodes are authors and they are connected if they coauthored at least one paper together.

In the literature, the bipartite configuration model described in [52] is deemed to be a good model to simulate social networks which have an event-based structure. According to this model, the number of papers authored by a scientist is fixed, as well as the number of coauthors per article. However, it is not sufficient to account for the structure of most coauthoring networks. Specifically, we focus on the structure of the projection of the bipartite graph on author nodes, that is to say the graph which nodes are the authors, and nodes are linked if they are related to at least one common article in the bipartite network. The bipartite configuration model does not correctly account for the number of small size patterns (cycles, cliques, etc.) or the distance distribution in this projection.

Our assumption is that we also need to set the number of collaborators of a scientist, as the number of one's coauthors is heavily constrained in practice (by the time available, the institutional affiliations, etc.). Consequently, the model that we proposed (named *monopartite-bipartite* configuration model and denoted MB), satisfies the following constraints:

- The bipartite degree distribution (author to articles, articles to author) is fixed, as in a standard bipartite configuration model.
- For each author with a given bipartite degree, the number of his or her coauthors is fixed. In other words, we set the degree distribution of the projection of the bipartite graph on author nodes.

To generate such graphs, we start from the real original network, then randomize it with the k-switch procedure, testing at each iteration that the constraints aforementioned are indeed satisfied. Then, we compare the topologies of the projected graphs generated to the original coauthoring graph. For comparison purposes, we generate graphs according to the standard monopartite configuration model (M), using the configuration model on the original coauthor network. We also produce another sample from the bipartite configuration model (B). I report in Figure 4.1 the distance distributions obtained on two scientific

collaboration datasets in the field of archeology, extracted from the *Anthropological Index Online*[1].

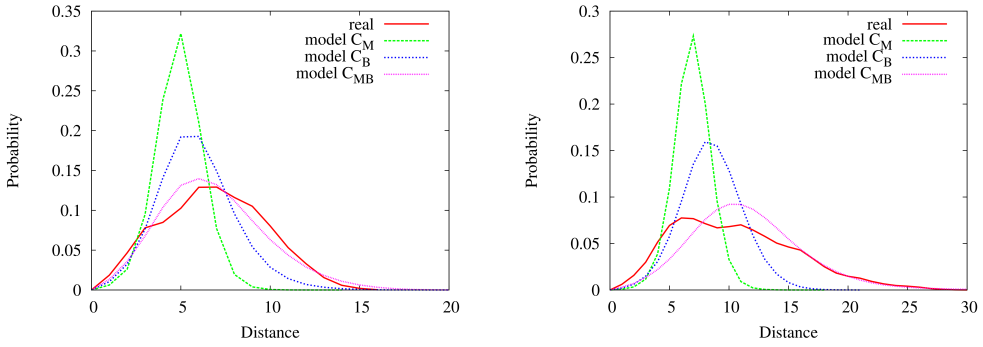


Figure 4.1 – Distance distributions of the coauthor graphs according to models M, B and MB and in the original graph. Datasets are collaboration in archeology on the topic of British isles (left) and Europe (right).

The results confirm that the MB model is much closer to the original dataset than the two other models are. However, it must be underlined that on other datasets, the results are not always as eloquent (even if the MB model steadily outperforms M and B).

From a methodological perspective, we observe that $k = 3$ is sufficient to generate the MB graphs on all the inputs that we have tested (and $k = 2$ concerning M and B, which was known from the theory). The main practical drawback of the method is the fact that the convergence of the Markov process is slow in the case of the MB model: depending on the input graph, we may have to achieve up to 10^{11} iterations (that is to say several days of computations on a standard processor).

4.1.4 A case study for simulation purposes

In the previous section, we have seen a case study which purpose is explanatory: we intended to understand which elements are sufficient to account for the structure of collaboration networks. In this section, we are interested in generating networks that have specific properties in order to explore how their structure impacts dynamic processes.

This work has been made in collaboration with economists Fanny Henriet and Stéphane Hallegatte. They were interested in evaluating how an economic network is impacted by a

[1] https://aio.therai.org.uk/

natural disaster, depending on its structural features. The specific example under consideration was the impact of hurricane Katrina on Lousiana economic network. A model to simulate the functioning of such a network at the production unit (PU) level was already available (the ARIO model [26]), and we discussed in [27] its behavior when changing the properties of its network input. More precisely, we start from a handmade synthetic network with a given degree distribution, and modify one of its structural property to a target value. Following this process for different target values, we can then investigate the effect of this structural property on the functioning of the network.

Targeting structural properties. We adapted the generating method described in Section 4.1.2 in a simple way. Let's suppose that we are targeting a specific topological constraint, for example a given clustering coefficient value. Then, we apply the same edge switch process but switches are validated only if we get closer to the target clustering level. Such a process does not generate a uniform sample of graphs, but when the target clustering value is reached, we can apply our standard generation method by including fixed clustering boundaries among the set of constraints.

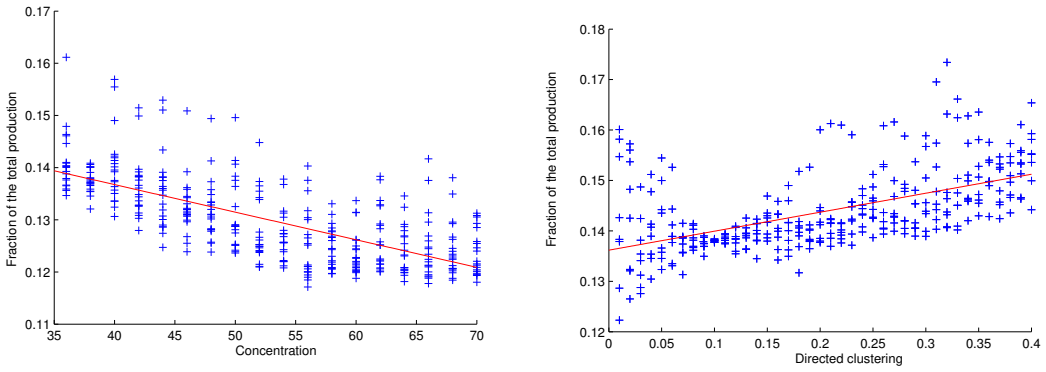


Figure 4.2 – Left: impact of the concentration on the overall production (measured as the fraction of a reference production level). Right: impact of the directed clustering coefficient.

Experimental results. The effect of two structural properties on the model have been explored: one is the directed clustering coefficient, the other is the concentration index, which measures if a PU depends on many or few other PUs (respectively low and high concentration). The natural disaster itself is modeled as the suppression of a subset

of nodes in the network. ARIO model estimates the impact by measuring the total production of the system during a given period after the disaster.

As can be seen on Figure 4.2, the effects of concentration are consistent with the intuition: the more concentrated the network, the heavier the reduction of the production. By contrast, the effects of the directed clustering are surprising: higher clustering levels correspond to a more robust network. A possible explanation is that a higher clustering creates small "islands" of PUs which are more isolated from the rest of the network and therefore less affected by the overall reduction of production.

4.2 From static to dynamic networks

The second part of this chapter is devoted to future works on the topic of generating null models in a dynamic context. Despite the limitations mentioned, we have at our disposal a flexible generation technique and we aim at generalizing it to dynamic networks.

4.2.1 Motivations

The important need for dynamic network models has been mentioned in Chapter 2: we were then looking for models that could identify specific elements which account for the properties of spreading cascades on a temporal network. We used simple models for that purpose: the popular time mixing model (see Sec. 2.1.3) which destroys the burstiness of individual activity pattern, and the correlation mixing model which has been designed to break more specifically the correlations in directed networks between the destination and the moment when a directed interaction occurs.

Various models have been proposed in the literature in order to identify the effect of other properties of dynamic networks (see [29] for a review[2]). For example, the *equal weight edge randomization* model in [33] keeps the structure unchanged but mixes the time sequences of interactions among links of equal weights, therefore breaking the weight-topology correlations. But other questions call for new dynamic network models.

For instance, it has long been said that triadic closure is one of the driving forces at the origin of social networks based on trust. Triadic closure can be translated in a dynamic network as a sequence of interactions of the form (u, v, t_1), (v, w, t_2), (w, u, t_3), where the delays satisfy specific rules, such as $0 \leq t_2 - t_1 \leq \tau$ and $0 \leq t_3 - t_2 \leq \tau$.

[2]Note also that a very recent preprint [22] reviews a large variety of models of this kind.

One way to evaluate the impact of these patterns on the global structure is to generate dynamic network models having a tunable number of these specific temporal motifs.

More generally, it is desirable to have a method that allows to progressively relax the topological and temporal constraints of the original networks. Generalizing edge switching processes could be a relevant way of doing so.

4.2.2 Challenges

Generating dynamic network null models using switch-based methods raises some issues that I describe here. On the positive side, edge timestamps may be considered as edge attributes, which can be dealt with by edge-switching methods. Indeed, attribute constraints can be managed exactly in the same way as a structural constraint, as shown in [68]. In practice, it means that an edge switch is allowed or not depending on the fact that the graph obtained satisfies the attribute constraint.

A first problem has already been mentioned: switch-based methods tend to get slower when considering more elaborate constraints. Adding temporal information to the equation certainly increases the difficulty, as we add a new dimension of constraints. Practically, addressing this problem is more of a case-by-case issue and the edge switching algorithms should be designed in order to optimize the constraint-checking phase which is frequently the time-limiting stage of the process.

Besides this technical problem, we also have to face a fundamental one. Let us consider the following model: at each timestamp, the structure is random except for the fact that the degree of a node is fixed. It corresponds to implementing a configuration model on each snapshot corresponding to a given timestamp. Using an edge-switching method, it is achieved by allowing switches among links occurring at the exact same time. A user might want to relax this rigid constraint by allowing switches among interactions occurring at close enough timestamps. But the effect would be dramatic: if (u, v, t) can be switched with $(w, x, t + 1)$, which then can be switched with $(y, z, t + 2)$, etc. we gradually destroy the temporal characteristics of the dataset.

Admittedly, this example is quite basic and one could think of means to circumvent the problem. However, it is revealing of a recurring issue when dealing with temporal constraints. It stems from the fact that in many circumstances, a user wants the model to have some flexibility on temporal constraints and may consider that two events occurring in a short timespan are in fact simultaneous. Switching methods on graphs are based on the idea of realizing a random walk on the target set of graphs, going from one element

to the other by an edge-switching operation. In the case of dynamic networks, the definition of the target set would depend on the time granularity chosen, in other words the discretization of time has an important impact on the definition of the set of temporal networks that we aim at producing. How can we manage to add some latitude to the temporal granularity of a model and still be able to generate it with an edge-switching process is an open question which I wish to address.

Chapter 5

Applicative prospects

In the course of this manuscript, I mentioned a few prospects that often combine different aspects of the topics that I have been dealing with. For instance, investigating the impact of link stream properties on spreading processes calls for the use of generating dynamic network null models. Also, as noticed in [43], the control strategies of propagation processes would greatly benefit from knowing which interactions will happen and when, that is to say from predicting interactions in a link stream. Most of these perspectives are methodological in the sense that they would bring new tools to address issues related to dynamic networks analysis.

However, my day-to-day work is often motivated by application-oriented topics. It brings specific questionings, related to the particularities of the data under study. Thus, I wish to end this manuscript with applicative works and prospects, corresponding to either ongoing or under construction projects, that I am taking part in.

Spreading in computer networks

Some specific problems of information spreading in computer networks have striking similarities with the question of epidemic spreading. The benefits that this field could draw from complex networks analysis have been acknowledged for years. For instance, an entire family of communication protocols for delay-tolerant networks (DTN) has been defined using *ad hoc* centrality measurements, either in a static, graph-based setting [18] or in a dynamic setting [31]. These applications use the fact that centralities – and in particular betweenness centrality – may be employed to identify nodes which are efficient for spreading.

However, betweenness is also known to suffer from severe drawbacks, among which its computational cost and its sensitivity to noise in the network structure. It has been suggested that spreading in computer networks does not necessarily require to use the highest centrality nodes: secondary spreaders may be sufficient [44]. Consequently, we are looking for a relaxed definition of centrality that detect nodes that are efficient enough. Also, this definition should be robust to structural changes and computable in reasonable times. In a recent work, colleagues and I proposed a substitute for betweenness centrality in the context of dynamic networks, which can be computed online, as it is based on local properties of a link stream [24]. We are now considering using this centrality to investigate DTN protocols, tackling questions such as: do the nodes used to relay information have a high betweenness? Or can we find suitable substitute nodes in the neighborhood of a relay based on their betweenness ?

Diversity in recommendation

Other works not detailed in this manuscript [11, 12], on influence and information spreading in social media have raised my interest for the social implications of complex networks analysis. Besides that, there is a deep relationship between link prediction and recommendation problems. Indeed, deciding if a user of a platform will buy a product, click on a link, etc. can be seen as the prediction of an interaction in the bipartite network associating users to items. That's why I took the opportunity to participate to the Algodiv project[1].

This ANR project gathers computer scientists and quantitative sociologists in order to address the problem of diversity on online platforms. It originates from the observation of the well-known *filter-bubble effect*. Its core idea is that web personalization algorithms lead users of online platforms to consume a relatively low variety of contents when compared to what is theoretically available [55, 14]. But there are also strong arguments supporting the idea that this observation stems instead from the natural human tendency to homophily [6], and others defend that the magnitude of this effect remains minor [21]. So, quantifying the notion of diversity and understanding the mechanisms at stake are critical issues that the project addresses.

Here, we intend to use complex networks analysis as an auditing method of recommendation systems. Our team follows a three parts research plan:

1. First, we define diversity measurements which are generic enough to represent most

[1] http://algodiv.huma-num.fr

of the situations addressed in the project. In a few words, the measurements that we have devised are based on a description of the data as a tripartite dynamic network. The three layers represent respectively the users, the contents that they select and the categories that these contents belong to.

2. The second phase, which is currently in progress, consists in describing the navigation behaviors of the users on a large French "infotainment" website. Above all, it aims at evaluating the content consumption diversity of the sessions. In parallel, we investigate the theoretical effects that classical recommendation strategies have in the three-layer environment that is used as a framework in this study.

3. The third and final phase is to confront these two aspects by running *in vivo* experiments. The idea is to substitute recommendation algorithms to the one currently used on the website (which is based on popularity) and measure the effects on the navigation behaviors. For instance, we have observed that the diversity consumed by the users is on average slightly lower than the diversity available on the platform, and both of them are relatively stable, as represented in Figure 5.1. The experiments would allow to see if a sudden increase or decrease in the diversity proposed would lead the diversity consumed to follow the same trend or not.

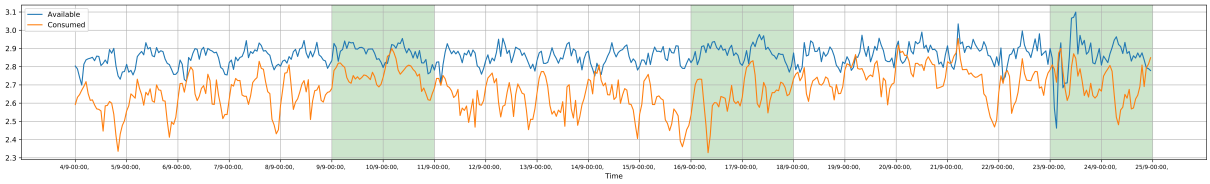


Figure 5.1 – Evolution of the diversity on the website during three weeks (entropy of the categories related to the contents). Blue: diversity available, yellow: diversity consumed.

Another collaboration project is currently under construction with industrial partners specialized in live shows recommendation. The purpose of this project is to practically implement the analysis that we have made of diversity in order to foster recommendations which allow users to discover new cultural spaces.

These projects fall within a long-term and global interrogation that I would like to tackle in my works. That is the question of how a subtle and polysemous concept should be translated into algorithms and quantitative tools in order to render their full meaning. *Diversity* is an obvious example, but social networks analysis regularly deals with such puzzles, for example when deciding what is *relevant* or *fair* to a user.

Bibliography

[1] Mohammad Al Hasan and Mohammed J Zaki. A survey of link prediction in social networks. In *Social network data analytics*, pages 243–275. Springer, 2011.

[2] Thibaud Arnoux, Lionel Tabourier, and Matthieu Latapy. Combining structural and dynamic information to predict activity in link streams. In *International Symposium on Foundations and Applications of Big Data Analytics*, 2017.

[3] Thibaud Arnoux, Lionel Tabourier, and Matthieu Latapy. Predicting interactions between individuals with structural and dynamical information. *arXiv preprint arXiv:1804.01465*, 2018.

[4] Martin Atzmueller. Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49, 2015.

[5] Paolo Bajardi, Alain Barrat, Fabrizio Natale, Lara Savini, and Vittoria Colizza. Dynamical patterns of cattle trade movements. *PloS one*, 6(5):e19869, 2011.

[6] Eytan Bakshy, Solomon Messing, and Lada A Adamic. Exposure to ideologically diverse news and opinion on facebook. *Science*, 348(6239):1130–1132, 2015.

[7] Duygu Balcan, Bruno Gonçalves, Hao Hu, José J Ramasco, Vittoria Colizza, and Alessandro Vespignani. Modeling the spatial spread of infectious diseases: The global epidemic and mobility computational model. *Journal of computational science*, 1(3):132–145, 2010.

[8] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[9] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. *Dynamical processes on complex networks*. Cambridge university press, 2008.

[10] Marc Barthélemy, Alain Barrat, Romualdo Pastor-Satorras, and Alessandro Vespignani. Velocity and hierarchical spread of epidemic outbreaks in scale-free networks. *Physical Review Letters*, 92(17):178701, 2004.

[11] Abdelhamid Salah Brahim, Bénédicte Le Grand, Lionel Tabourier, and Matthieu Latapy. Citations among blogs in a hierarchy of communities: Method and case study. *Journal of Computational Science*, 2(3):247–252, 2011.

[12] Abdelhamid Salah Brahim, Lionel Tabourier, and Bénédicte Le Grand. A data-driven analysis to question epidemic models for citation cascades on the blogosphere. In *ICWSM*, 2013.

[13] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Computer networks*, 33(1-6):309–320, 2000.

[14] Dominique Cardon. Dans l'esprit du pagerank. *Réseaux*, (1):63–95, 2013.

[15] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.

[16] Olivier Chapelle, Yi Chang, and T-Y Liu. Future directions in learning to rank. In *Proceedings of the Learning to Rank Challenge*, pages 91–100, 2011.

[17] Paulo Ricardo da Silva Soares and Ricardo Bastos Cavalcante Prudêncio. Time series based link prediction. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–7. IEEE, 2012.

[18] Elizabeth M Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40. ACM, 2007.

[19] Bhagat Lal Dutta, Pauline Ezanno, and Elisabeta Vergu. Characteristics of the spatio-temporal network of cattle movements in france over a 5-year period. *Preventive veterinary medicine*, 117(1):79–94, 2014.

[20] Stephen Eubank, Hasan Guclu, VS Anil Kumar, Madhav V Marathe, Aravind Srinivasan, Zoltan Toroczkai, and Nan Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429(6988):180, 2004.

[21] Seth Flaxman, Sharad Goel, and Justin M Rao. Filter bubbles, echo chambers, and online news consumption. *Public Opinion Quarterly*, 80(S1):298–320, 2016.

[22] Laetitia Gauvin, Mathieu Génois, Márton Karsai, Mikko Kivelä, Taro Takaguchi, Eugenio Valdano, and Christian L Vestergaard. Randomized reference models for temporal networks. *arXiv preprint arXiv:1806.04032*, 2018.

[23] Laetitia Gauvin, André Panisson, Ciro Cattuto, and Alain Barrat. Activity clocks: spreading dynamics on temporal networks of human contact. *Scientific reports*, 3:3099, 2013.

[24] Marwan Ghanem, Florent Coriat, and Lionel Tabourier. Ego-betweenness centrality in link streams. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 667–674. ACM, 2017.

[25] Christos Gkantsidis, Milena Mihail, and Ellen Zegura. Spectral analysis of internet topologies. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 364–374. IEEE, 2003.

[26] Stéphane Hallegatte. An adaptive regional input-output model and its application to the assessment of the economic cost of katrina. *Risk analysis*, 28(3):779–799, 2008.

[27] Fanny Henriet, Stéphane Hallegatte, and Lionel Tabourier. Firm-network characteristics and economic robustness to natural disasters. *Journal of Economic Dynamics and Control*, 36(1):150–167, 2012.

[28] Petter Holme. Network reachability of real-world contact sequences. *Physical Review E*, 71(4):046119, 2005.

[29] Petter Holme and Jari Saramäki. Temporal networks. *Physics reports*, 519(3):97–125, 2012.

[30] Zan Huang and Dennis KJ Lin. The time-series link prediction problem with applications in communication surveillance. *INFORMS Journal on Computing*, 21(2):286–303, 2009.

[31] Pan Hui, Jon Crowcroft, and Eiko Yoneki. Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions on Mobile Computing*, 10(11):1576–1589, 2011.

[32] Márton Karsai, Hang-Hyun Jo, and Kimmo Kaski. *Bursty human dynamics*. Springer, 2018.

[33] Márton Karsai, Mikko Kivelä, Raj Kumar Pan, Kimmo Kaski, János Kertész, A-L Barabási, and Jari Saramäki. Small but slow world: How network topology and burstiness slow down spreading. *Physical Review E*, 83(2):025102, 2011.

[34] Lauri Kovanen, Márton Karsai, Kimmo Kaski, János Kertész, and Jari Saramäki. Temporal motifs in time-dependent networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(11):P11005, 2011.

[35] Renaud Lambiotte, Lionel Tabourier, and Jean-Charles Delvenne. Burstiness and spreading on temporal networks. *The European Physical Journal B*, 86(7):320, Jul 2013.

[36] Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time. *arXiv preprint arXiv:1710.04073*, 2017.

[37] Hartmut HK Lentz, Andreas Koher, Philipp Hövel, Jörn Gethmann, Carola Sauter-Louis, Thomas Selhorst, and Franz J Conraths. Disease spread through animal movements: a static and temporal network analysis of pig trade in germany. *PloS one*, 11(5):e0155196, 2016.

[38] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *journal of the Association for Information Science and Technology*, 58(7):1019–1031, 2007.

[39] Ryan N Lichtenwalter, Jake T Lussier, and Nitesh V Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 243–252. ACM, 2010.

[40] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[41] Clémence Magnien, Frédéric Ouédraogo, Guillaume Valadon, and Matthieu Latapy. Fast dynamics in internet topology: Observations and first explanations. In *Internet Monitoring and Protection, 2009. ICIMP'09. Fourth International Conference on*, pages 137–142. IEEE, 2009.

[42] Priya Mahadevan, Dmitri Krioukov, Marina Fomenkov, Xenofontas Dimitropoulos, Amin Vahdat, et al. The internet as-level topology: three data sources and one definitive metric. *ACM SIGCOMM Computer Communication Review*, 36(1):17–26, 2006.

[43] Naoki Masuda and Petter Holme. Predicting and controlling infectious disease epidemics using temporal networks. *F1000prime reports*, 5, 2013.

[44] Dianne SV Medeiros, Miguel Elias M Campista, Nathalie Mitton, Marcelo Dias de Amorim, and Guy Pujolle. Weighted betweenness for multipath networks. In *Global Information Infrastructure and Networking Symposium (GIIS), 2016*, pages 1–6. IEEE, 2016.

[45] Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, 2004.

[46] Ron Milo, Nadav Kashtan, Shalev Itzkovitz, Mark EJ Newman, and Uri Alon. On the uniform generation of random graphs with prescribed degree sequences. *arXiv preprint cond-mat/0312028*, 2003.

[47] Giovanna Miritello, Esteban Moro, and Rubén Lara. Dynamical strength of social ties in information spreading. *Physical Review E*, 83(4):045102, 2011.

[48] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random structures & algorithms*, 6(2-3):161–180, 1995.

[49] Martina Morris and Mirjam Kretzschmar. Concurrent partnerships and the spread of hiv. *Aids*, 11(5):641–648, 1997.

[50] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[51] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[52] Mark EJ Newman and Juyong Park. Why social networks are different from other types of networks. *Physical Review E*, 68(3):036122, 2003.

[53] Maria Nöremark, Nina Håkansson, Susanna Sternberg Lewerin, Ann Lindberg, and Annie Jonsson. Network analysis of cattle and pig movements in sweden: measures relevant for disease control and risk based surveillance. *Preventive veterinary medicine*, 99(2-4):78–90, 2011.

[54] Jukka-Pekka Onnela, Jari Saramäki, Jorkki Hyvönen, György Szabó, David Lazer, Kimmo Kaski, János Kertész, and A-L Barabási. Structure and tie strengths in mobile communication networks. *Proceedings of the national academy of sciences*, 104(18):7332–7336, 2007.

[55] Eli Pariser. *The filter bubble: What the Internet is hiding from you.* Penguin UK, 2011.

[56] Romualdo Pastor-Satorras and Alessandro Vespignani. Immunization of complex networks. *Physical Review E*, 65(3):036104, 2002.

[57] Aurore Payen, Lionel Tabourier, and Matthieu Latapy. Impact of temporal features of cattle exchanges on the size and speed of epidemic outbreaks. In *International Conference on Computational Science and Its Applications*, pages 84–97. Springer, 2017.

[58] Fernando Peruani and Lionel Tabourier. Directedness of information flow in mobile phone communication networks. *PloS one*, 6(12):e28860, 2011.

[59] Manisha Pujari and Rushed Kanawati. Link prediction in complex networks by supervised rank aggregation. In *Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on*, volume 1, pages 782–789. IEEE, 2012.

[60] Luis EC Rocha, Fredrik Liljeros, and Petter Holme. Simulated epidemics in an empirical spatiotemporal network of 50,185 sexual contacts. *PLoS computational biology*, 7(3):e1001109, 2011.

[61] Christoph Scholz, Martin Atzmueller, and Gerd Stumme. On the predictability of human contacts: Influence factors and the strength of stronger ties. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 312–321. IEEE, 2012.

[62] Michele Starnini, Andrea Baronchelli, Alain Barrat, and Romualdo Pastor-Satorras. Random walks on temporal networks. *Physical Review E*, 85(5):056115, 2012.

[63] Lionel Tabourier. *Méthode de comparaison des topologies de graphes complexes: applications aux réseaux sociaux*. PhD thesis, Paris 6, 2010.

[64] Lionel Tabourier, Daniel Faria Bernardes, Anne-Sophie Libert, and Renaud Lambiotte. Rankmerging: A supervised learning-to-rank framework to predict links in large social network. *arXiv preprint arXiv:1407.2515*, 2015.

[65] Lionel Tabourier, Jean-Philippe Cointet, and Camille Roth. Génération de graphes aléatoires par échanges multiples d'arêtes. *Journal de la Société Française de Statistique*, 158(2):118–134, 2017.

[66] Lionel Tabourier, Anne-Sophie Libert, and Renaud Lambiotte. Rankmerging: Learning to rank in large-scale social networks. In *DyNakII, 2nd International Workshop on Dynamic Networks and Knowledge Discovery (PKDD 2014 workshop)*, 2014.

[67] Lionel Tabourier, Anne-Sophie Libert, and Renaud Lambiotte. Predicting links in ego-networks using temporal information. *EPJ Data Science*, 5(1):1, 2016.

[68] Lionel Tabourier, Camille Roth, and Jean-Philippe Cointet. Generating constrained random graphs using multiple edge switches. *Journal of Experimental Algorithmics (JEA)*, 16:1–7, 2011.

[69] Lionel Tabourier, Alina Stoica, and Fernando Peruani. How to detect causality effects on large dynamical communication networks: a case study. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference On*, pages 1–7. IEEE, 2012.

[70] Robert Taylor. Constrained switchings in graphs. *Combinatorial Mathematics*, 8:314—336, 1980.

[71] Alexei Vazquez, Balazs Racz, Andras Lukacs, and Albert-Laszlo Barabasi. Impact of non-poissonian activity patterns on spreading processes. *Physical review letters*, 98(15):158702, 2007.

[72] Tiphaine Viard, Matthieu Latapy, and Clémence Magnien. Computing maximal cliques in link streams. *Theoretical Computer Science*, 609:245–252, 2016.

[73] Fabien Viger and Matthieu Latapy. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. In *International Computing and Combinatorics Conference*, pages 440–449. Springer, 2005.

[74] Yang Yang, Ryan N Lichtenwalter, and Nitesh V Chawla. Evaluating link prediction methods. *Knowledge and Information Systems*, 45(3):751–782, 2015.

[75] Qian Zhang, Márton Karsai, and Alessandro Vespignani. Link transmission centrality in large-scale social networks. *arXiv preprint arXiv:1802.05337*, 2018.